Bounded Relay Hop Mobile Data Gathering in Wireless Sensor Networks

Miao Zhao and Yuanyuan Yang
Department of Electrical and Computer Engineering, State University of New York, Stony Brook, NY 11794, USA

Abstract— **Recent study reveals that great benefit can be achieved for data gathering in wireless sensor networks by employing mobile collectors that gather the data via short-range communications. To pursue maximum energy saving at sensor nodes, intuitively, a mobile collector should traverse the transmission range of each sensor in the field such that the transmission of each packet can be constrained to a single hop. However, this approach may lead to significantly increased data collection latency due to the low moving velocity of the mobile collector. On the other hand, data collection latency can be effectively shortened by performing local aggregation via multi-hop transmissions and then uploading the packets from relay sensors to the mobile collector. However, local transmission hops should not be arbitrarily increased since it may incur too much energy consumption on packet relays, which would adversely affect the overall efficiency of mobile data collection. Based on these observations, in this paper, we study the tradeoff between energy saving and data collection latency in mobile data gathering by exploring a balance between the relay hop count of local data aggregation and the moving tour length of the mobile collector. We first propose a polling-based mobile collection approach and formulate it into an optimization problem, named *bounded relay hop mobile data collection (BRH-MDC)*. Specifically, a subset of sensors will be selected as polling points that buffer locally aggregated data and upload the data to the mobile collector when it arrives. In the meanwhile, when sensors are affiliated with these polling points, it is guaranteed that any packet relay is bounded within a given number of hops. We then give two efficient algorithms to select polling points among sensors. The effectiveness of our approach is validated through extensive simulations.**

I. Introduction

Recent years have witnessed the emergence of wireless sensor networks (WSNs) as a new information-gathering paradigm, in which a large number of sensors scatter over a surveillance field and extract data of interests by reading real-world phenomena from the physical environment. Since sensors are typically battery-powered and left unattended after the initial deployment, it is generally infeasible to replenish the power supplies once they deplete the energy. Thus, energy consumption becomes a primary concern in a WSN, as it is crucial for the network to functionally operate for an expected period of time.

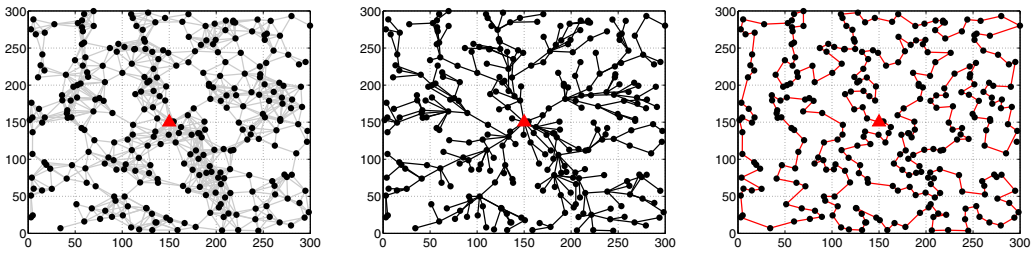
Besides the energy consumed on monitoring the environment with periodical sampling, a major portion of energy expenditure in WSNs is attributed to the activities of aggregating data to the data sink. Due to the stringent energy constraints in WSNs, recent research has striven to address the issue of energy saving in data aggregation. One trend of the research, see, for example, [1]-[6], focused on sensor nodes themselves. In such schemes, data packets are forwarded to the data sink via multi-hop relays among sensors. Some related issues, such as schedule pattern [1], load balance [2], and data redundancy [3]-[6], were also jointly considered along with routing to further improve the energy efficiency. However, due to the inherent nature of multi-hop routing, packets have to experience multiple relays before reaching the data sink. As a result, much energy is consumed on data forwarding along the path. Moreover, minimizing energy consumption on the forwarding path does not necessarily prolong network lifetime as some popular sensors on the path may run out of energy faster than others, which may cause non-uniform energy consumption across the network.

Another recent trend of the research indicated a focus shift to mobile data collection, which employs one or more mobile collectors that are robots or vehicles equipped with powerful transceivers and batteries. A typical scenario is that a mobile collector roams over a sensing field, "transports" data while moving, or pauses at some anchor points on its moving path to collect data from sensors via short-range communications. In this way, energy consumption at sensors can be greatly reduced, since the mobility of the collector effectively dampens the relay hops of each packet. Intuitively, to pursue maximum energy saving, a mobile collector should traverse the transmission range of each sensor in the field so that each packet can be transmitted to the mobile collector in a single hop. However, due to the low velocity of the mobile collector, it would incur long latency in data collection, which may not meet the delay requirement of time-sensitive applications.

According to the empirical studies [18], the packet relay speed in a WSN is about several hundred meters per second, which is much higher than the velocity at which the mobile collector moves. Hence, in general, the latency of multi-hop relay routing and its variants is much shorter than that of the mobile collection. Whereas, as aforementioned, mobile collection pursues energy saving by simply reducing the relay hops among sensors. From these observations, it is clear that there is an intrinsic tradeoff between the energy saving and the data collection latency. To better understand this tradeoff, we illustrate it with an example in Fig. 1. A network with 300 sensors is configured as shown in Fig. 1(a) with the static data sink located at the center of a 300m by 300m field. When we adopt multi-hop routing for data collection and each packet is forwarded along its shortest path with the minimum hop count to the data sink, the result is depicted in Fig. 1(b), where each packet needs 5.3 hops on average to reach the data sink. On the other hand, when a mobile collector is employed, one of the extreme cases for energy saving is that the mobile collector gathers data packets by sequentially visiting each sensor, which guarantees that each sensor can upload data within a single hop. In this way, the number of transmissions is greatly reduced, however, the mobile collector has to travel along a tour of length 4012m as shown in Fig. 1(c). Since the typical velocity of a practical mobile system is about 0.1 - 2m/s [12], it will

Research supported by NSF grant numbers ECCS-0801438 and ECS-0427345 and ARO grant number W911NF-09-1-0154.

978-1-4244-5113-5/09/$25.00 ©2009 IEEE



(a) 300 sensors deployed over a 300m × 300m field.

(b) Relay routing along shortest paths with minimum hop counts.

(c) Mobile collection by visiting each sensor and static data sink.

Fig. 1. An example to illustrate the tradeoff between the energy saving and the data collection latency in a sensor network.

take the mobile collector about 66.9 minutes on the tour when it moves at an average speed of 1m/s.

Therefore, in order to shorten data collection latency, it is necessary to incorporate multi-hop relay into mobile data collection, while the hop count should be constrained to a certain level to limit the energy consumption at sensors. In this paper, we address this issue by proposing a polling-based approach that pursues a tradeoff between the energy saving and data collection latency, which achieves a balance between the relay hop count for local data aggregation and the moving tour length of the mobile collector. Specifically, a subset of sensors will be selected as the *polling points* (PPs), each aggregating the local data from its affiliated sensors within a certain number of relay hops. These PPs will temporarily cache the data and upload them to the mobile collector when it arrives.

The main contributions of this paper can be summarized as follows. (1) We characterize the polling-based mobile data collection as an optimization problem, named *bounded relay hop mobile data collection,* or *BRH-MDC* for short. We then formulate it into an integer linear program (ILP) and prove its NP-hardness. (2) We propose two efficient algorithms to find a set of PPs. The first algorithm is a centralized algorithm that places the PPs on the shortest path trees rooted at the sensors closest to the data sink, and takes into consideration the constraints on relay hops for local aggregation while shortening the tour length of the mobile collector. The second algorithm is a distributed algorithm, where sensors compete to be a PP based on their priorities in a distributed manner. (3) We evaluate the performance of the proposed algorithms by comparing them not only with the optimal solution obtained by CPLEX [19] based on our ILP formulation modeling in AMPL [20], but also with other two existing mobile data collection schemes. Simulation results demonstrate that the proposed algorithms achieve superior performance.

The rest of the paper is organized as follows. Section II reviews the related work on mobile data collection. Section III outlines the polling-based approach and formulate the BRH-MDC problem. Sections IV and V present two algorithms to solve the BRH-MDC problem. Section VI evaluates the efficiency of the proposed algorithms with extensive simulations. Finally, Section VII concludes the paper.

II. Related work

In this section, we briefly review some recent work on mobile data collection in wireless sensor networks. Based on the mobility pattern, we can divide the mobile data collection schemes into two categories.

The first category has uncontrollable mobility, in which the mobile collector either has random motion or moves along a fixed track, see, for example, [7]-[12]. In [7], Shah, et. al used a special type of mobile nodes to facilitate connectivity among static sensors and transport data with random mobility. Jea, et. al [8] further restricted the mobile nodes to move along straight lines to collect data in the vicinity of the lines. In [9][10], radio-tagged zebras and whales were used as mobile nodes in a wild area. Batalin, et. al [11], [12] setup a system named NIMs, where mobile collectors can only move along fixed cables between trees to ensure that they can be recharged any time during the movement. The common feature of these approaches is that the system maintenance is simple with high stability and reliability. However, they lack agility and cannot be adaptive to the sensor distribution and environmental dynamics.

The second category has controlled mobility, in which mobile collectors can freely move to any location in the field and its trajectory can be planned for specific purposes, see, for example, [13]-[18]. Within this category, the schemes can be further divided into three sub-classes. In the first subclass, the mobile collector is controlled to visit each sensor or traverse the transmission range of each sensor and gather data from them with single hop transmission [13][14]. Somasundara, et. al [13] studied the scheduling of mobile elements to ensure no data loss due to buffer overflow. To achieve perfect uniformity of energy consumption, Ma and Yang [14] proposed the planning algorithms for a short data collection tour to ensure that each data uploading is completed within a single hop. While these approaches minimize the energy cost and balance energy consumption among different sensors by completely avoiding multi-hop relays, they may result in long data collection latency especially in a large-scale sensor network. In the second subclass, mobile collectors gather data from the sensors in the vicinity via multi-hop transmissions along its trajectory. Ma and Yang [15] gave a moving path planning algorithm by finding some turning points, which is adaptive to the sensor distribution and can effectively avoid obstacles on the path. In this scheme, along each moving line segment, the sensors forward packets to the mobile collector in multiple hops. Luo and Hubaux [16] proposed that data packets should be gathered with multi-hop relays while the tour is along the perimeter of the sensing field, which is considered as the optimal path for the mobile collector. These approaches can effectively shorten the moving tour of the mobile collector, however, they do not impose any constraint on the relay hop count. As a result, network lifetime (or a certain level of energy efficiency) cannot be guaranteed. The last subclass includes the approaches that

jointly consider data transmissions and the moving tour length. For instance, Zhao, et. al [17] proposed a data collection scheme that jointly considers the full utilization of concurrent data uploading and minimizing the tour length. In the scheme, multiple sensors can simultaneously upload their data packets to the mobile collector in a single hop, which efficiently shortens the data uploading time. Xing, et. al [18] proposed a rendezvous design to minimize the distance of multi-hop routing paths for local data aggregation under the constraint that the tour length of the mobile collector is no longer than a threshold. Our work in this paper falls into this subclass, which aims to minimize the tour length of the mobile collector and guarantee the local data aggregation within bounded relay hops.

III. BRH-MDC Problem

In this section, we first give an overview of the proposed polling-based mobile data collection scheme and then formulate it into an optimization problem.

A. Overview

Since the mobile collector has the freedom to move to any place in the sensing field, it provides us an opportunity to plan an optimal tour for it. Our basic idea is to find a set of special nodes referred to as *polling points* (PPs) in the network and determine the tour of the mobile collector by visiting each PP in a specific sequence. With sensors properly affiliated with these PPs, the relay routing for local data aggregation is constrained within d hops, where d is a system parameter for the relay hop bound. Or, alternatively, we can say that a PP covers its affiliated sensors within d hops. The PPs can simply be a subset of sensors in the network or some other special devices, such as storage nodes [21] with large memory and more battery power. In the latter case, the storage nodes will not necessarily be placed at the positions of sensors, which may bring more flexibility for the tour planning. However, such special devices will incur a significantly amount of extra cost. Therefore, in this paper, we will focus on selecting a subset of sensors as the PPs. Each PP temporarily buffers the data originated from its affiliated sensors. When the mobile collector arrives, it will poll each PP to request data uploading. Upon receiving the polling message, a PP uploads data packets to the mobile collector in a single hop. The mobile collector starts its tour from the static data sink, which is located either inside or outside the sensing field, collects data packets at the PPs and then returns the data to the data sink. Since the data sink is the starting and ending points of the data collection tour, it can also be considered as a special PP. We refer to this scheme as the *polling-based mobile data collection scheme*. It is further illustrated in Fig. 2, where the sensors in the shadowed area will locally aggregate data packets to their affiliated PP within two hops (i.e, $d = 2$). For generality, we do not make any assumption on the distribution of the sensors or node capability, such as location-awareness. Each sensor is assumed to be able to communicate only with its neighbors, that is, the nodes within its proximity.

In practice, there are at least two reasons that the relay hop count should be bounded. First, a sensor network may expect to achieve a certain level of systematic energy efficiency. For instance, if each transmission costs one unit of energy and the energy efficiency of 0.33 energy_unit/packet is expected, each packet forwarding from its originating sensor to the data sink should be no more than three hops on average, i.e., each packet should be relayed to its PP within two hops. Second, the bound is necessary due to buffer constraint on the sensors. Since the PPs need to buffer the locally aggregated data before the mobile collector arrives, it is not desirable to associate too many sensors with a PP. Otherwise, the buffer of the PP may not be able to accommodate all the data packets. For example, consider a sensor network with an average node degree of 4. If a sensor is selected as a PP and the local relaying is constrained within two hops, there will be up to 20 sensors affiliated with this PP. Therefore, the buffer capacity of the PPs and the sensor density will limit the relay hops.

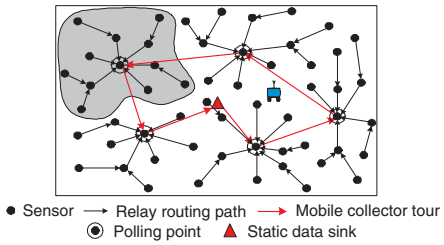


Fig. 2. Illustration of polling-based mobile data collection with d set to 2.

B. BRH-MDC Problem Formulation

Having described the polling-based mobile data collection scheme, in this subsection, we formulate it into an optimization problem, named *bounded relay hop mobile data collection,* or *BRH-MDC* for short. Our objective is to find a subset of sensors as the PPs and a set of routing paths that connect each sensor in the field to a PP within d hops, such that the tour length of the mobile collector can be minimized. The problem is formally defined as follows.

Definition 1: (Bounded Relay Hop Mobile Data Collection (BRH-MDC) Problem). Given a set of sensors $\mathcal{S}$ and a relay hop bound d, find (1) A subset of $\mathcal{S}$, denoted by $\mathcal{P}$ ($\mathcal{P} \subseteq \mathcal{S}$), which represents the PPs; (2) A set of geometric trees $\{T_i(V_i, E_i)\}$ that are rooted at each PP in $\mathcal{P}$ and $\bigcup_i V_i = \mathcal{S}$. The depth of each geometric tree is at most d; (3) The data collection tour U by visiting each PP in $\mathcal{P}$ and the data sink π exactly once, such that $\sum_{(u,v)\in U} |uv|$ is minimized, where $u, v \in \mathcal{P} \cup \{\pi\}$, (u, v) is a line segment on the tour and $|uv|$ is its Euclidean distance.

Apparently, the BRH-MDC problem consists of several subproblems. The first one is the affiliation pattern of the sensors, which can be stated as: by which PP a sensor is covered within the relay hop bound. The second one is how to construct a routing tree rooted at a particular PP with depth at most d that connects all its affiliated sensors. The third one is to find a shortest round trip among the PPs and the static data sink, which is exactly the Traveling Salesman Problem (TSP). The key characteristic of the BRH-MDC problem is that these three subproblems should be jointly considered in order to find optimal PPs among the sensors. Based on these subproblems and using the notations in Table 1, the BRH-MDC problem can be formulated as the following integer linear program.

$$\textbf{Minimize} \quad \sum_{u,v\in\mathcal{S}\cup\{\pi\},u\neq v} l_{uv}e_{uv} \tag{1}$$

$$\text{Subject to} \quad a_{iu} \leq I_u, \forall i, u \in \mathcal{S} \tag{2}$$

TABLE I
NOTATIONS USED IN FORMULATION OF BRH-MDC PROBLEM

Indices:		
$\mathcal{S} = \{1, 2, \ldots, N\}$		A set of sensors, which is also the set of candidate PPs.
π		The static data sink.
Constants:		
$d > 0$		The relay hop bound for local data aggregation.
$f_{ij} = \{0, 1\}$	$\forall i, j \in \mathcal{S}$	If sensor i is one-hop neighbor of sensor j, $f_{ij} = 1$, otherwise, $f_{ij} = 0$.
$l_{ij} > 0$	$\forall i, j \in \mathcal{S}$	Distance between sensor i and sensor j.
Variables:		
$I_i = \{0, 1\}$	$\forall i \in \mathcal{S} \cup \{\pi\}$	If sensor i is selected as a PP, $I_i = 1$, otherwise, $I_i = 0$. The static data sink is a special PP, i.e., $I_\pi = 1$.
$a_{iu} = \{0, 1\}$	$\forall i, u \in \mathcal{S}$	If sensor i is on the routing tree rooted at PP (= sensor j), $a_{ij} = 1$, otherwise, $a_{ij} = 0$.
$x^h_{iju} = \{0, 1\}$	$\forall i, j, u \in \mathcal{S}$	A node (i, u, h) is associated with sensor i, if the path from which to PP (= sensor u) contains h arcs.
	$h = 1, 2, \ldots, d$	If arc $\{(i, u, h-1), (j, u, h)\}$ is contained in the optimal solution, $x^h_{iju} = 1$, otherwise, $x^h_{iju} = 0$. When $i = j$, x^h_{iiu} represents whether sensor i is on the layer h of the routing tree rooted at PP (= sensor u).
$e_{uv} = \{0, 1\}$	$\forall u, v \in \mathcal{S} \cup \{\pi\}$	If the moving tour contains the line segment between u and v, $e_{uv} = 1$, otherwise, $e_{uv} = 0$.
$y_{uv} > 0$	$\forall u, v \in \mathcal{S} \cup \{\pi\}$	The flow from sensor u to sensor v.

$$\sum_{u \in \mathcal{S}} a_{iu} = 1, \forall i \in \mathcal{S} \tag{3}$$

$$\sum_{i \in \mathcal{S}} a_{iu} \geq I_u, \forall u \in \mathcal{S} \tag{4}$$

$$a_{uu} = I_u, \forall u \in \mathcal{S} \tag{5}$$

$$x^h_{iju} <= I_u, \forall i, j, u \in \mathcal{S}, h = 1, 2, \ldots, d \tag{6}$$

$$x^h_{uuu} = 0, \forall u \in \mathcal{S}, h = 1, 2, \ldots, d \tag{7}$$

$$x^h_{iju} \leq \frac{1}{2}(a_{iu} + a_{ju}) \cdot f_{ij}, \forall i, j, u \in \mathcal{S}, i < j, h = 1, 2, \ldots, d \tag{8}$$

$$x^1_{iiu} = x^1_{uiu} = a_{iu} \cdot f_{iu}, \forall i, u \in \mathcal{S} \tag{9}$$

$$\sum_{h=1}^{d} \sum_{u \in \mathcal{S}} x^h_{iiu} = 1 - I_u, \forall i \in \mathcal{S} \tag{10}$$

$$x^h_{iiu} = \sum_{j \in \mathcal{S}, i \neq j} x^h_{jiu}, \forall i, u \in \mathcal{S}, i \neq u, h = 1, 2, \ldots, d \tag{11}$$

$$x^h_{iju} \leq 0.5 \cdot \left(x^{h-1}_{iiu} + x^h_{jju} \right), \forall i, j \in \mathcal{S}, i \neq j, h = 2, \ldots, d \tag{12}$$

$$\sum_{h=1}^{d} \sum_{i,j \in \mathcal{S}, i \neq j} x^h_{iju} = \sum_{i \in \mathcal{S}, i \neq u} a_{iu}, \forall u \in \mathcal{S} \tag{13}$$

$$\sum_{v \in \mathcal{S} \cup \{\pi\}} e_{uv} = I_u, \forall u \in \mathcal{S} \cup \{\pi\} \tag{14}$$

$$\sum_{u \in \mathcal{S} \cup \{\pi\}} e_{uv} = I_v, \forall v \in \mathcal{S} \cup \{\pi\} \tag{15}$$

$$y_{uv} \leq (|\mathcal{S}| + 1) \cdot e_{uv}, \forall u, v \in \mathcal{S} \cup \{\pi\}, u \neq v \tag{16}$$

$$\sum_{u \in \mathcal{S} \cup \{\pi\}, u \neq \pi} y_{u\pi} = \sum_{u \in \mathcal{S} \cup \{\pi\}} I_u \tag{17}$$

$$\sum_{v \in \mathcal{S} \cup \{\pi\}, u \neq v} y_{uv} - \sum_{w \in \mathcal{S} \cup \{\pi\}, w \neq u} y_{wu} = I_u, \forall u \in \mathcal{S} \cup \{\pi\} \tag{18}$$

In the above formulation, objective function (1) minimizes the tour length of the mobile collector, which also implies the shortened latency for data collection. The constraints are explained as follows.

Constraints (2)-(5) for subproblem 1: These constraints ensure that a sensor should be affiliated with (or covered by) one and only one PP such that its sensing data can be collected during the tour. A sensor selected as a PP will be affiliated with itself.

Constraints (6)-(13) for subproblem 2: Since the relay hop bound d limits the number of hops between the root PP and its affiliated sensors, each geometric tree can be considered as having at most d layers. Sensor i is associated with a triple (i, u, h) and variable x^h_{iiu} indicates whether sensor i is in layer h ($1 \leq h \leq d$) of the tree rooted at sensor u. Variable x^h_{iju} ($i \neq j$) is associated with arc $\{(i, u, h-1), (j, u, h)\}$, which indicates whether arc (i, j) is on the geometric tree rooted at sensor u with sensors i and j in layers $h-1$ and h, respectively. Constraints (6)-(7) guarantee that each sensor can only be associated with the tree rooted at a PP and exclude the sensors selected as the PPs since they are automatically the roots and will not be in any layer of a tree (or can be considered in layer 0). Constraints (8)-(9) address that only if two neighboring sensors, say, i and j, are simultaneously affiliated with the same PP u, arc $\{(i, u, h-1), (j, u, h)\}$ is qualified to be on the tree in the optimal solution. For the special case that sensor i is the neighbor of its affiliated root u, sensor i will be in layer 1 of the tree rooted at u, and arc $\{(u, u, 0), (i, u, 1)\}$ is the edge connected u and i on the tree. Constraints (10)-(12) enforce that each sensor can only be in one layer of a tree and it has only one connection with the sensors in the immediate upper layer to ensure the tree structure. Constraints (13) indicates that the number of edges on each tree is equivalent to the number of affiliated sensors excluding the PP itself.

Constraints (14)-(18) for subproblem 3: Constraints (14) - (15) guarantee that the mobile collector enters and departs each PP as well as the data sink only once. Constraint (16) restricts that the network flow can take place only when the arc is on the moving tour. Constraints (17)-(18) enforce that for each PP, the units of outgoing flow are one unit more than that of the incoming flow. The flow units entering the data sink, which acts as the starting and ending points of the tour, are equal to the number of PPs [14]. It has been shown in [22] that constraints (16)-(18) can effectively exclude the solution with subtours.

We have the following theorem concerning the BRH-MDC problem.

Theorem 1: The BRH-MDC problem is NP-hard.

Proof: The NP-hardness of the BRH-MDC problem can be shown by giving a polynomial-time reduction from TSP problem to a special case of BRH-MDC problem. Given a complete graph $G = (V, E)$ as an instance of TSP. We construct an instance of BRH-MDC on graph $G' = (V', E')$, which is topologically identical to G. V' is the set of vertices that include all the sensors and the data sink, and E' represents the edges between any two vertices. We let the sensors be unreachable from each other for wireless transmissions, which can be achieved by reducing the transmission range below a certain level. This reduction is straightforward and can

certainly be done in polynomial time. Now, in this case it is infeasible for the data packets of a sensor to be relayed by others. The mobile collector has to visit each sensor to gather the data packets, which implies that all the sensors and the data sink are the PPs. Hence, the tour length of the data collection in G' will be the same as the total cost in G. The TSP in G will have a path with minimum cost in distance if and only if the same path in G' is the tour with the minimum length for BRH-MDC. Thus, the BRH-MDC problem is NP-hard. ■

IV. Centralized Algorithm for BRH-MDC Problem

Due to the NP-hardness of the BRH-MDC problem, in this section, we first develop a centralized heuristic algorithm for the BRH-MDC problem. It will serve as a basis for the distributed algorithm in the next section.

As discussed earlier, in order to find the optimal PP locations among the sensors, relay routing paths and the tour of the mobile collector should be jointly considered. On one hand, when no mobile collector is employed, for each sensor, the best way to relay data packets to the static data sink is along its shortest path with the minimum hop count, under the assumption that energy consumption is proportional to the number of transmissions. On the other hand, when a mobile collector is available, the data collection tour can be effectively shortened in two ways: First, the sensors selected as the PPs are compactly distributed and close to the data sink. Second, the number of the PPs is the smallest under the constraint of the relay hop bound. Based on these observations, we propose an algorithm, named *shortest path tree based data collection algorithm (SPT-DCA)* with its pseudo code listed in Algorithm 1. The basic idea of the algorithm is to iteratively find a PP among the sensors on a shortest path tree (SPT), which is the nearest sensor to the root that can connect the remote sensors on the tree. Also, each PP strives to link as many as possible sensors it can reach within the relay hop bound in order to minimize the total number of PPs.

The first task of SPT-DCA is to construct SPTs that cover every sensor in the network (see Algorithm 1, line 1). Since the network can be disconnected, there may exist more than one SPTs when the sensors are sparsely distributed. Considering this, when we find a root for the SPT to be constructed, we will choose the sensor closest to the data sink from the sensors not on the existing SPTs. We simply call it a "centroid." The motivation of introducing "centroid" rather than randomly choosing a sensor as the root is that we expect the PPs could converge towards the static data sink. Each SPT would link all the possible sensors under the connectivity restriction. This way, our scheme can be applied to not only connected networks, but also disconnected networks, which is one of the main advantages of the mobile data collection over the traditional relay routing.

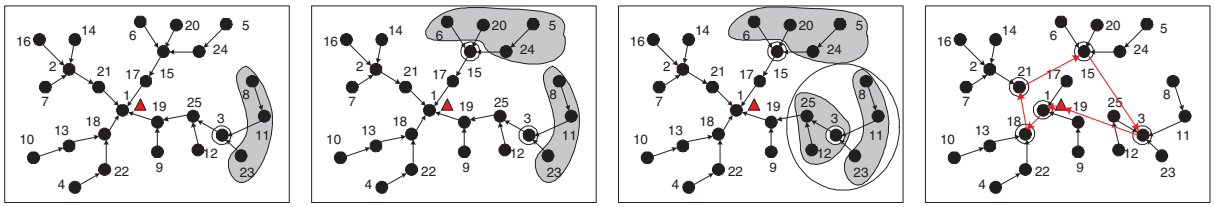
The next task of SPT-DCA is to iteratively find a PP on SPTs. We consider the sensor network as a graph $G(V, E)$, where $V = \mathcal{S}$ represents all the sensors in the network, and E is the set of edges connecting any two neighboring sensors. In the following discussion, for clarity and simplicity, we will focus on a single SPT. The operation of the algorithm can be described as follows. We consider a SPT denoted by $T'(V', E')$ with $V' \subseteq V$ and $E' \subseteq E$. In each step, we first find the

Algorithm 1: Centralized algorithm: SPT-DCA

Input: A sensor network $G(V, E)$, the relay hop bound d, and the static data sink π.
Output: A set of PPs $\mathcal{P}$, a set of geometric trees $\{t_u | u \in \mathcal{P}\}$, and the tour U visiting the PPs and the data sink.

1 Construct SPTs for G that cover all the vertices in V;
2 **for** *each SPT* $T'(V', E')$ **do**
3 **while** T' *is not empty* **do**
4 Find the farthest leaf vertex v on T';
5 **if** v *is not a PP* **then**
  // Find v's d-hop parent vertex u on T'.
6 **for** $i = 1$ *to* d **do**
7 $u \longleftarrow$parent(v); $v \longleftarrow u$;
8 **if** *u is the root of* T' **then** Break;
9 Consider u as a PP and add corresponding sensor into $\mathcal{P}$;
10 **if** u *is not the root of* T' **then**
11 Update T' by removing all the child vertices of u and the pertinent edges.
12 Corresponding sensors of these removed vertices are affiliated with u on the geometric tree t_u.
13 **else**
14 All the sensors on T' are affiliated with t_u;
15 T' is set to be empty;
16 **else**
17 **if** $d = 1$ **then**
18 Remove v from current T' and it belongs to t_v;
19 **else**
  // Find v's $\lfloor \frac{d}{2} \rfloor$-hop parent vertex w on T'.
20 **for** $i = 1$ *to* $\lfloor \frac{d}{2} \rfloor$ **do**
21 $w \longleftarrow$parent(v); $v \longleftarrow w$;
22 **if** *w is the root of* T' **then** Break;
23 **if** *w is not the root of* T' **then**
24 Remove the subtree rooted at w from T';
25 Corresponding sensors on the removed subtree are affiliated with v on the geometric tree t_v.
26 **else**
27 All the sensors on T' that are not selected as PPs are affiliated with v on the geometric tree t_v;
28 T' is set to be empty;
29 Find an approximate shortest tour U visiting π and all the PPs in $\mathcal{P}$;

farthest leaf vertex v on T'. There are two possible cases for v, either already being a PP or not. In the first case that v has not been selected as a PP yet (see Algorithm 1, lines 5-15), T' is traversed along the shortest path of v towards the root to find its d-hop parent vertex. Let u denote the d-hop parent of v. Since v is the vertex with the farthest depth, all other child vertices of u can reach u within d hops. Hence, we can let the corresponding sensor u be the PP found in the current iteration since it is the nearest one to the root that can connect the sensors in the periphery of the network based on the SPT structure. Then T' is updated by removing all the child vertices of u and their pertinent edges, which implies that the corresponding sensors will be affiliated with u for local data aggregation. It is worth pointing out that we still keep u on



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Final result.

Fig. 3. An example to illustrate the SPT-DCA algorithm ($N = 25, d = 2$).

the updated T' in order to facilitate the possible affiliations of other nearby sensors with u in the future iterations. In the rare case that the root of T' was reached during the process of finding the d-hop parent vertex of v, the algorithm terminates since all the vertices on current T' are definitely within d hops to the root. Correspondingly, the root will be selected as the PP. In the second case that the farthest leaf vertex v on current T' has already been selected as a PP (see Algorithm 1, lines 16-28), we aim to affiliate more sensors with v if possible in order to limit the number of PPs. Specifically, in order to find more sensors in the vicinity of v, we first find v's $\lfloor \frac{d}{2} \rfloor$-hop parent vertex w. As v is the farthest leaf vertex on current T', all other child vertices of w will be within $\lfloor \frac{d}{2} \rfloor$ hops away from w so that they are able to reach v within d hops along the edges on T'. Hence, besides the existing affiliated sensors of v, the sensors on the subtree rooted at w can also be affiliated with v. Thus, all the affiliated sensors of a PP will be found in these two steps. The inherited edges among these sensors from T' will be used to determine their relay paths to the affiliated PP for local data aggregation.

To better understand the algorithm, we give an example in Fig. 3, where 25 sensors are scattered over a field with the static data sink located in the center of the area, and d is set to 2, which means that it is required for each sensor to forward its data to the affiliated PP within two hops. The constructed SPT among the sensors rooted at sensor 1, denoted by T', are depicted in Fig. 3(a). In the first iteration, sensor 8 is found as the farthest leaf vertex on T' with 5 hops away from the root, i.e., $v = 8$. Its 2-hop parent vertex u on current T' is sensor 3 (i.e., $u = 3$.), which will be marked as a PP. All the child vertices of u, including sensors 8, 11 and 23, and their associated edges will be removed from T'. The result is depicted in Fig. 3(a), where sensor 3 is still kept on the updated SPT, and the removed vertices highlighted by the shadowed area are its affiliated sensors found in the current iteration. In the second iteration, the farthest leaf vertex on the updated T' turns to be sensor 5 with 4 hops away from the root. Similarly, its 2-hop parent (i.e., sensor 15) is selected as another PP to cover the sensors in the other shadowed area as shown in Fig. 3(b). In the third iteration, sensor 3 is chosen as the farthest leaf vertex on current T' and it happens to be marked as a PP already. In this case, we strive to search for more qualified sensors to affiliate with it. We find that sensor 25 is its 1-hop parent, i.e., $w = 25$. Sensor 25 and all its child vertices on current T' can reach sensor 3 within two hops along the edges on T'. Therefore, the subtree rooted at sensor 25 will be pruned from T'. All the sensors on the subtree, including sensors 25, 12 and 3 will also be affiliated with sensor 3. Fig. 3(c) indicates that a total of 6 sensors will be covered by sensor 3, which are found in iterations 1 and 3, respectively. In this way, T' is decomposed into a set of subtrees, each of which contains a selected PP and its affiliated sensors. Fig. 3(d) gives the final result, where the data collection tour is highlighted by the red line segments linking the PPs and the static data sink.

We now analyze the time complexity of SPT-DCA. Assume that there are a total of N sensors distributed in K disconnected subnetworks ($1 \leq K \leq N$). For subnetwork k ($k = 1, 2, \ldots, K$), it takes $\mathcal{O}(N_k^2)$ time to find a SPT [23], where N_k represents the number of sensors in subnetwork k. It takes $\mathcal{O}(N_k^2 + N_k d)$ time to iteratively find a PP and its affiliated sensors on the SPT in subnetwork k. Moreover, the work of finding an approximate shortest tour on the PPs and the data sink can be done in at most $\mathcal{O}(N^2)$ time. Thus, the total time of SPT-DCA is $\sum_{k=1}^{K} \left[\mathcal{O}(N_k^2) + \mathcal{O}(N_k^2 + N_k d)\right] + \mathcal{O}(N^2)$. Hence, in the worst case, the time complexity of SPT-DCA is $\mathcal{O}(N^2 + Nd)$.

V. Distributed Algorithm for BRH-MDC Problem

Given the complete knowledge of sensor distribution, the centralized SPT-DCA algorithm can work well in finding a good data collection tour. However, in practice, such global information is difficult to obtain. In this section, we propose a distributed algorithm searching for suitable sensors as the PPs to achieve better scalability, which follows the same basic idea as the centralized algorithm.

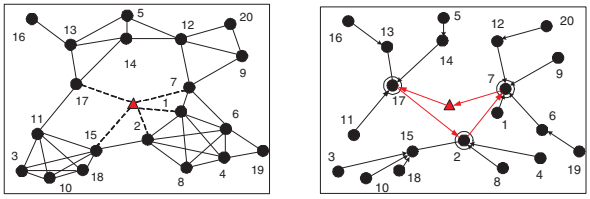
As discussed in the previous section, two factors greatly affect the suitability of a sensor to be a PP. One is the number of sensors within its d-hop range and the other is its distance to the data sink. A sensor that can cover more sensors in its d-hop neighborhood and is close to the data sink will be more favorable to be a PP since it leads to a smaller total number of PPs and a more compact distribution among the PPs. Considering these factors, we propose an algorithm named *priority based PP selection algorithm,* or *PB-PSA* for short. Two parameters are used to prioritize each sensor in the network, which can be easily obtained in a distributed manner. The primary parameter is the number of d-hop neighbors, which are the sensors in its d-hop range. The secondary parameter is the minimum hop count to the data sink. The basic idea of PB-PSA is that each sensor uses the primary parameter to select an initial set of sensors as its preferred PPs, and then uses the secondary parameter to "break ties." A tie in this context means that the preferred PPs of a sensor have the same number of d-hop neighbors.

We now describe PB-PSA in more detail. Its pseudo code is given in Algorithm 2. Before a sensor makes the decision on whether it becomes a PP, d rounds of local information exchange are performed to ensure that each sensor can gather the node information in its d-hop neighborhood. In each round, each sensor locally maintains a structure, named TENTA_PP, based on the information exchange. TENTA_PP is the selected sensor temporarily considered as a preferred

Algorithm 2: Distributed Algorithm: PB-PSA

```
My.TENTA_PP ⟵ My; My.status ⟵ Tentative;
for i = 1 to d do
    Send_Msg (My.TENTA_PP);
    if received packets from all my 1-hop neighbors then
        A ⟵ {a| is a received or own TENTA_PP with
        maximum a.d_Nbrs};
        My.TENTA_PP ⟵ arg min_{a∈A} a.Hop;
if My.TENTA_PP.ID = My.NodeID then
    My.status⟵ PP;
    Send_Declar_Msg (My.NodeID, My.status, 0);
else
    Set my delay timer t =My.Hop×T_s±rand(△);
    while My delay timer is not expired do
        Record source node IDs of the received declaration
        messages;
        Forward the received declaration messages if it has
        been propagated for less than d hops;
    if ever received a declaration message then
        My.PP ⟵ the nearest PP among all the PPs in the
        received declaration messages;
        My.status ⟵ non_PP;
        Send_Join_Msg (NodeID, My.PP);
    else
        My.status ⟵ PP;
        Send_Declar_Msg (My.NodeID, My.status, 0);
```

PP in a particular round by the sensor. TENTA_PP has three sub-domains: TENTA_PP.ID, TENTA_PP.d_Nbrs and TENTA_PP.Hop which denote the node identification, the number of its d-hop neighbors and the minimum hop count of the tentative PP to the data sink, respectively. Initially, each sensor treats itself as its TENTA_PP and labels its status as "Tentative." In a particular round, each sensor first broadcasts the information of its TENTA_PP to its 1-hop neighbors. When it has heard from all the neighbors, the sensor will update its TENTA_PP according to the following rule: among the pool of all the received TENTA_PPs and its own TENTA_PP, choose the one with maximum TENTA_PP.d_Nbrs to set it as its updated TENTA_PP. If there are more than one such TENTA_PP, choose the one with minimum TENTA_PP.Hop from it. Such treatment implies that a sensor having the ability to cover more other sensors and also being close to the data sink has higher chance to be a PP than others. After d rounds of iterations are completed, each sensor is able to tell whether it is the one with the highest priority among its d-hop neighbors. If a sensor finds that its TENTA_PP is still itself after d rounds of information exchange, it will declare to be a PP instantly by sending out a declaration message and changes its status accordingly. This message will then be propagated up to d hops. For other sensors still with "Tentative" status, they will be delayed for a period of time. The delay time for a sensor consists of a major part proportional to its hop count to the data sink plus a random small time duration to differentiate the sensors with the same hop count. During the delay period, a sensor keeps listening and receiving the declaration messages from others. Once its own delay timer expires, a sensor with "Tentative" status will check whether it has received any declaration message. If yes, the sensor will affiliate itself with a nearest PP among those whose declaration messages are received. Otherwise, the sensor itself will declare to be a PP since there is no PP in its d-hop neighborhood for the moment. This way, the sensors with "Tentative" status closer to the data sink will become a PP ahead of others due to the shorter delay, which effectively refrains other sensors with "Tentative" status from declaring to be the unwanted PPs promptly.



(a) Network configuration. (b) Tour along the PPs.

Fig. 4. An example to illustrate the PB-PSA algorithm ($n = 20, d = 2$).

To better understand the PB-PSA, we give an example as shown in Fig. 4 where there is a total of 20 sensors and the data sink is assumed to be located at the center of the area. The connectivity among the sensors and the data sink is shown by the links between neighboring nodes in Fig. 4(a). We set d to 2, which implies that each sensor needs to do two rounds of local data exchange. Every sensor updates its TENTA_PP based on the received information, and the result in each round is listed in Table II. When the iterations are completed, sensors 2, 7 and 17 find that they are the TENTA_PPs for themselves and consequently send out the declaration messages to claim to be the PPs. During the delay period, all other sensors can receive some declaration messages. Thus, there will be no other PPs. In the next step, each sensor with "Tentative" status will choose to be affiliated with a PP among those it has heard from, which will not necessarily be constrained to the current TENTA_PP of the sensor. The final PPs, the sensors' affiliation pattern and the data collection tour are depicted in Fig. 4(b).

Finally, we have the following two properties concerning the complexity of the PB-PSA algorithm.

Property 1: PB-PSA has the worst-case time complexity of $O(Nd)$ per node, where N is the number of sensors.

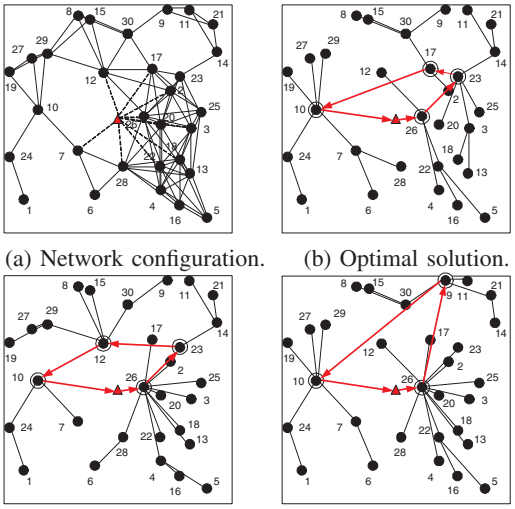
Proof: Each sensor first experiences d rounds of iterations. In each iteration, it takes up to N time for a sensor to gather the information of TENTA_PPs from its one-hop neighbors. Except the sensors that declare themselves to be the PPs once the iterations complete, each of other sensors will delay for a period time. Since the delay time is proportional to the minimum hop count of a sensor to the data sink, in the worst case, it takes $N \times T_s$ time for a sensor to finally determine its status, where T_s is a pre-defined constant time slot. Hence, the total time complexity of the PB-PSA is $O(Nd)$ per node. ■

Property 2: PB-PSA has the worst-case message exchange complexity of $O(N + d)$ per node.

Proof: During the execution of iterations in the PB-PSA, each sensor generates d messages to broadcast its current TENTA_PPs. Once a sensor reaches its final status, either a PP or a regular sensor, it will generate a declaration message to claim to be a PP or a joining message for affiliation. Since each declaration message or joining message will be propagated up to d hops, in the worst case, a sensor may help forward up to $2(N - 1)$ messages. Thus, the total number of messages a sensor handles is at most $d + 1 + 2(N - 1)$, i.e., the message

TABLE II
TWO ROUND UPDATE OF TENTA_PP BY EACH SENSOR IN THE EXAMPLE

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Initialization	TENTA_PP.ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	TENTA_PP.d_Nbrs	10	12	7	8	9	10	12	8	8	7	9	10	7	10	11	5	11	7	7	6
	TENTA_PP.Hop	1	1	2	2	3	2	1	2	2	2	2	2	2	2	1	3	1	2	3	3
Round 1	TENTA_PP.ID	2	2	15	2	12	2	7	2	7	15	15	7	17	17	2	13	17	15	6	12
	TENTA_PP.d_Nbrs	12	12	11	12	10	12	12	12	12	11	11	12	11	11	12	7	11	11	10	10
	TENTA_PP.Hop	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	2	1	1	2	2
Round 2	TENTA_PP.ID	2	2	2	2	7	2	7	2	7	2	2	7	17	7	2	17	17	2	2	7
	TENTA_PP.d_Nbrs	12	12	12	12	12	12	12	12	12	12	12	12	11	12	12	11	11	12	12	12
	TENTA_PP.Hop	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1



(a) Network configuration. (b) Optimal solution.

(c) SPT-DCA. (d) PB-PSA.

Fig. 5. Different solutions for the BRH-MDC problem with d set to 2 in a 30-node network.

exchange complexity of the PB-PSA is $O(N+d)$ per node. ■

VI. PERFORMANCE EVALUATION

In the previous sections, we provide two efficient algorithms for the BRH-MDC problem. To evaluate their performance, in this section, we first implement the ILP formulation given in Section III for a small network and compare it with the proposed algorithms, and then we provide the simulation results of the algorithms for large networks and compare them with other two existing mobile data collection schemes.

A. Comparison with the Optimal Solution

We have solved the ILP formulation of BRH-MDC problem given in Section III for a sensor network with 30 nodes by using CPLEX [19]. We now compare this optimal solution with the results of the proposed algorithms.

As shown in Fig. 5(a), a network with 30 sensors scattered over a 70m $\times$ 70m square area. The connectivity is represented by a solid link between any two neighboring sensors. The static data sink is located at the center of the area and the connectivity between the sink and the sensors is plotted by the dashed links. d is set to 2. The results of different solutions, each of which contains the selected PPs, the relay routing trees for local data aggregation rooted at PPs and the moving tour of the mobile collector, are shown in Fig. 5(b)-(d), respectively. Moreover, the performance comparison is summarized in Table III.

From Fig. 5 and Table III, we can see that the optimal solution for the example achieves the shortest tour length of 94.78m at the expense of 1.27 relay hops on average for local data aggregation. In contrast, SPT-DCA and PB-PSA result in 3% and 24% longer tour length, however, 7.8% and 11% less average relay hop count, respectively. These observations further reveal the intrinsic tradeoff between the tour length and the relay hop count. Since the distributed PB-PSA algorithm commits itself to dampen the number of PPs by prioritizing the sensors with their d-hop neighbors and overhearing the declaration messages propagated in the d-hop proximity, it results in the minimum number of PPs compared with others. Apparently, this will also lead to the most average affiliated sensors for a PP for a given number of sensors. SPT-DCA achieves the same number of PPs as the optimal solution, thus, they have the same number of average affiliated sensors for a PP. However, due to the structural restriction of the shortest path tree among the sensors, the affiliation pattern among the sensors in SPT-DCA is not as uniform as that of the optimal solution, which results in 55% more maximum number of affiliated sensors to a PP.

TABLE III
PERFORMANCE COMPARISON WITH OPTIMAL SOLUTION

	Optimal Solution	SPT-DCA	PB-PSA
Sensors Selected as PPs	10, 17, 23, 26	10, 12, 23, 26	9, 10, 26
Tour Length (m)	94.78	97.56	117.86
Ave. Relay Hop Count	1.27	1.17	1.13
Max No. Affiliated Sensors of a PP	9	14	15
Ave. No. Affiliated Sensors of a PP	7.5	7.5	10

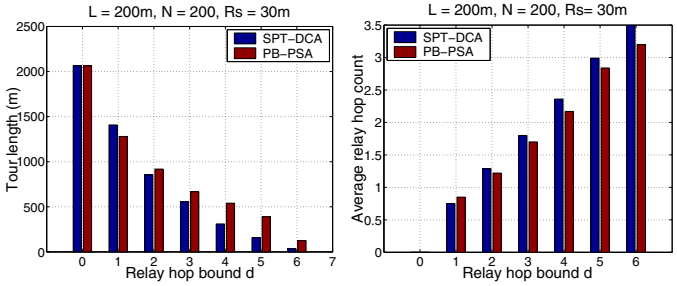
B. Performance of SPT-DCA and PB-PSA

We have also conducted a suite of simulations to evaluate the performance of our proposed algorithms in large sensor networks. In this subsection, we present the simulation results and compare them with other two existing mobile data collection schemes. The first scheme is the single-hop data gathering (SHDG) [14], in which a mobile collector stops at some selected points out of a set of predefined positions to collect data from each sensor such that a single hop data uploading from each sensor to the mobile collector can be guaranteed. Another scheme is the controlled mobile element scheme (CME) [8], where a mobile collector traverses the sensing field along parallel straight tracks and collects data from the sensors nearby with multi-hop relays. For clarity, we list the comparisons between the existing work and our polling-based approach in Table IV.

In the simulation, we consider a generic sensor network with N sensors randomly distributed over an $L \times L$ square area. The data sink is located at the center of the area. The transmission range of a sensor is R_s. Each packet is locally aggregated to a PP within the relay hop bound d before the mobile collector arrives. If not specified otherwise, d is set to 2. We adopt the nearest neighbor (NN) algorithm [23] in our simulation for the TSP problem to determine the moving tour, which lets

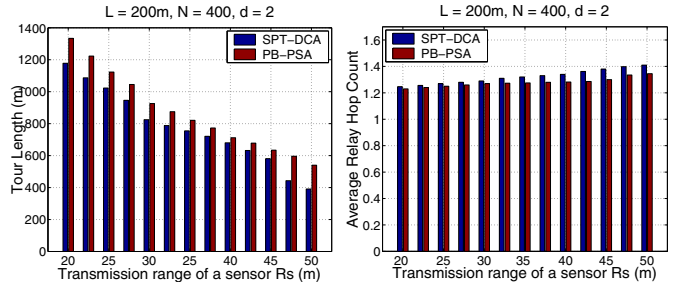
TABLE IV
COMPARISONS AMONG THREE MOBILE DATA COLLECTION SCHEMES

	Polling-based approach (SPT-DCA,PB-PSA)	SHDG	CME
Motion Pattern	Controllable Free to go anywhere	Controllable Free to go anywhere	Uncontrollable With fixed path
Moving Trajectory	Start from the data sink, visit each PP once and go back to data sink	Start from the data sink, visit some points covering the transmission range of each sensor, and finally go back to the data sink	Start from the data sink, go along the parallel straight lines back and forth, and finally go back to the data sink
Relay for Local Data Aggregation	Multi-hop relays with bounded hop count	No local relay	Multi-hop relays without hop count bound
Data Uploading	From specific PPs to the mobile collector when it arrives at the positions of PPs	Each sensor directly uploads data to the mobile collector in a single hop when it arrives within its transmission range	Some sensors close to the tracks upload aggregated packets to the mobile collector



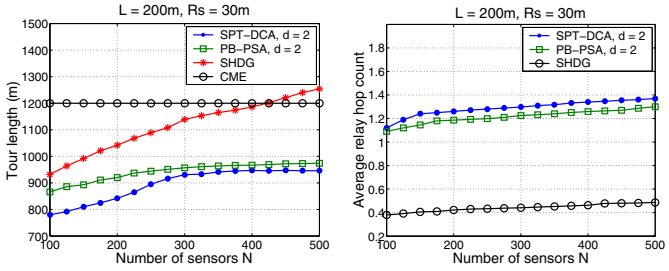
(a) Tour length. (b) Average relay hop count.

Fig. 6. Performance of SPT-DCA and PB-PSA as a function of d.



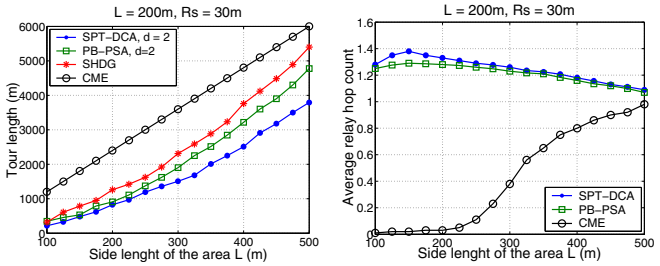
(a) Tour length. (b) Average relay hop count.

Fig. 7. Performance of SPT-DCA and PB-PSA as a function of R_s.



(a) Tour length. (b) Average relay hop count.

Fig. 8. Performance comparison for SPT-DCA, PB-PSA, SHDG and CME as a function of N.



(a) Tour length. (b) Average relay hop count.

Fig. 9. Performance comparison for SPT-DCA, PB-PSA, SHDG and CME as a function of L.

the mobile collector start from the data sink and choose the nearest unvisited PP for the next visit, and finally return to the data sink. Considering the randomness of the network topology, each performance point in the figures is the average of the results in 500 simulation experiments.

Fig. 6 plots the performance of SPT-DCA and PB-PSA as a function of d, in terms of the tour length and the average relay hop count for local data aggregation. When d is set to zero, it means that the mobile collector will visit the sensors one by one for the data collection. N and L are 200 and 200m, respectively. R_s is equal to 30m. From the figure, we can see that as d becomes larger, the tour length is evidently shortened and the average relay hop count gradually increases in both algorithms. Furthermore, in most cases in Fig. 6(a), SPT-DCA always outperforms PB-PSA with about 39% shorter tour length on average, and such superiority becomes even more noticeable as d increases. There are two reasons for this. First, since sensors are densely deployed in the scenario under consideration, there are quite a few sensors scattered around the sink. This provides a good opportunity to build a SPT rooted at a sensor close to the sink in the centralized SPT-DCA algorithm. As a result, with the increase of d, the selected PPs become convergent towards the data sink and also get closer to each other. Another reason is that in the distributed PB-PSA algorithm, though the number of total PPs drops as d increases, some sensors who are still in "Tentative" status after d rounds of iterations tend to claim themselves to be the PPs with higher probability. Some of them may be located far from other PPs such that the tour length in PB-PSA is somewhat longer than that of SPT-DCA. We can also observe in Fig. 6(b) that PB-PSA results in a smaller average relay hop count as compared to SPT-DCA since the PPs in PB-PSA are distributed in a more relaxed pattern.

Fig. 7 shows the performance of SPT-DCA and PB-PSA when R_s is varied from 20m to 50m. Clearly, more sensors will become neighbors to each other as R_s increases. As a result, the situation that most of the PPs are far away from the sink can be avoided and the number of PPs can also be effectively reduced as each PP is able to link up more sensors. Therefore, the tour length will be correspondingly shortened as R_s becomes large. For instance, as shown in Fig. 7(a), when R_s is 20m, the tour length of SPT-DCA and PB-PSA is 1178m and 1334m, respectively. However, when R_s increases to 45m, their tour length drops to 591m and 634m, respectively. Moreover, the average relay hop counts for both algorithms slightly increase with R_s since more sensors will be affiliated with the same PP with a larger hop count under the constraint of the relay hop bound.

Fig. 8 depicts the performance of the proposed algorithms as a function of N, and compares it with SHDG and CME. L

is set to 200m and N is varied from 100 to 500 to represent different node density. R_s is still fixed to 30m. The relay hop bound d is set to 2 for the proposed algorithms. For SHDG, we assume that each predefined position where the collector could stop for data collection is on a grid, which is apart from its adjacent positions in horizontal and vertical directions with the same distance of 20m. Also, in the CME scheme, we assume that the parallel straight tracks traversing the field are 100m apart from each other. The track in the middle goes through the center of the field. The mobile collector can go along the area border to change onto other tracks. Both SHDG and CME schemes are implemented in a centralized fashion. We can observe in Fig. 8(a) that as N increases, the tour length of SPT-DCA and PB-PSA first gradually increases and then stabilizes when N becomes sufficiently large. This is because that when sensors become more densely scattered, they will have higher probability to be affiliated with a PP close to the sink. Hence, the further increase on the number of sensors will have little impact on the selection of the preferred PPs. In contrast, the tour length of SHDG continuously increases with N, which is around 33% longer than that of SPT-DCA. Since the mobile collector travels along the fixed tracks in a given area, the tour length will stay constant for CME. In addition, Fig. 8(b) shows the average relay hop counts for SPT-DCA, PB-PSA and CME. Since a sensor always directly uploads data to the mobile collector in SHDG, there is no local relay required. Thus, we did not include it in the figure. We can see that the average relay hop count slightly increases for each scheme as N becomes large. SPT-DCA and PB-PSA result in more relay hops for local data aggregation compared to CME, which is the cost to achieve a shorter data collection tour.

Fig. 9 further plots the tour length and the average relay hop counts obtained with different schemes when L is varied from 100m to 500m. N is set to 400 and R_s is 30m. We fix 5 parallel straight tracks with the same interval distance in CME scheme, which traverse the field with the outermost two tracks on the border of the area. All other settings are kept unchanged as in the previous set of simulations. From Fig. 9(a), we can see that as L increases, the tour length of all the schemes becomes longer. This is reasonable since sensors become more sparsely distributed as L becomes larger. The mobile collector needs to go further away from the sink and visit more positions to collect data from all the sensors. Also, with the increase of the field area, the fix track traversing the field in CME scheme also becomes longer than the case with a smaller L. However, our proposed algorithms always outperform others, with up to 38% and 80% shorter tour length compared with SHDG and CME, respectively. This attributes to the effort in SPT-DCA and PB-PSA algorithms on minimizing the tour length by fully utilizing the available relays for local data aggregation. Consequently, as the expense, in Fig. 9(b) the average relay hop counts for SPT-DCA and PB-PSA algorithms are higher than that of CME. However, such a gap in the relay hop count quickly shrinks as L increases.

VII. Conclusions

In this paper, we have studied mobile data gathering in wireless sensor networks by exploring the tradeoff between the relay hop count of sensors for local data aggregation and the tour length of the mobile collector. We have proposed a polling-based scheme and formulated it into the BRH-MDC problem. We then presented two efficient algorithms to give practically good solutions. Extensive simulations have been carried out to validate the efficiency of the scheme. The results demonstrate that the proposed algorithms can greatly shorten the data collection tour length with a small relay hop bound, and achieve 38% and 80% improvement on the tour length compared to SHDG and CME schemes, respectively.

References

[1] W. C. Cheng, C. Chou, L. Golubchik, S. Khuller and Y. C. Wan, "A coordinated data collection approach: design, evaluation, and comparison," *IEEE JSAC*, vol. 22, no. 10, Dec. 2004.

[2] A. Manjeshwar and D. P. Agrawal, "Teen: a routing protocol for enhanced efficiency in wireless sensor networks," *IEEE IPDPS*, April 2001.

[3] A. Scaglione and S. D. Servetto, "On the interdependence of routing and data compression in multi-hop sensor networks," *ACM MobiCom*, 2002.

[4] X. Tang and J. Xu, "Adaptive data collection strategies for lifetime-constrained wireless sensor networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 19, no. 6, June 2008.

[5] B. Gedik, L. Liu and P. S. Yu, "ASAP: an adaptive sampling approach to data collection in sensor networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 12, PP. 1766-1783, Dec. 2007.

[6] C. Liu, K. Wu and J. Pei, "An energy-efficient data collection framework for wireless sensor networks by exploiting spatiotemporal correlation," *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 7, pp. 1010-1023, Jul. 2007.

[7] R. Shah, S. Roy, S. Jain and W. Brunette, "Data MULEs: modeling a three-tier architecture for sparse sensor networks," *Elsevier Ad Hoc Networks Journal*, vol. 1, pp. 215-233, Sept. 2003.

[8] D. Jea, A. A. Somasundara and M. B. Srivastava, "Multiple controlled mobile elements (data mules) for data collection in sensor networks," *IEEE/ACM DCOSS*, June 2005.

[9] 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," *in Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2002.

[10] T. Small and Z. Haas, "The shared wireless infostation model - a new ad hoc networking paradigm (or where there is a whale, there is a way)," *ACM MobiHoc* 2003.

[11] M. A. Batalin, M. Rahimi, Y. Yu, D. Liu, A. Kansal, G. S. Sukhatme, W. J. Kaiser, M. Hansen, G. J. Pottie, M. Srivastava and D. Estrin, "Call and response: experiments in sampling the environment," *ACM SenSys*, 2004.

[12] R. Pon, M. A. Batalin, J. Gordon, A. Kansal, D. Liu, M. Rahimi, L. Shirachi, Y. Yu, M. Hansen, W. J. Kaiser, M. Srivastava, G. Sukhatme and D. Estrin, "Networked infomechanical systems: a mobile embedded networked sensor platform," *ACM/IEEE IPSN*, 2005.

[13] A. A. Somasundara, A. Ramamoorthy and M. B. Srivastava, "Mobile element scheduling with dynamic deadlines," *IEEE Trans. Mobile Computing*, vol. 6, no. 4, 2007.

[14] M. Ma and Y. Yang, "Data gathering in wireless sensor networks with mobile collectors," *IEEE IPDPS*, 2008.

[15] M. Ma and Y. Yang, "SenCar: an energy-efficient data gathering mechanism for large-scale multihop sensor networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 10, Oct. 2007.

[16] J. Luo and J. P. Hubaux, "Joint mobility and routing for lifetime elongation in wireless sensor networks," *IEEE INFOCOM*, 2005.

[17] M. Zhao, M. Ma and Y. Yang, "Mobile data gathering with space-division multiple access in wireless sensor networks," *IEEE INFOCOM*, 2008.

[18] G. Xing, T. Wang, W. Jia and M. Li, "Rendezvous design algorithm for wireless sensor networks with a mobile base station," *ACM Mobihoc*, Hong Kong, 2008.

[19] *CPLEX package,* http://www.ilog.com/products/cplex/.

[20] *AMPL package,* http://www.ampl.com/.

[21] B. Sheng, Q. Li and W. Mao, "Data storage placement in sensor networks," *ACM Mobihoc*, Florence, Italy, May 2006.

[22] B. Gavish, "Formulations and algorithms for the capacitated minimal directed tree problem," *Journal of ACM*, vol. 30, pp. 118-132, 1983.

[23] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to algorithms*, 2nd edition, The MIT Press, 2001.