

Article

HiCoDG: A Hierarchical Data-Gathering Scheme Using Cooperative Multiple Mobile Elements †

Duc Van Le, Hoon Oh and Seokhoon Yoon *

Department of Electrical and Computer Engineering, University of Ulsan, Ulsan 680-749, Korea;
E-Mails: anhduc.mta@gmail.com (D.V.L.); hoonoh@ulsan.ac.kr (H.O.)

† This paper is an extended version of our paper published in "A Novel Hierarchical Cooperative Data Gathering Architecture Using Multiple Mobile Elements. In Proceedings of the 2014 6th International Conference on Ubiquitous and Future Networks (ICUFN), Shanghai, China, 8–11 July 2014; pp. 522–527".

* Author to whom correspondence should be addressed; E-Mail: seokhoonyoon@ulsan.ac.kr;
Tel.: +82-52-259-1403.

External Editor: Leonhard Reindl

Received: 6 November 2014; in revised form: 11 December 2014 / Accepted: 12 December 2014 /
Published: 17 December 2014

Abstract: In this paper, we study mobile element (ME)-based data-gathering schemes in wireless sensor networks. Due to the physical speed limits of mobile elements, the existing data-gathering schemes that use mobile elements can suffer from high data-gathering latency. In order to address this problem, this paper proposes a new hierarchical and cooperative data-gathering (HiCoDG) scheme that enables multiple mobile elements to cooperate with each other to collect and relay data. In HiCoDG, two types of mobile elements are used: the mobile collector (MC) and the mobile relay (MR). MCs collect data from sensors and forward them to the MR, which will deliver them to the sink. In this work, we also formulated an integer linear programming (ILP) optimization problem to find the optimal trajectories for MCs and the MR, such that the traveling distance of MEs is minimized. Two variants of HiCoDG, intermediate station (IS)-based and cooperative movement scheduling (CMS)-based, are proposed to facilitate cooperative data forwarding from MCs to the MR. An analytical model for estimating the average data-gathering latency in HiCoDG was also designed. Simulations were performed to compare the performance of the IS and CMS variants, as well as a multiple traveling salesman problem (mTSP)-based approach.

The simulation results show that HiCoDG outperforms mTSP in terms of latency. The results also show that CMS can achieve the lowest latency with low energy consumption.

Keywords: mobile data gathering; multiple mobile elements; cooperative data gathering; optimal trajectories

1. Introduction

Wireless sensor networks (WSNs) that consist of a number of low-cost, low-power stationary sensors have been widely used in various applications such as environmental monitoring, industrial sensing, and battlefield surveillance [1]. In WSNs, data gathering from distributed sensors deployed over a large area is one of the most important tasks. Data gathering in WSNs has traditionally been performed via multi-hop data forwarding to the sink. However, it is known that multi-hop forwarding leads to the energy-hole problem, where the energy of sensors near the sink is depleted quickly since they forward more data packets than sensors distant from the sink [2]. As a result, the energy-hole problem makes the network disconnected and decreases overall network lifetime.

In order to avoid the energy-hole problem, mobile data-gathering schemes have been studied [3–21]. In such schemes, mobile elements (MEs) (e.g., autonomous robots) are used to collect the data from sensors and bring the data to the sink. By using MEs, not only can the energy-hole effect be avoided but data gathering in a sparse or disconnected network also becomes possible because MEs can travel and directly collect the data from sensors. The drawback to this approach is long data-gathering latency since data delivery relies on the physical movements of MEs. In addition, high energy consumption by MEs is another issue to be addressed.

In order to achieve low data-gathering latency with minimal energy consumption, this paper proposes a new hierarchical and cooperative data-gathering (HiCoDG) scheme where two different types of MEs cooperate to collect and relay data from sensors to the sink. One type of ME is the mobile collector (MC), which collects data from sensors. Another type is the mobile relay (MR), which gathers data from MCs and delivers them to the sink. In HiCoDG, mobile collectors do not need to visit the sink, which can result in a significant improvement in terms of latency and energy consumption.

More specifically, in HiCoDG, sensors are first organized into one-hop clusters. The positions of clusterheads are considered *points of interest* in the network, which are partitioned into several groups with respect to geographic position. In each group, an MC is scheduled to periodically visit the points of interest, where it collects data from sensors within the cluster via one-hop communications. Then, MCs forward the data to the MR, which periodically travels from the sink to visit points of interest called meeting points (MPs) to receive data from MCs. Finally, the MR delivers the data when it returns to the sink.

In order to find optimum trajectories for MCs and the MR to minimize the total traveling distance, we define an optimization problem, named local data gathering with global relay (LDG-GR). Then, an integer linear programming (ILP) problem is formulated to find the solution to LDG-GR.

In this paper, depending on the way data are relayed from MCs to the MR, two variants of HiCoDG are considered: intermediate station (IS)-based and cooperative movement scheduling (CMS)-based schemes. In the IS scheme, each MC *drops off* data to an MP, which will store the data until the MR visits and *picks up* the data. In this case, special hardware (e.g., high-capacity data storage and batteries) is required for MPs. On the other hand, in the CMS scheme, the optimal movement speeds of MCs and the MR are predetermined, such that MCs and the MR can meet just in time during their journey to directly forward data using a wireless channel.

In addition, an analytical model was designed to estimate the average data-gathering latency in HiCoDG. The model can be useful for estimating whether MEs with a certain movement speed are able to achieve the required latency before they are actually deployed.

Extensive simulation and analysis were conducted to evaluate the performance of the two variants of HiCoDG. Those variants were also compared with multiple traveling salesman problem (mTSP)-based approach. The simulation results show that HiCoDG outperforms the mTSP-based approach in terms of latency and energy consumption. The simulation results also indicate that the CMS scheme can achieve the lowest latency.

The rest of the paper is organized as follows. In Section 2, we present related work and compare it to our algorithm. Section 3 presents the proposed scheme and algorithms in detail. In Section 4, an analytical model for estimating latency is described. Section 5 presents the simulation setup and performance analysis. Finally, Section 6 concludes the paper.

2. Related Work

In this section, we present an overview of previous studies on mobile data gathering, and compare those studies with our proposed algorithm.

There are a lot of recent studies on data gathering in WSNs using mobile elements. The majority of these studies considered the use of a single ME [5–12].

In these studies, an ME is scheduled to periodically visit sensors in the network and collect the data, then deliver them to the sink. For example, two studies [5,6] focused on the problem of finding the optimal trajectory and movement schedule for an ME, with the objective of maximizing network lifetime. More specifically, Wang *et al.* [5] assumed that the ME only visits some specific nodes in the network and stays for a given sojourn time to collect the data through multi-hop data transmissions. An linear programming (LP) optimization formulation was proposed to jointly determine the specific nodes the ME should visit and its sojourn time at each node, so that the network lifetime is maximized.

Gu *et al.* [8] and Somasundara *et al.* [9] proposed movement scheduling algorithms to schedule an ME such that the packet loss at sensors due to buffer overflow can be avoided. For example, Gu *et al.* [8] first organized the sensors in the network into a number of groups based on their data generation rates and locations. Then, the ME was scheduled to visit sensors in each group at an adequate frequency so the buffer overflow at the sensors is reduced.

In addition, Kumar *et al.* [11] and Sugihara and Gupta [12] focused on the problem of reducing data-gathering latency. Kumar *et al.* [11] proposed a single mobile collector-based data collection architecture that lowers latency by considering sensor clustering and long-range wireless

communications. Sugihara and Gupta [12] proposed a heuristic scheduling algorithm that considers both sensor locations and time constraints to minimize latency in an 1D case problem.

Those studies differ from our work in that they use only one ME to collect the data from sensors. The potential drawback of this approach is less scalability. When the number of sensors or the network area increases, the path length the ME must travel increases accordingly. The ME will take a longer time to finish one round, which results in high data-gathering latency. Therefore, a single ME may not be sufficient for certain applications that require low latency in a large-scale WSN. In our work, we consider the use of multiple MEs for gathering data to reduce latency.

There have also been several studies that use multiple MEs for data gathering in WSNs [14–20]. For example, Zhao *et al.* [15] proposed a data-gathering scheme using multiple MEs that are controlled to move along parallel straight lines through the area of interest and collect data from sensors using multi-hop transmission. This scheme can work well in a large-scale WSN where the sensors are uniformly distributed. However, if not all of the sensors in the network are connected using a wireless link, the MEs may not cover some of them because they traverse only straight lines.

Kim *et al.* [16] defined two combinatorial optimization problems to find the trajectories, with the objective of minimizing data-gathering latency: the k -traveling salesperson problem with neighborhood (k -TSPN) and the k -rooted path cover problem with neighborhood (k -PCPN). They also proposed constant-factor approximation algorithms for those problems.

Those studies on multiple ME-based mobile data gathering differ from our work in that they schedule each ME to visit a part of the sensing field to collect data and all MEs need to return to the sink. This may lead to high data-gathering latency in a large-scale WSN where a number of sensors are deployed over a large area distant from the sink. In our scheme, sensors are first organized into groups, and a mobile collector (MC) that does not need to return to the sink is assigned to each group to collect data. The mobile relay is scheduled to periodically visit each group, receive the data from the MCs, and bring the data to the sink. Moreover, in our work, a cooperative movement scheduling algorithm is proposed to control the movements of the MCs and the MR in order to reduce the data-gathering latency in the network.

In addition, Aslanyan *et al.* [21] considered the use of multiple MEs that move with a random mobility pattern in the area of interest to collect the data, and deliver the collected data to the sink when they are close enough to the sink to ensure a direct transmission. Due to the random mobility, the data-gathering latency may not be estimated. In contrast, in our work, MEs are scheduled to move in cooperation to collect and relay the data to the sink. As a result, the average latency in our proposed schemes can be estimated, which is useful to determine whether or not the data-gathering scheme using MEs can satisfy the latency requirement before MEs are actually deployed.

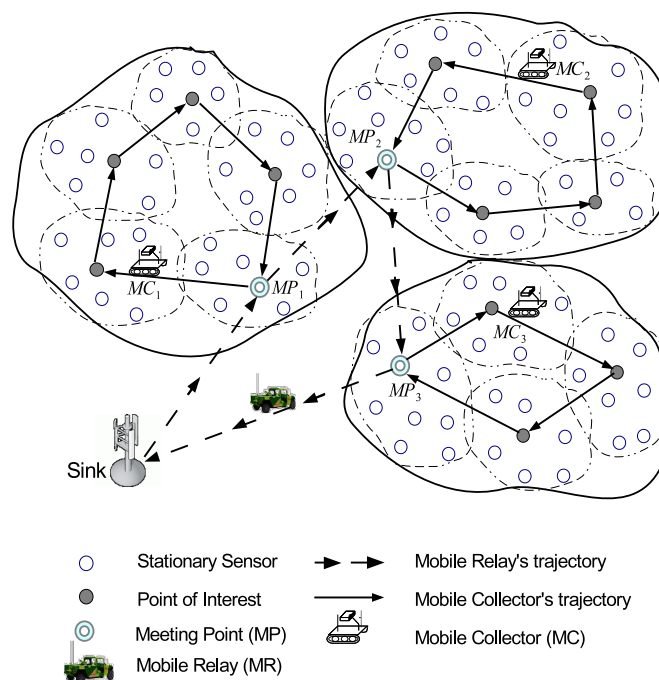
3. Hierarchical and Cooperative Data Gathering (HiCoDG) Scheme

In this section, we present the proposed HiCoDG scheme in detail. The considered network consists of a number of stationary sensors that are deployed over an area of interest. Each sensor periodically generates data packets and saves them in its buffer. It is assumed that the positions of sensors are known a priori.

We consider the use of multiple mobile elements (e.g., autonomous robots equipped with RF transceivers) to gather data from sensors and to deliver the data to the sink located at a given position. More specifically, multiple mobile elements including mobile collectors and a mobile relay are deployed to cooperatively collect and relay the data.

Figure 1 illustrates the overview of the proposed data-gathering scheme. In Figure 1, sensors are partitioned into three groups, each of which consists of several one-hop clusters. In each group, an MC moves along a closed path that consists of points of interest at which an MC collects data from sensors via one-hop wireless communications. The path of the MC begins and ends at an meeting point, which is one of the points of interest. The MC forwards the collected data to the MR via the MP or directly. The MR starts moving from the position of the sink, and visits each group at an MP and receives the data from the MP or MCs. It is assumed that MCs can be replaced with spare MCs in case they can not perform the mission due to power depletion or malfunction. In this paper, it is also assumed that collision between MEs can be avoided using existing collision avoidance algorithms [22,23], even if their movement paths intersect.

Figure 1. An overview of HiCoDG.



From here, we first describe the clustering algorithms that are used to partition sensors into several groups. Then, the optimization problem formulation to find the optimal trajectories for MEs is presented. Finally, two cooperative data relay schemes are described.

3.1. Clustering and Grouping Algorithms

In HiCoDG, sensor nodes are partitioned into k groups where k is the number of MCs used in the network. To form the groups, sensor nodes are first organized into one-hop clusters using a highest degree-based clustering algorithm [24]. More specifically, a node that has the maximum number of neighboring nodes is selected as the first clusterhead, and its neighboring nodes join the cluster. Among

nodes that do not belong to the first cluster, another node with the maximum number of neighbors is selected to form the second cluster, and so on. This process is repeated until every node belongs to a cluster.

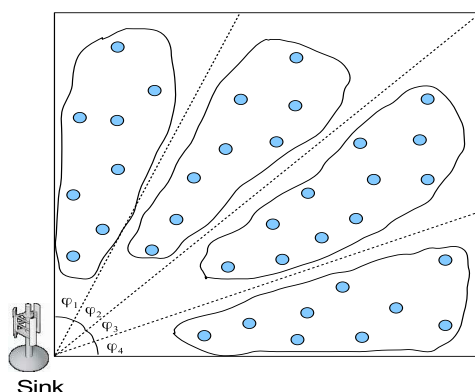
Then, the positions of clusterheads, which are regarded as points of interest (or points), are divided into k groups, each of which has an MC assigned to it. In this paper, we consider the use of two methods for partitioning the points into k groups: a K-means-based grouping algorithm and a sink position-based grouping (SPG) algorithm.

K-means-based grouping algorithm: We adopt K-means like a minimum mean distance algorithm (or K-means algorithm) [25], which uses the minimum mean distance between points as an objective function of grouping. More specifically, in the K-means algorithm, the points are divided into groups such that the mean distance between the group-head and members in each group is minimized.

Sink position-based grouping (SPG) algorithm: In order to achieve low data-gathering latency, it is desirable for the MR to travel a short tour so that it can deliver the data from the MCs to the sink in a short amount of time. However, when the K-means-based algorithm is used, some groups might be located far from the sink, which results in a long tour for the MR. Therefore, we propose the SPG algorithm. The main idea behind SPG is to divide the points into groups such that every group includes one point that is close to the sink. More specifically, in SPG, the area is first divided into k sub-parts such that the sink position is included in every sub-part. The points located in the same sub-part belong to the same group.

Figure 2 illustrates grouping when SPG is applied. In Figure 2, the sink is located in the left corner of a rectangular area. Assume there are four MCs, *i.e.*, $k = 4$. Then, the area of interest is divided into four sub-parts represented by the dotted lines starting from the sink position such that $\varphi_1 = \varphi_2 = \varphi_3 = \varphi_4$, as shown in Figure 2.

Figure 2. An example of sink position-based grouping.



3.2. Optimal Trajectory Formulation

Now, we consider the problem of finding the optimal trajectories for the MR and MCs that minimize latency and energy consumption in the network. Note that there have been a lot of studies on finding the optimal trajectory over visiting points in order to minimize total distance, such as the traveling salesman problem (TSP) [26], multiple salesman problem [27], and vehicle routing problem [28].

However, those studies cannot be used in our data-gathering architecture because they do not consider different types of MEs, the hierarchical architecture of MEs' movements, and the need for data exchange

among MEs. More specifically, in our work, two types of mobile elements (MC and MR) are used. Each MC collects data from sensors in its group and forwards the data to the MR at a meeting point, which is one of the sensor positions in the group. The MR receives data from MCs at meeting points, and delivers the data to the sink.

Therefore, we define a new optimization problem.

Problem 1. Local data gathering with global relay (LDG-GR): Assume a network consists of a set of points $V = \{0, 1 \dots N\}$, where the element 0 represents the sink. V is partitioned into $k + 1$ subsets, $V_0, V_1 \dots V_k$ where $V_0 = \{0\}$. Using k MCs, and one MR, find the minimum total path length of the MCs and the MR. MC_i ($1 \leq i \leq k$) must visit every point in V_i once and come back to its starting point. The MR must visit at least one point in each V_i and come back to its starting position *i.e.*, point 0 in V_0 .

Proposition 1. LDG-GR is NP-complete.

Proof. It can be readily shown that a Euclidean TSP [29] is polynomial-time reducible to LDG-GR, *i.e.*, Euclidean TSP \leq_p LDG-GR. Suppose that a Euclidean TSP instance consists of a set of k “cities” c_1, \dots, c_k , a distance $d(i, j)$ between every pair of cities c_i and c_j ($1 \leq i, j \leq k$), and an integer D . The corresponding LDG-GR instance has k subsets, V_1, \dots, V_k , each of which has only one element *i.e.*, $|V_i| = 1$. Let V_i have a point i as an element that corresponds to c_i , and define the distance between a pair of points in $\bigcup_i^k V_i$ the same as $d(i, j)$. Let point 0 be the sink. The instance of LDG-GR can be constructed in polynomial time. Note that in the constructed LDG-GR instance, only the MR has a traveling distance since every subset has only one point. Then, it is clear that if the length of the tour is D in Euclidean TSP, the total path length in LDG-GR becomes D . Conversely, if the total path length in LDG-GR is D , the length of the tour becomes D in the Euclidean TSP. LDG-GR is in NP, since a guessed path, obtained by using a non-deterministic algorithm, can be checked in polynomial time as to whether it has a total length of D and if every point is visited. \square

In this paper, we propose ILP formulation for jointly finding the optimal trajectories for both MCs and the MR. We model the network as a complete directed graph $G = (V, E)$ where $E = \{(i, j) : i, j \in V, i \neq j\}$ is the set of arcs. Let c_{ij} denote the travel cost (Euclidean distance) from point i to point j . We also define x_{ij} as a binary variable that becomes 1 if $arc(i, j)$ is on the trajectories of the MCs, but becomes 0 otherwise. Similarly, a binary variable y_{ij} represents whether or not $arc(i, j)$ is on the trajectory of the MR. We denote n_i as the number of points in subset V_i *i.e.*, $|V_i| = n_i$. Note that $n_0 = 1$. Also, u_i is defined as the number of points that have been visited by the MCs and the MR up to point i . We also define $p_i^{org} \in V_i$ as the origin point of the trajectory solution of MC_i . Note that p_i^{org} can be an arbitrary point in V_i and can be chosen a priori.

Then, the ILP problem can be formulated as follows:

$$\text{minimize} \left(\sum_{i=1}^N \sum_{j=1}^N c_{ij} x_{ij} + \sum_{i=0}^N \sum_{j=0}^N c_{ij} y_{ij} \right) \quad (1)$$

subject to:

$$\sum_{i=1, i \neq j}^N x_{ij} = 1, \quad j = 1 \dots N \quad (2)$$

$$\sum_{j=1, j \neq i}^N x_{ij} = 1, \quad i = 1 \dots N \quad (3)$$

$$\sum_{i \in V_l} \sum_{j \in V \setminus V_l \cup V_0} x_{ij} = 0, \quad l = 1 \dots k \quad (4)$$

$$u_{p_l^{org}} = 1, \quad l = 1 \dots k \quad (5)$$

$$u_i - u_j + 1 \leq (n_l - 1)(1 - x_{ij}) \\ i, j \in V_l \setminus \{p_l^{org}\}, l = 1 \dots k \quad (6)$$

$$\sum_{i \in V_l} \sum_{j \in V \setminus V_l} y_{ij} = 1, \quad l = 1 \dots k \quad (7)$$

$$\sum_{i \in V \setminus V_l} y_{ij} = \sum_{i \in V \setminus V_l} y_{ji}, \quad j \in V_l, l = 1 \dots k \quad (8)$$

$$u_0 = 1 \quad (9)$$

$$u_i - u_j + 1 \leq (N - 1)(1 - y_{ij}), \quad i, j \in V \setminus V_0 \quad (10)$$

The objective function in Equation (1) is for minimizing the total distance traveled by both the MCs and the MR. Equations (2)–(6) define constraints on the optimal trajectories of the MCs, whereas Equations (7)–(10) represent constraints on the optimal trajectory of the MR. More specifically, constraints Equations (2) and (3) ensure that every point except 0 is visited exactly once by only one MC. Constraint Equation (4) states that MCs do not travel between two points that belong to two different subsets (*i.e.*, groups). Constraints Equations (5) and (6) eliminate the subtours in the trajectories of the MCs in a way similar to the MTZ subtour elimination of the TSP [26].

On the other hand, constraint Equation (7) ensures that the MR visits each group exactly once at one point. Note that among all points of interest, the point that will be visited by the MR is called a meeting point (MP). Constraint Equation (8) ensures that the MR enters and leaves each group at the same MP. Note that MP_i may be different from p_i^{org} which was arbitrarily chosen to find the optimal trajectories. Constraints Equations (9) and (10) eliminate the subtour in the trajectory of MR.

Note that, in case there are constraints (e.g., obstacles) on the link between two points, the link cost value can be set to a high value (or even an infinite number). Due to the objective function that minimizes the path cost, such high cost links will be avoided in the solution. In other words, the ILP formulation can find the solution even when there are physical constraints on the links.

3.3. Cooperative Data Relay Scheme

Along the optimal trajectories obtained from the ILP formulation, each MC periodically performs data gathering in its group. On each tour, an MC starts from an MP, visits all points in the group, and returns to the MP. The MR also periodically travels to receive data from the MCs at the MPs. In order for the MR to relay the data from MCs to the sink, we consider two schemes: IS-based data relay and

cooperative movement scheduling CMS-based data relay. In the IS scheme, MPs require more powerful HW than regular points of interest, while CMS does not require special HW.

(1) IS-based data relay

In this scheme, it is assumed that the points that are selected as MPs are equipped with high-capacity data storage and high-capacity batteries. During its journey, an MC sends its collected data to its MP when they can establish a direct wireless communications channel. Then, the MP stores the data and forwards them when the MR visits. The MR receives data from all MPs and returns to the sink.

Note that, in the IS scheme, MCs and the MR move independently. Due to lack of cooperation among nodes, the MR can visit an MP before an MC sends data to that MP. In that case, the MR cannot receive any data from the MP until the next visit. As a result, the MR wastes energy, and data-gathering latency may increase. Moreover, the need for data storage and high-capacity batteries at MPs increases system cost.

In order to address these problems, we propose a CMS-based data relay algorithm, where the movements of MCs and the MR are scheduled such that the MR can directly receive the collected data from MCs while the MR and MCs are traveling.

(2) CMS-based data relay

The main idea behind CMS is that the movement speeds of MCs and the MR are controlled so that MCs can meet the MR at MPs during the periodic movements and can forward data to the MR using a direct wireless channel. Let L_i ($i = 0 \dots k$) denote the path (or tour) length of each ME (MC or MR), where L_0 is the tour length of the MR and L_i ($1 \leq i \leq k$) the tour length of MC_i . We also define s_{max} as the maximum movement speed of MEs.

Recall that the path of the MR consists of $k + 1$ meeting points, MP_i ($0 \leq i \leq k$) where MP_0 is the sink's position. Let τ denote the time duration in which the MR stays to receive the data from MC_i at MP_i ($i \neq 0$), or to send the data to the sink at MP_0 . We also define T_m as the time it takes for the MR to finish one tour, which includes moving time and sojourn times at MPs. Then, the value of T_m can be calculated as

$$T_m = \frac{L_{max}}{s_{max}} + (k + 1)\tau \quad (11)$$

where $L_{max} = \max(L_0, L_1, \dots, L_k)$.

The speeds of MEs are determined such that the MR can periodically meet MC_i at every T_m . Using T_m and the tour length of each ME, the speed of each ME can be calculated as follows:

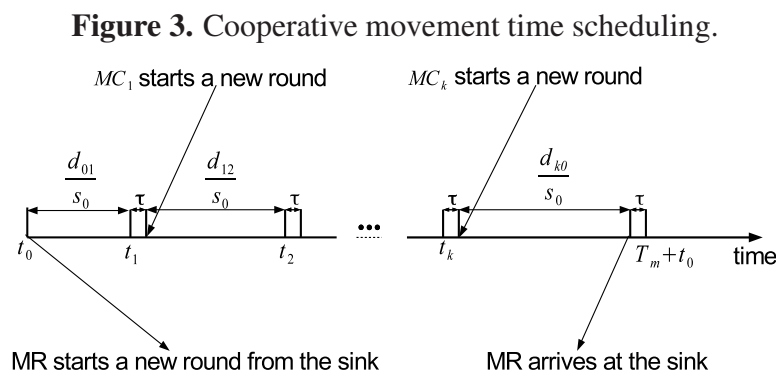
$$\begin{cases} s_0 = \frac{L_0}{T_m - (k+1)\tau} \\ s_i = \frac{L_i}{T_m - \tau}, \quad 1 \leq i \leq k \end{cases} \quad (12)$$

where s_0 is the speed of the MR and s_i is the speed of MC_i . Note that s_0 and s_i are equal to or less than s_{max} .

Also, denote t_0 as the time at which the MR leaves the sink and d_{ij} as the distance from MP_i to MP_j . Then, during the same tour, the MR meets MC_i at time $t_i (i \neq 0)$, which can be calculated as

$$t_i = t_0 + (i - 1)\tau + \frac{1}{s_0} \sum_{j=1}^i d_{(j-1)j} \tag{13}$$

In CMS, the MR starts a new tour at the sink at every T_m as shown in Figure 3. The MR begins its journey from MP_0 , and visits MP_i , where it meets MC_i and stays for τ seconds to receive data. Then, the MR returns to MP_0 , stays for τ seconds to deliver data to the sink, and then starts another journey. Note that it is assumed that the initial positions of $MC_i (1 \leq i \leq k)$ are also determined such that MC_i can meet the MR at t_i . After sending the data to MR, MC_i starts new round at $t_i + \tau$ as shown in Figure 3. By using the speed from Equation (12), MC_i will periodically meet the MR every T_m seconds during future rounds.



4. Analytical Model for Estimating Data Gathering Latency in HiCoDG

In wireless sensor networks, the application often has a specific requirement for data-gathering latency. Therefore, it is important and necessary to estimate whether or not the mobile data-gathering system using MEs can satisfy the latency requirement before the MEs are actually deployed. In order to address this issue, in this section, we discuss an analytical model to estimate data-gathering latency in the proposed HiCoDG.

We first present some general assumptions for analysis. Then, an analytical model to estimate data-gathering latency is described. Finally, we verify the model by comparison with simulation results.

4.1. General Assumptions

We consider a network consisting of $k + 1$ MEs, including k MCs and one MR. The points of interest (or points) in the network are partitioned into k groups $G_i (i = 1, \dots, k)$. MC_i visits and collects data at points in group G_i . Recall that L_0 and $L_i (i = 1, \dots, k)$ denote the path lengths of the MR and MC_i , respectively. Similarly, we define $T_i (i = 0, \dots, k)$ as the time that it takes for MR and MC_i to finish one tour, respectively. In addition, recall that s_0 and s_i are movement speeds of the MR and MC_i , respectively. Without loss of generality, in the model, we do not consider latency of wireless

communications since the network transmission speed is much higher than the movement speed of MEs. Thus, we can have

$$\begin{cases} T_0 = \frac{L_0}{s_0} \\ T_i = \frac{L_i}{s_i} \quad (i = 1, \dots, k) \end{cases} \quad (14)$$

It is assumed that at every point on the paths of the MCs, a data packet is periodically generated at the rate of R (in packets per second). The packet is stored in the buffer of the point until an MC collects it. We also assume that on the path of each MEs (the MR or MCs), the points are randomly placed in an uniform distribution.

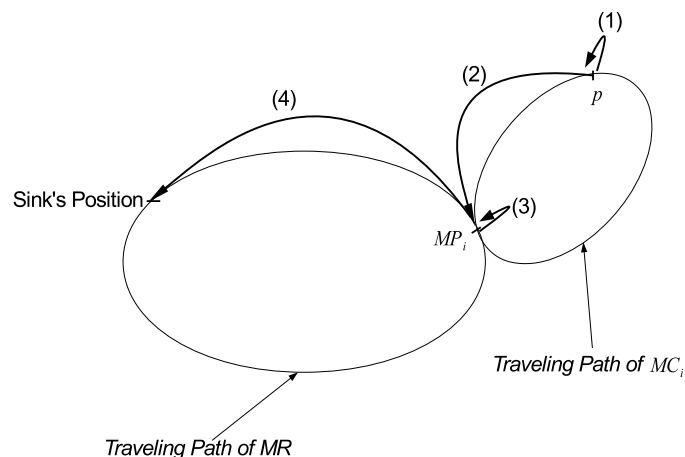
4.2. Estimation of Average Data-Gathering Latency

In the model, we estimate the average data-gathering latency of a packet generated at an arbitrary point p located on the path of MC_i in group G_i . In HiCoDG, the latency of the packet at point p is the sum of four time components as shown in Figure 4, which are:

- (1) the time duration for the packet to wait in the buffer at point p from its generation until MC_i collects it.
- (2) the time taken for MC_i , which carries the packet, to travel from point p to meeting point MP_i .
- (3) the time duration for the packet to wait in MP_i until the MR picks it up.
- (4) the time taken for the MR to travel from MP_i to the sink.

From here, we present the estimation of each of the above time components to obtain the average data-gathering latency of the packet collected by MC_i at point p .

Figure 4. Time components of data latency.



4.2.1. Estimation of Waiting Time in the Buffer

We first estimate the expected latency for a packet waiting in the buffer at point p . Let M denote the total number of packets that MC_i collects at point p at each arrival. Since MC_i periodically visits and collects the packets from point p at every interval T_i , the value of M is the total number of packets generated at point p during period T_i . Then, we have $M \approx RT_i$, assuming $T_i \gg \frac{1}{R}$.

We suppose that packets are generated in the time interval $[0, T_i]$. Then, those packets will be collected by MC_i at T_i as shown in Figure 5. Let Δt denote the time that the first packet is generated at point p during T_i , and let $\Delta\tau$ denote the time difference between the time the last packet is generated and T_i . If we define Υ as the total waiting time of M packets in the buffer, then the value of Υ can be simply calculated as

$$\Upsilon = \sum_{j=1}^M \frac{j}{R} + M\Delta\tau = \frac{M(M+1)}{2R} + M\Delta\tau \tag{15}$$

Let t_p denote the packet interval and $t_p = \frac{1}{R}$. Then, we have

$$\Delta\tau = T_i - \left(\left\lfloor \frac{T_i}{t_p} \right\rfloor t_p + \Delta t \right) \tag{16}$$

By substituting $\Delta\tau$ into Equation (15), we obtain

$$\Upsilon = \frac{t_p M(M+1)}{2} + M \left(T_i - \left\lfloor \frac{T_i}{t_p} \right\rfloor t_p - \Delta t \right) \tag{17}$$

Let t_b^i denote the mean time of a packet waiting in the buffer at point p . Then,

$$t_b^i = \frac{\Upsilon}{M} = \frac{t_p(M+1)}{2} + T_i - \left\lfloor \frac{T_i}{t_p} \right\rfloor t_p - \Delta t \tag{18}$$

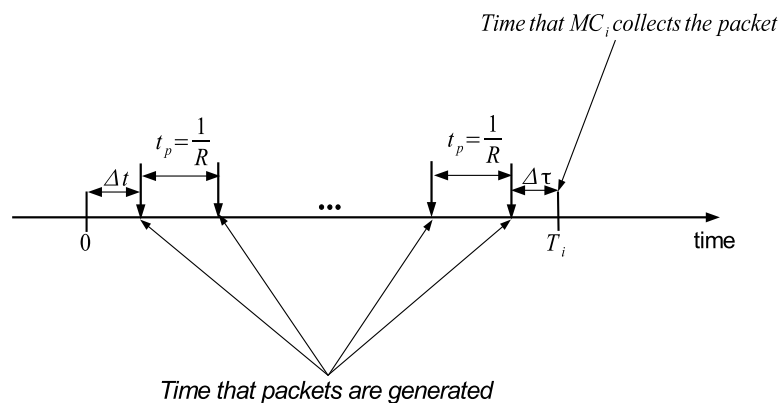
By substituting $M = RT_i = \frac{T_i}{t_p}$ into Equation (18), we get

$$t_b^i = \frac{3T_i}{2} - \left\lfloor \frac{T_i}{t_p} \right\rfloor t_p + \frac{t_p}{2} - \Delta t \tag{19}$$

Since the first packet can be generated in a random time, Δt can be considered a uniform random variable that takes the value in $[0, t_p)$. As a result, t_b^i is also a random variable because it is a function of Δt . Thus, the expected value of t_b^i can be expressed as

$$E[t_b^i] = \frac{1}{t_p} \int_0^{t_p} \left(\frac{3T_i}{2} - \left\lfloor \frac{T_i}{t_p} \right\rfloor t_p + \frac{t_p}{2} - t \right) dt = \frac{3T_i}{2} - \left\lfloor \frac{T_i}{t_p} \right\rfloor t_p \tag{20}$$

Figure 5. Estimation of the average waiting time of a packet in the buffer.



4.2.2. Estimation of Traveling Time in MC_i

After MC_i collects the packet from point p , it will bring the packet to MP_i . Recall that all points on the path of MC_i are randomly placed in a uniform distribution. Let χ be a random variable that represents the traveling distance from MP_i to arbitrary point p on the path of MC_i . Then, the probability density function (pdf) of χ , $f_\chi(x)$, is expressed as

$$f_\chi(x) = \begin{cases} \frac{1}{L_i} & \text{if } x \in [0, L_i) \\ 0 & \text{o/w} \end{cases} \quad (21)$$

Let t_{mc}^i denote the time the packet is in MC_i before MC_i arrives at MP_i at the end of its round. Then, t_{mc}^i is the time it takes for MC_i to travel from point p to MP_i , which is

$$t_{mc}^i = T_i - \frac{\chi}{s_i} \quad (22)$$

Since t_{mc}^i is a function of the random variable χ , the expected value of t_{mc}^i can be calculated as

$$E[t_{mc}^i] = \frac{1}{L_i} \int_0^{L_i} (T_i - \frac{x}{s_i}) dx = T_i - \frac{L_i}{2s_i} = \frac{T_i}{2} \quad (23)$$

note that $T_i = \frac{L_i}{s_i}$

4.2.3. Estimation of Waiting Time in MP_i

In the IS scheme, the packet collected by MC_i is stored in MP_i until the MR arrives and picks it up. Now we estimate the time that the packet might wait at MP_i for the MR under the IS scheme. Note that in CMS, since MR and MC_i are scheduled to periodically meet each other at MP_i , the time a packet waits in MP_i equals zero.

Recall that MR and MC_i periodically visit MP_i every T_0 and T_i ($1 \leq i \leq k$), respectively. We define $\{Y_1, Y_2, \dots, Y_l, \dots\}$ as the time that MC_i arrives at MP_i during its journey, where Y_l is the time of the l^{th} arrival. Note that $Y_{l+1} - Y_l = T_i$. Also, let $\{Z_1, Z_2, \dots, Z_h, \dots\}$ denote the time that the MR visits MP_i , where Z_h is the time that MR arrives at MP_i in its h^{th} round.

Suppose that MC_i collects the packet at point p in the l^{th} round and brings the packet to MP_i at time Y_l ($l \geq 1$). Also, assume that MC_i arrives at MP_i in the interval $[(h-1)T_0, hT_0]$, i.e., $Y_l \in [(h-1)T_0, hT_0]$ as shown in Figure 6. In the time interval $[(h-1)T_0, hT_0]$, the MR starts its h^{th} round from the sink's position at $(h-1)T_0$, visits MP_i at Z_h , and finally returns to the sink at hT_0 .

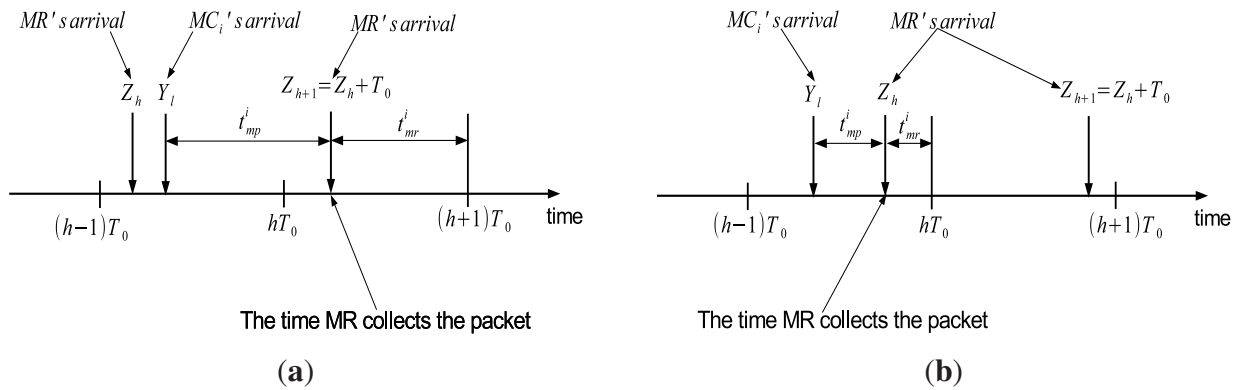
Since MP_i is considered as an arbitrary point randomly placed in the path of the MR, Z_h is a uniform random variable that represents the time the MR arrives at MP_i in the interval $[(h-1)T_0, hT_0]$. Thus, the pdf of Z_h can be expressed as

$$f_Z(z) = \begin{cases} \frac{1}{T_0} & \text{if } z \in ((h-1)T_0, hT_0) \\ 0 & \text{o/w} \end{cases} \quad (24)$$

In the interval $[(h-1)T_0, hT_0]$, the packet stored in MP_i at time Y_i is collected by the MR at time Z_h if the MR visits MP_i after the time MC_i arrives i.e., $Z_h \geq Y_i$. Otherwise, the MR will collect the

packet during its next visit at time Z_{h+1} in the interval $[hT_0, (h + 1)T_0]$. Note that $Z_{h+1} = Z_h + T_0$. Two cases for this collection are shown in Figure 6.

Figure 6. Estimating the waiting time of a packet in the MP. (a) Case 1: $Z_h < Y_l$; (b) Case 2: $Z_h \geq Y_l$.



We define t_{mp}^i as the waiting time of the packet in MP_i . Then, the value of t_{mp}^i is calculated as shown in Figure 6, which is

$$t_{mp}^i = \begin{cases} Z_h + T_0 - Y_l & \text{if } Z_h < Y_l \\ Z_h - Y_l & \text{o/w} \end{cases} \tag{25}$$

Since t_{mp}^i is a function of the random variable Z_h , it is also a random variable. Thus, the expected value of t_{mp}^i can be calculated as

$$E[t_{mp}^i] = \frac{1}{T_0} \left(\int_{(h-1)T_0}^{Y_l} (z + T_0 - Y_l) dz + \int_{Y_l}^{hT_0} (z - Y_l) dz \right) = \frac{T_0}{2} \tag{26}$$

4.2.4. Estimation of Time in MR

After the MR collects the packet at MP_i , it brings the packet to the sink. Let t_{mr}^i denote the time it takes for the MR to move from MP_i to the sink's position. That is

$$t_{mr}^i = \begin{cases} (h + 1)T_0 - Z_{h+1} & \text{if } Z_h < Y_l \\ hT_0 - Z_h & \text{o/w} \end{cases} \tag{27}$$

Since $Z_{h+1} = Z_h + T_0$, the value of t_{mr}^i is rewritten as

$$t_{mr}^i = hT_0 - Z_h \tag{28}$$

Similarly, the expected value of t_{mr}^i can be calculated as

$$E[t_{mr}^i] = \frac{1}{T_0} \int_{(h-1)T_0}^{hT_0} (hT_0 - z) dz = \frac{T_0}{2} \tag{29}$$

Define t_{mean}^i as the mean latency of a packet collected by MC_i from arbitrary point p in group G_i . From Equations (20), (23), (26) and (29), the value of t_{mean}^i can be approximated as

$$t_{mean}^i = \begin{cases} E[t_b^i] + E[t_{mc}^i] + E[t_{mp}^i] + E[t_{mr}^i] = 2T_i - \left\lfloor \frac{T_i}{t_p} \right\rfloor t_p + T_0 & \text{if IS is used} \\ E[t_b^i] + E[t_{mc}^i] + E[t_{mr}^i] = 2T_i - \left\lfloor \frac{T_i}{t_p} \right\rfloor t_p + \frac{T_0}{2} & \text{if CMS is used} \end{cases} \quad (30)$$

Note that in the IS scheme, the packet may need to wait at the MP until the MR visits the MP and picks it up. Thus, the latency of the packet under IS consists of the time the packet waits in the MP. In contrast, under CMS, the packet is immediately delivered to the MR when the MC arrives at the MP since the MR and MC are scheduled to periodically meet each other at the MP.

By substituting $T_i = \frac{L_i}{s_i}$ and $T_0 = \frac{L_0}{s_0}$ into Equation (30), we obtain

$$t_{mean}^i = \begin{cases} \frac{2L_i}{s_i} - \left\lfloor \frac{L_i}{s_i t_p} \right\rfloor t_p + \frac{L_0}{s_0} & \text{if IS is used} \\ \frac{2L_i}{s_i} - \left\lfloor \frac{L_i}{s_i t_p} \right\rfloor t_p + \frac{L_0}{2s_0} & \text{if CMS is used} \end{cases} \quad (31)$$

Note that in Equation (31), the movement speeds s_i of MEs in two cases for IS and CMS may have different values from each other.

We also estimate the movement energy consumption of MEs for data gathering. Let Π denote the total energy consumed by MEs for their movement during data collection time t_c . The value of Π can be calculated as

$$\Pi = \rho \left(t_c s_0 + \sum_{i=1}^k t_c s_i \right) \quad (32)$$

where ρ represents the energy consumed to travel a meter, and we assume $\rho = 8.27$ joules.

4.3. Result Analysis

Equation (31) gives the estimated mean latency of a packet collected by MC_i . Now, in order to verify the accuracy of the analytical model, we compare the results from the model with the results from simulations using a simulator developed in Matlab. For comparison, we consider a network that consists of one MR and four MCs. The MR and MCs have a given path length. There are 100 points randomly placed on the path of each MC. Each point periodically generates a packet every 1 s, *i.e.*, the data generation rate $R = 1$ (packet/second). Also, we assume the MEs have maximum movement speed s_{max} . In CMS, the movement speeds of the MR and MCs are determined using Equations (11) and (12). In IS, every ME moves at the same speed.

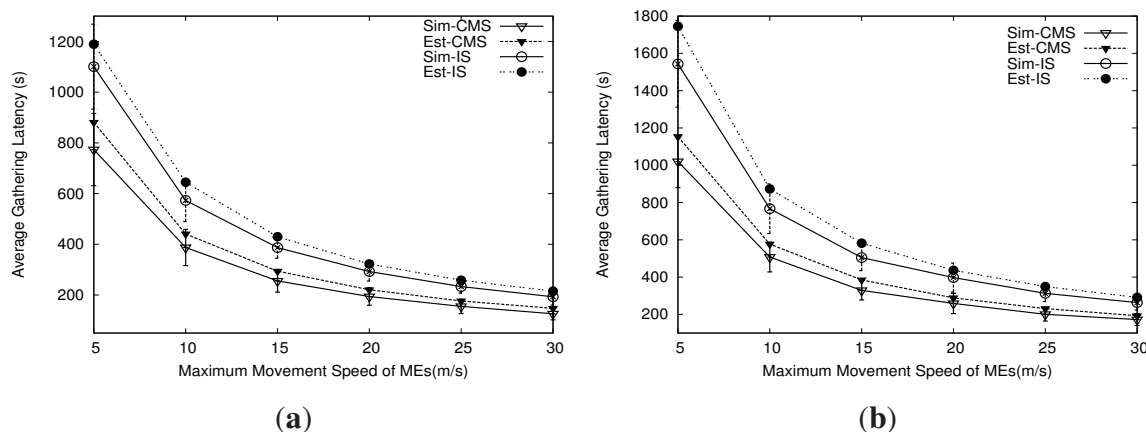
We collect the average latency of the packet from the simulation and the analytical model with two different path length configurations for MEs, as shown in Table 1. For each path length configuration for MEs, we compare the latency under CMS and IS. All MEs in IS move at the average movement speed for MEs in CMS.

Table 1. Path length configurations.

Mobile Elements	Path Length (m)	
	Configuration 1	Configuration 2
<i>MR</i>	2935	3844
<i>MC</i> ₁	1222	1750
<i>MC</i> ₂	2897	3277
<i>MC</i> ₃	1999	3320
<i>MC</i> ₄	2589	1946

Figure 7 shows the average latency of the packet from the estimation and the simulation over variations in maximum movement speeds for MEs. More specifically, Figure 7 show the results for path length configurations 1 and 2, respectively. Each result from the simulation is the average value over 100 simulation runs. In each simulation run, the positions of points on the path for the MEs are randomly chosen. Also, the result from the simulation is the average latency of all packets collected at the sink during the simulation time of 3000 s. In the analytical model, the result is the average of the estimated latency of the packets collected by four MCs.

Figure 7. Average data gathering latency from estimation and simulation (mean ± standard deviation). (a) Configuration 1; (b) Configuration 2.



As shown in Figure 7, the average latency estimated from the analytical model is close to the average latency from the simulation in the two different path length configurations. The gap between the results from the model and the simulation is narrow and the model consistently has a higher latency than the simulation. In other words, the analytical model can provide an upper bound for average latency compared with the simulation.

We also collect energy consumption by the MEs for movement. Table 2 shows the total energy consumed by MEs’ movements during simulation time with two different path length configurations. As shown in Table 2, for each configuration, both CMS and IS show the same movement energy consumption, since they have the same average speed of MEs. Note that for each data relay scheme, the analytical model and the simulation also have the same result of energy consumption.

Table 2. Movement energy consumption.

Maximum Speed (m/s)	Energy Consumption (10^4 J)	
	Configuration 1	Configuration 2
	CMS & IS	CMS & IS
5	49.2	45.6
10	98.4	91.2
15	147.6	136.9
20	196.8	182.5
25	246.1	228.2
30	295.2	273.7

Moreover, the results shown in Figure 7 and Table 2 indicate that on a given amount of energy consumed for the movement of MEs, CMS achieves lower latency than IS. This confirms that the movement scheduling scheme can reduce the data gathering latency.

In summary, in this section we present an analytical model for approximating average latency of packets collected by MEs in HiCoDG, which is verified by comparison with the simulation results. Note that the model is useful for assessing whether MEs can provide the required latency before they are deployed.

5. Performance Evaluation

In this section, we evaluate the performance of HiCoDG variants by comparing them with an mTSP-based approach. Recall that in HiCoDG, two cooperative data relay schemes are proposed, IS and CMS. Also, two different grouping algorithms (*i.e.*, K-means and SPG) are considered. Therefore, four combinations of HiCoDG variants are evaluated in the simulations.

- IS with K-means
- IS with SPG
- CMS with K-means
- CMS with SPG

We first present the simulation parameters and performance metrics, and then discuss the performance results.

5.1. Simulation Setup

Recall that in HiCoDG, MCs visit a point of interest (*i.e.*, a clusterhead) and collect the data from the clusterhead, which received them from sensors within its domain. In this section, for the sake of brevity, we consider a simplified network configuration that only includes a certain number of points (*i.e.*, clusterheads) that are assumed to generate data for their cluster.

The considered network consists of 32 points of interest (POIs) that are randomly deployed in an area of $2000\text{ m} \times 2000\text{ m}$. Each POI generates a data packet every 1 s and stores the packet in its buffer until

an MC collects it. One MR collects data from four MCs and delivers the data to the sink (*i.e.*, five MEs are used). Accordingly, POIs are divided into four groups.

In the mTSP-based approach, we use the ILP formulation for the mTSP in [27] to find m optimal trajectories that begin and end at the position of the sink with the objective of minimizing the total distance traveled. One ME is assigned to each trajectory. Also under mTSP, five MEs (*i.e.*, $m = 5$) are arranged on five routes to visit POIs and return to the sink.

To simulate data gathering in wireless sensor networks, we use a network simulator, Qualnet 6.1, in which the controllable mobility simulator architecture, ConMoSim [30] was integrated to simulate the mobility of MEs. The GNU Linear Programming Kit [31] was used to solve the ILP problem of finding optimal trajectories for MEs in both HiCoDG and mTSP.

Four performance metrics are used to evaluate the performance of algorithms:

- *maximum gathering latency*: the maximum value over elapsed times for all packets from generation to arrival at the sink.
- *average gathering latency*: the average value over elapsed times for all packets from generation to arrival at the sink.
- *movement energy consumption*: the total energy consumed by MEs' movements.
- *maximum number of packets in buffer of ME*: the maximum number of packets that is stored in the buffer of mobile elements.

Also, the simulation time is 10,000 s, and we assume that an ME consumes 8.27 joules to travel one meter (or 0.21 J/inch) [32].

5.2. Performance Analysis

Figure 8 plot the movement trajectories (*i.e.*, movement paths) for MEs generated by HiCoDG with SPG and K-means algorithms, and mTSP, respectively. The length of each path (*i.e.*, the distance that each ME travels to finish one round) and total length of those paths are presented in Table 3. Note that in Table 3, ME_1 and $ME_i (i = 2, \dots, 5)$ in mTSP correspond to MR and $MC_i (i = 1, \dots, 4)$ in HiCoDG, respectively. More specifically, as shown in Figure 8 and Table 3, under mTSP, the difference between the longest and shortest paths for MEs is higher than those of HiCoDG. In other words, under mTSP, an ME may travel a longer distance than others, which results in unbalanced energy consumption. Moreover, under mTSP, the total path length of all MEs is much longer than the total length in HiCoDG.

In addition, as shown in Figure 8 and Table 3, under HiCoDG, when the SPG grouping algorithm is used, the path length of MR is shorter than when using K-means algorithm. The reason is that when the SPG algorithm is used, the path of MR includes POIs which are closer to the sink than when using the K-means algorithm. Note that due to the nature of the K-means algorithm, groups are formed such that POIs in a group are closed to each other without consideration of the sink position. As a result, all POIs in a group can be distant from the sink. In other words, when the K-means algorithm is used, the path of MR may consist of POIs that are distant from the sink, which leads to a longer path length. Moreover, as shown in Table 3, the maximum path length of MCs in K-means algorithm is longer than that of the SPG algorithm. This is possible because a large number of POIs that are close to each other can be in a group when K-means algorithm is used.

Figure 8. Movement paths of mobile elements. (a) Path by HiCoDG with SPG; (b) Path by HiCoDG with K-means; (c) Path by mTSP.

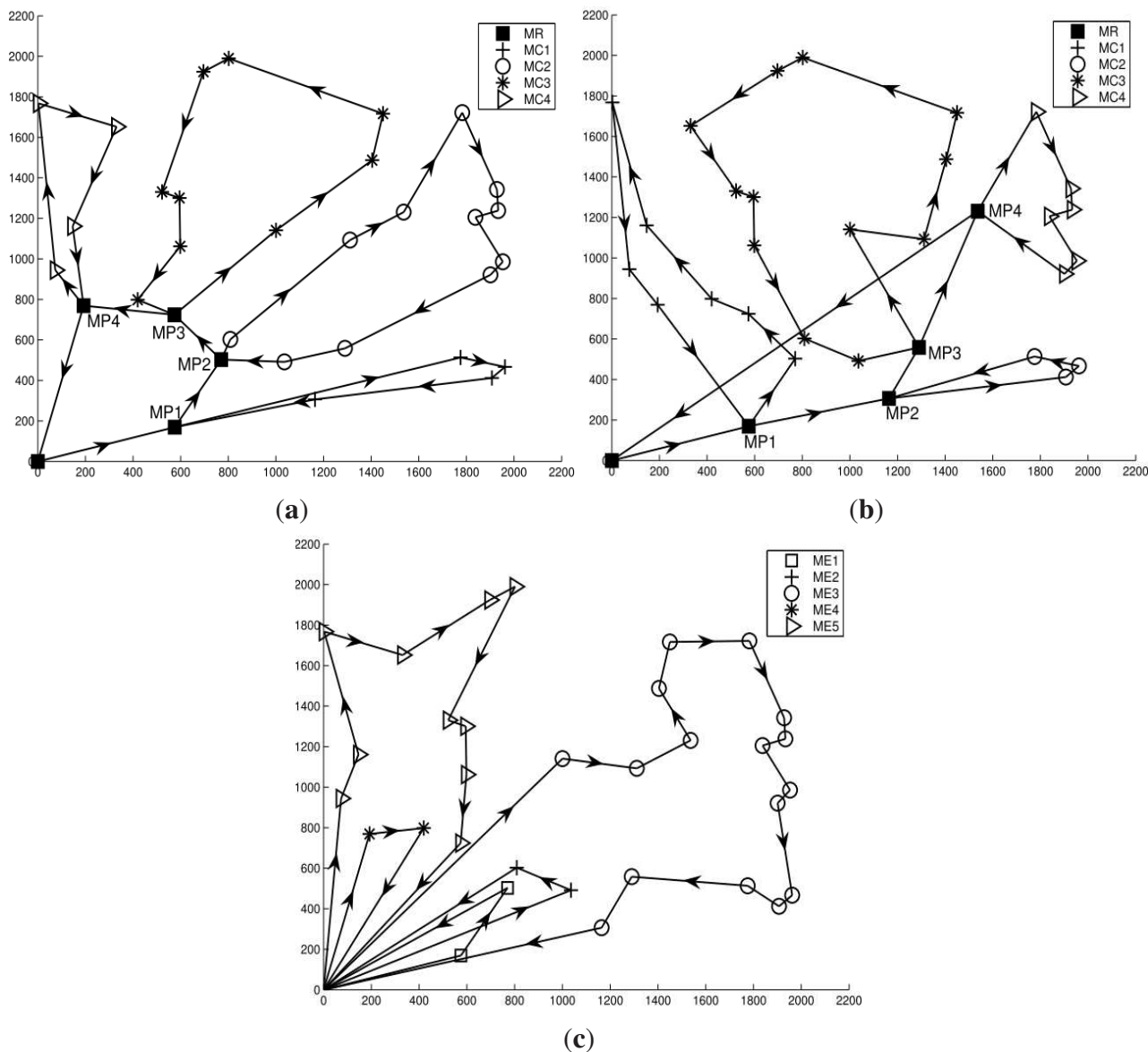


Table 3. Path length of Mobile Elements.

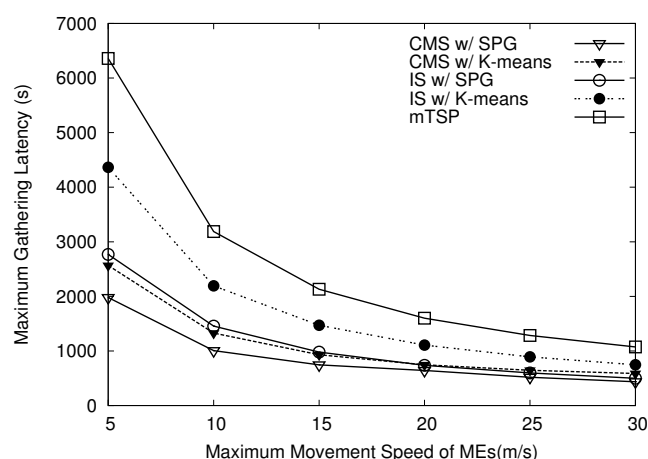
Mobile Elements	Path Length (m)		
	HiCoDG w/SPG	HiCoDG w/K-Means	mTSP
ME_1 (MR)	2459	4172	2346
ME_2 (MC ₁)	2873	3680	5643
ME_3 (MC ₂)	3805	1665	13,534
ME_4 (MC ₃)	3616	4600	4526
ME_5 (MC ₄)	2305	1971	9909
Total distance	15,058	16,088	35,958

Recall that in HiCoDG, along the optimal paths as shown in Figure 8, the MR and MCs cooperate with each other to collect and relay the data from POIs to the sink. More specifically, under IS, the MR and MCs cooperate in a way that each MC periodically drops off the collected data to the MP, and the MR periodically visits the MP and relays the data to the sink. On the other hand, under CMS, a higher level of cooperation is performed. The movement speeds of the MR and MCs are determined such that the MR can periodically meet MCs at MPs to directly receive the data from MCs.

Now we analyze the effects of movement speed of MEs on the performance metrics. The maximum speed for mobile elements varies from 5 to 10, 20, 25, 30 m/s. Recall that, in CMS, the speed of the MEs is determined by using the maximum speed and the path length of the MEs, *i.e.*, the actual movement speeds of the MEs are below the maximum speed. In order to compare the data-gathering latency between CMS and IS that have the same energy consumption for MEs' movements, under IS, MEs always move at the average speed of MEs under CMS for collecting the data. Note that under CMS, when the SPG algorithm is used, the average speed of MEs is higher than when using the K-means algorithm. In this work, in order to favor mTSP in terms of data-gathering latency, we make MEs under mTSP move at the average speed of MEs under CMS with the SPG algorithm.

Figure 9 shows the maximum data-gathering latency of the schemes over variations in maximum speed of the MEs. As shown in Figure 9, four variants of HiCoDG always exhibit a lower maximum latency than mTSP over variations in the maximum speed of MEs. For example, when the maximum speed is 20 m/s, the maximum latency of mTSP is about 1600 s, while the maximum latency of HiCoDG schemes is less than 1100 s. The reason is that in mTSP, every ME has to return to the sink at the end of a data-gathering tour, which results in high latency. Moreover, in mTSP, the optimal trajectories for MEs are unbalanced and an ME travels a long path compared to others. On the other hand, in HiCoDG, MCs travel only in the region assigned to them and do not need to return to the sink.

Figure 9. Effects of movement speed on maximum data-gathering latency.

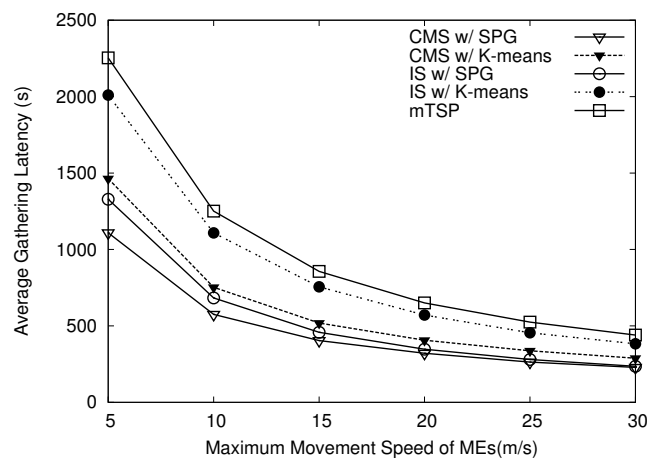


In addition, as shown in Figure 9, in both CMS and IS, when the SPG algorithm is used for grouping, the maximum gathering latency is always lower than when using the K-means algorithm. The reason is that when K-means is used, the path length of the MR is longer than when using the SPG algorithm. Moreover, as shown in Table 3, under K-means, the maximum path length of MCs is longer than that under the SPG algorithm. Note that when the path length of an MC is longer, the time duration from

time the packet is generated to time the MC brings it to the MP also becomes longer. As a result, the packet has higher maximum latency when the K-means algorithm is used for grouping.

Figure 10 compares the average data-gathering latency of the algorithms. As shown in Figure 10, the average latency of algorithm are inversely proportional to the maximum speed of MEs, which agrees with the relationship between traveling time and the movement speed on a given distance. Moreover, all variants of HiCoDG have a lower average gathering latency than mTSP. In particular, CMS with SPG consistently shows the lowest latency compared to others. For example, when the maximum speed of MEs is 10 m/s, the average latency of mTSP is 54% higher than the latency of CMS with SPG. In Figure 10, we also see that when the maximum speed of MEs becomes very high, the average latency tends converge to a certain value, which depends on the sojourn time of MEs at POIs and the sink for data transfer as well as the movement speed of MEs.

Figure 10. Effects of movement speed on average data-gathering latency.



In addition, as shown in Figure 10, when the same grouping algorithm is used, CMS has a lower average latency than IS. The gap between latency values of CMS and IS tends to decrease as the maximum speed of MEs grows. The reason of lower latency in CMS compared to IS is that, under IS, due the lack of cooperation between the MR and MCs, the packets may need wait for a long time at MPs before the MR arrives at MPs and picks them up. On the other hand, under CMS, the waiting time of the packets at MPs is zero. Moreover, as shown in Figure 10, when the K-means algorithm is used, the latency gap between CMS and IS is higher than when using the SPG algorithm. For example, at the maximum speed of 10 m/s, under K-means algorithm, the latency of CMS is about 32% lower than that of IS, while it is about 16% under SPG algorithm.

Figures 11 and 12 show the variation of maximum and average latency of algorithms over time. The maximum speed of MEs is set to 10 m/s. Note that at several time points around time 0, the latency is zero since no packets are collected at the sink. As shown in Figures 11 and 12, all variants of HiCoDG exhibit more stable latency (*i.e.*, maximum and average latency) than mTSP over time. In particular, CMS shows the most consistent latency among algorithms. Recall that in CMS, the initial positions of MCs are determined such that each MC can meet the MR at MP. Thus, at the first round, the MC may start from a middle point of its path, and send the smaller number of packets with a low latency to the

MR at the first meeting at MP. Thus, as shown in Figures 11 and 12, at time before 1000 s, CMS has a lower latency than at time after 1000 s.

Figure 11. Maximum data-gathering latency over time.

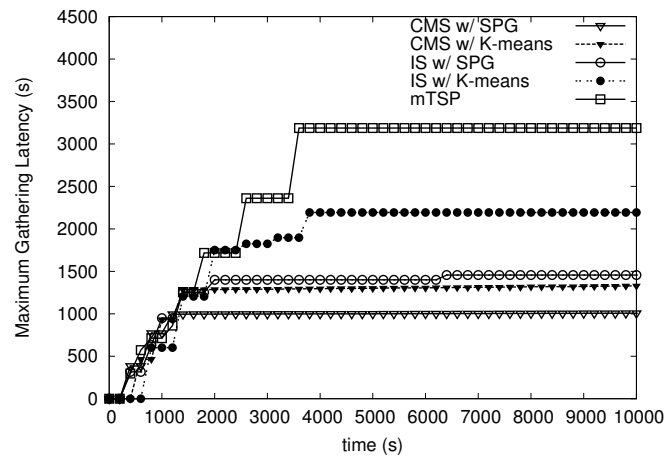
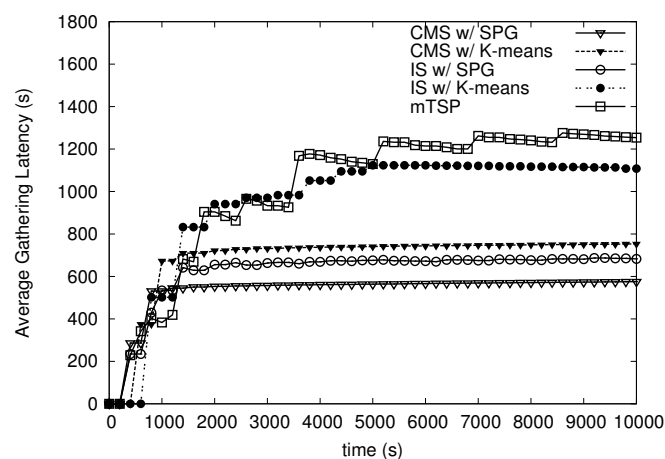


Figure 12. Average data-gathering latency over time.



In addition, note that under IS, due to the lack of movement cooperation among the MR and MCs, the number of packets collected and latency are different over the rounds. Thus, IS shows less consistent latency than CMS. Moreover, as shown in Figure 12, the average latency of mTSP is fluctuating due to the unbalanced trajectories of MEs. From results shown in Figures 11 and 12, we see that variants of HiCoDG achieve not only lower but also more consistent latency than mTSP.

Figure 13 plots the energy consumed by movements of MEs as their maximum speed increases. As shown in Figure 13, when the same grouping algorithm is used, movement energy consumption under the CMS and IS schemes is similar to each other because they have the same average movement speed of MEs. In addition, in both CMS and IS, the SPG algorithm results in higher energy consumption than the K-means algorithm as shown in Figure 13. This is because when the SPG algorithm is used, the average speed of the MEs under CMS is higher than when using K-means algorithm. Note that mTSP shows similar energy consumption to CMS with SPG because MEs under mTSP move at the average speed of MEs in CMS with SPG.

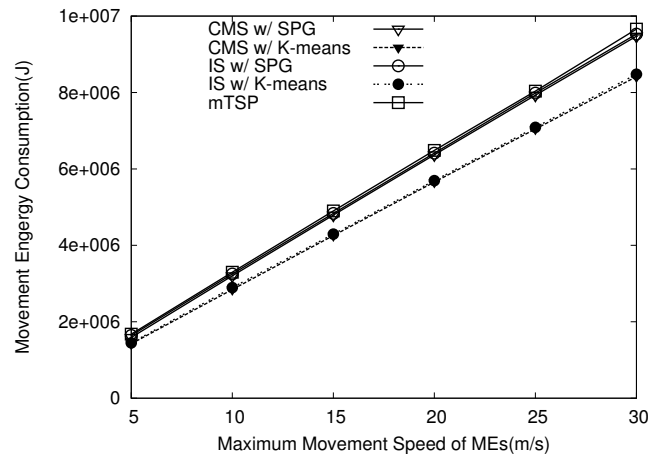
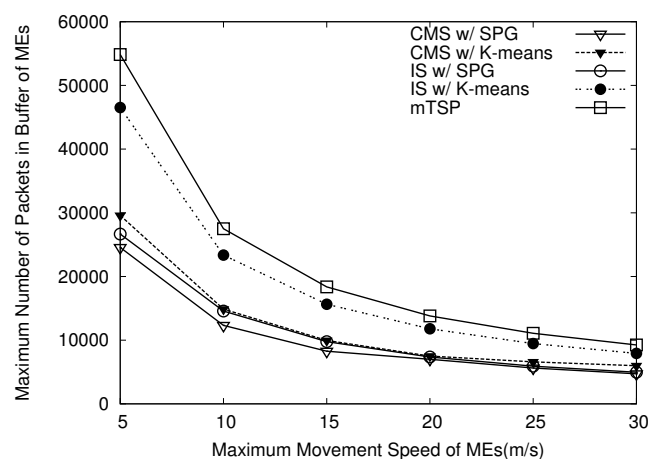
Figure 13. Effects of movement speed on energy consumption.

Figure 14 shows the maximum number of packets stored in the buffer of MEs when their maximum movement speed varies. As shown in Figure 14, when the speed of MEs increases, the maximum number of packets in the buffer of MEs decreases since the number of packets that are collected by MEs in one round decreases. Moreover, mTSP always exhibits a highest maximum number packets in buffer than others since, under mTSP, an ME travels on a much longer tour compared to others. Note that the maximum number of packets in the buffer of MEs indicates a large size of buffer is required for MEs. Thus, those results imply that mTSP requires a higher capacity buffer for MEs compared to HiCoDG, which leads to the increase of the system cost.

Figure 14. Effects of movement speed on maximum number of packets in buffer of MEs.

In addition, as shown in Figure 14, when the same grouping algorithm is used, IS always shows a higher maximum number of packets that are stored in the buffer of MEs than CMS does. In both CMS and IS, the MR always has the largest number of packets in the buffer since it brings the data collected by all MCs to the sink at each round. Note that, under IS, the MR might receive the data collected during several rounds of MCs at MPs. On the other hand, under CMS, the MR always receives the data collected in one round of MCs. As a result, under IS, the maximum number of packets stored in the buffer is higher.

Recall that IS requires not only a larger buffer size for MEs but also the data storage in MPs. In order to estimate the capacity of data storage required for MPs in IS, we also collect the maximum number of packets stored in the buffer of MPs. Table 4 shows the maximum number of packets stored in MPs under IS with SPG and K-means algorithms over variations in maximum speed of MEs. When the path lengths of MC and MR are longer, a larger number of packets will be stored in the buffer of the MP waiting for the MR. Thus, as shown in Table 4, when the K-means algorithm is used, the maximum number of packets stored in MPs is larger than when using the SPG algorithm.

Table 4. Maximum number of packet in buffer of MPs.

Maximum Speed (m/s)	Number of Packets	
	IS w/SPG	IS w/K-Means
5	11,266	17,056
10	5668	8568
15	3800	5740
20	2867	4323
25	2308	3475
30	1934	2909

In summary, those results imply that on given movement energy consumption, CMS can achieve lower data-gathering latency than IS. In addition, the SPG algorithm results in lower latency but higher energy consumption than the K-means algorithm. Also note that by scheduling the movements of the MEs, the need of the data storage at MPs can be avoided. In other words, lower and more consistent data-gathering latency than the existing approach (e.g., mTSP) can be achieved by using cooperative movement among MEs without the need for special special hardware.

6. Concluding Remarks

In this paper, we have proposed a new hierarchical and cooperative data-gathering (HiCoDG) scheme that enables multiple mobile elements to cooperate to collect and relay data. In HiCoDG, there are two types of mobile elements: mobile collectors (MCs) and a mobile relay (MR). MCs collect the data from sensors, and the MR gathers the data from the MCs, delivering them to the sink. An ILP optimization problem has been formulated to find the optimal trajectories for the MCs and the MR with the objective of minimizing energy consumption. Also, we have proposed a cooperative movement scheduling algorithm to determine the optimal movement speeds for MCs and the MR. Simulations have been conducted to compare the performance of HiCoDG with the mTSP-based approach.

For future work, in order to reduce the execution time for finding the trajectories for the MR and MCs, we plan to design a solution method where an approximation algorithm for TSP and a greedy algorithm are applied to find the trajectories for the MC and MR, respectively. We also intend to extend HiCoDG to take into consideration the problem of the limited buffer size of sensors as well as MEs when scheduling the movements for the MEs such that the buffer overflow at sensors and MEs can be avoided.

Acknowledgments

This work was supported by 2014 Research Funds of Hyundai Heavy Industries for University of Ulsan.

Author Contributions

Duc Van Le and Seokhoon Yoon developed the system architecture and algorithms, and conducted the experimental simulations. Duc Van Le, Hoon Oh and Seokhoon Yoon performed analysis and wrote the manuscript. All authors have read and approved the final manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Yick, J.; Mukherjee, B.; Ghosal, D. Wireless Sensor Network Survey. *Comput. Netw.* **2008**, *52*, 2292–2330.
2. Wu, X.; Chen, G.; Das, S. Avoiding Energy Holes in Wireless Sensor Networks with Nonuniform Node Distribution. *IEEE Trans. Parallel Distrib. Syst.* **2008**, *19*, 710–720.
3. Di Francesco, M.; Das, S.K.; Anastasi, G. Data Collection in Wireless Sensor Networks with Mobile Elements: A Survey. *ACM Trans. Sens. Netw.* **2011**, *8*, 7:1–7:31.
4. Le, D.V.; Oh, H.; Yoon, S. A Novel Hierarchical Cooperative Data Gathering Architecture Using Multiple Mobile Elements. In Proceedings of the 2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN), Shanghai, China, 8–11 July 2014; pp. 522–527.
5. Wang, Z.; Basagni, S.; Melachrinoudis, E.; Petrioli, C. Exploiting Sink Mobility for Maximizing Sensor Networks Lifetime. In Proceedings of the 38th Annual Hawaii International Conference on System Sciences, Big Island, HI, USA, 3–6 January 2005; p. 287a.
6. Ma, M.; Yang, Y. SenCar: An Energy-Efficient Data Gathering Mechanism for Large-Scale Multihop Sensor Networks. *IEEE Trans. Parallel Distrib. Syst.* **2007**, *18*, 1476–1488.
7. Yuan, B.; Orłowska, M.; Sadiq, S. On the Optimal Robot Routing Problem in Wireless Sensor Networks. *IEEE Trans. Knowl. Data Eng.* **2007**, *19*, 1252–1261.
8. Gu, Y.; Bozdog, D.; Ekici, E.; Ozguner, F.; Lee, C.G. Partitioning Based Mobile Element Scheduling in Wireless Sensor Networks. In Proceedings of the Second Annual IEEE SECON, Santa Clara, CA, USA, 26–29 September 2005; pp. 386–395.
9. Somasundara, A.A.; Ramamoorthy, A.; Srivastava, M.B. Mobile Element Scheduling for Efficient Data Collection in Wireless Sensor Networks with Dynamic Deadlines. In Proceedings of the 25th IEEE International Real-Time Systems Symposium, Lisbon, Portugal, 5–8 December 2004; pp. 296–305.
10. Guo, S.; Wang, C.; Yang, Y. Mobile Data Gathering with Wireless Energy Replenishment in Rechargeable Sensor Networks. In Proceedings of the 2013 IEEE INFOCOM, Turin, Italy, 14–19 April 2013; pp. 1932–1940.

11. Kumar, A.K.; Sivalingam, K.M.; Kumar, A. On Reducing Delay in Mobile Data Collection Based Wireless Sensor Networks. *Wirel. Netw.* **2013**, *19*, 285–299.
12. Sugihara, R.; Gupta, R. Optimal Speed Control of Mobile Node for Data Collection in Sensor Networks. *IEEE Trans. Mob. Comput.* **2010**, *9*, 127–139.
13. Gao, S.; Zhang, H.; Das, S.K. Efficient Data Collection in Wireless Sensor Networks with Path-Constrained Mobile Sinks. *IEEE Trans. Mob. Comput.* **2011**, *10*, 592–608.
14. Zhao, M.; Ma, M.; Yang, Y. Efficient Data Gathering with Mobile Collectors and Space-Division Multiple Access Technique in Wireless Sensor Networks. *IEEE Trans. Comput.* **2011**, *60*, 400–417.
15. Jea, D.; Somasundara, A.; Srivastava, M. Multiple Controlled Mobile Elements (Data Mules) for Data Collection in Sensor Networks. In *Distributed Computing in Sensor Systems*; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3560, pp. 244–257.
16. Kim, D.; Abay, B.; Uma, R.N.; Wu, W.; Wang, W.; Tokuta, A. Minimizing Data Collection Latency in Wireless Sensor Network with Multiple Mobile Elements. In Proceedings of the 2012 IEEE INFOCOM, Orlando, FL, USA, 25–30 March 2012; pp. 504–512.
17. Gu, Z.; Hua, Q.S.; Wang, Y.; Lau, F. Reducing Information Gathering Latency Through Mobile Aerial Sensor Network. In Proceedings of the 2013 IEEE INFOCOM, Turin, Italy, 14–19 April 2013; pp. 656–664.
18. He, L.; Pan, J.; Xu, J. Analysis on Data Collection with Multiple Mobile Elements in Wireless Sensor Networks. In Proceedings of the 2011 IEEE GLOBECOM, Houston, TX, USA, 5–9 December 2011; pp. 1–5.
19. He, L.; Pan, J.; Xu, J. Reducing Data Collection Latency in Wireless Sensor Networks with Mobile Elements. In Proceedings of the 2011 IEEE INFOCOM Workshop, Shanghai, China, 10–15 April 2011; pp. 572–577.
20. He, L.; Pan, J.; Xu, J. A Progressive Approach to Reducing Data Collection Latency in Wireless Sensor Networks with Mobile Elements. *IEEE Trans. Mob. Comput.* **2013**, *12*, 1308–1320.
21. Aslanyan, H.; Leone, P.; Rolim, J. Data Propagation with Guaranteed Delivery for Mobile Networks. In Proceedings of the 9th International Conference on Experimental Algorithms, Naples, Italy, 20–22 May 2010; pp. 386–397.
22. Cai, C.; Yang, C.; Zhu, Q.; Liang, Y. Collision Avoidance in Multi-Robot Systems. In Proceedings of the International Conference on Mechatronics and Automation, Harbin, China, 5–8 August 2007; pp. 2795–2800.
23. Hennes, D.; Claes, D.; Meeussen, W.; Tuyls, K. Multi-robot Collision Avoidance with Localization Uncertainty. In Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, Valencia, Spain, 4–8 June 2012; pp. 147–154.
24. Tselikis, C.; Mitropoulos, S.; Komninos, N.; Douligeris, C. Degree-Based Clustering Algorithms for Wireless Ad Hoc Networks Under Attack. *IEEE Commun. Lett.* **2012**, *16*, 619–621.
25. Peng, W.; Edwards, D. K-Means Like Minimum Mean Distance Algorithm for wireless sensor networks. In Proceedings of the 2nd International Conference on Computer Engineering and Technology (ICCET), Chengdu, China, 16–18 April 2010; pp. V1:120–V1:124.
26. Pataki, G. Teaching Integer Programming Formulations Using The Traveling Salesman Problem. *SIAM Rev.* **2003**, *45*, 116–123.

27. Kara, I.; Bektas, T. Integer linear Programming Formulations of Multiple Salesman Problems and Its Variations. *Eur. J. Oper. Res.* **2006**, *174*, 1449–1458.
28. Kara, I.; Bektas, T. Integer Linear Programming Formulation of the Generalized Vehicle Routing Problem. In Proceedings of the 5th EURO/INFORMS Joint International Meeting, Istanbul, Turkey, 6–10 July 2003; pp. 1–21.
29. Papadimitriou, C.H. The Euclidean Travelling Salesman Problem is NP-complete. *Theor. Comput. Sci.* **1977**, *4*, 237–244.
30. Le, D.V.; Oh, H.; Yoon, S. RoCoMAR: Robots' Controllable Mobility Aided Routing and Relay Architecture for Mobile Sensor Networks. *Sensors* **2013**, *13*, 8695–8721.
31. GLPK (GNU Linear Programming Kit). Available online: <https://www.gnu.org/software/glpk/> (accessed on 27 January 2014).
32. Rahimi, M.; Shah, H.; Sukhatme, G.; Heideman, J.; Estrin, D. Studying the Feasibility of Energy Harvesting in a Mobile Sensor Network. In Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA-03), Taipei, Taiwan, 14–19 September 2003; pp. 19–24.

© 2014 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).