Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2011, Article ID 235724, 13 pages
doi:10.1155/2011/235724

Research Article

Energy-Aware Distributed Intelligent Data Gathering Algorithm in Wireless Sensor Networks

Rongbo Zhu, Yingying Qin, and Jiangqing Wang

College of Computer Science, South-Central University for Nationalities, Wuhan 430074, China

Correspondence should be addressed to Rongbo Zhu, rongbozhu@gmail.com

Received 29 September 2010; Accepted 1 April 2011

To plan the data collecting path for the mobile collector in wireless sensor network (WSN), an efficient energy-aware distributed intelligent data gathering algorithm (DIDGA) is proposed, which includes cluster formation and path formation phases. In cluster formation phase, an energy-efficient distributed clustering scheme is proposed to form a coverage-efficient WSN, which constructs a minimum connected dominating set (MCDS) based on maximal independent sets (MISs) in distributed and localized manner, and the node with more power is selected to be the cluster head in turn to prolong the network lifetime. In path formation phase, a path formation optimized algorithm (PFOA) is proposed to resolve the path formation NP problem with dynamic requirements. Then DIDGA uses the cluster head relay mechanism for planning the data gathering path. Compared with existed algorithms, detailed simulation results show that the proposed DIDGA can reduce average hop counts, average data gathering time, energy consumption, increase the efficiency of event detection ratio and prolong the network lifetime.

1. Introduction

Recent advances in wireless communications and electronics have enabled the development of low-cost, low-power, multifunctional sensor nodes [1], which consist of sensed and data processing, and communicating components, leverage the idea of wireless sensor networks (WSNs) [1, 2]. Typical applications of WSNs are the unmanned environmental monitoring, military surveillance, unmanned health monitoring, target tracking, inventory management, multimedia transmitting, and so on [3, 4]. Considering that battery is the main source of energy for the sensor nodes, how to reduce the high-energy expenditure in multihop routing and extend WSN's lifetime is a major challenge [2, 4].

One important task of WSNs is to collect useful information from the sensory field [5]. For a large-scale, data centric sensor network, it is inefficient to use a single, static data sink to gather data from all sensors [6, 7]. In some applications, sensors are deployed to monitor separate areas. In each area, sensors are densely deployed and connected, while sensors that belong to different areas may be disconnected. Unlike fully connected networks, some sensors cannot forward data to the data sink via wireless links [8, 9]. In some complex terrain environment, especially in noise interference and mobile case, how to effectively gather data is a challenge task with limited power. In general, most data-gathering schemes aim to prolong lifetime of WSNs by saving power consumption and optimized data transmitting scheme [9]. In [10], the minimal aggregation time (MAT) NP-hard problem with collision-free transmission was studied, where a sensor can not receive any data if more than one sensor within its transmission range sends data at the same time. Another important way to save energy is to decrease data transmitting with some schemes, such as gathering correlated data [11], or compressing the data [12]. However, to some extend, such schemes are complex and only suitable in certain specific situations.

In actual monitoring environment, the shelter, noise interference, and complex terrain will degrade the performance of data-gathering schemes [13–16]. However, a mobile data collector is perfectly suited to such applications [17, 18]. More recently, the use of the mobile robot to collect data has been explored for improving the networking facilities in the system [19, 20]. A mobile entity [21] was proposed to pick up data from sensor nodes, where sensor nodes transmit data only over a short range that requires less transmission power. The objective of such research mentioned above is meant for reducing the communication energy required at the sensor nodes and to maximize the sensor network lifetime. But significant challenges are encountered how to design the data-gathering path for the mobile robot and how to improve the efficiency of data-gathering with the help of mobile robots [22–24]. In the path planning of the mobile robot, it is an NP-hard problem [2, 25] to find the shortest path, and in large-scale wireless sensor network, the latency of the data will be large and too. Moreover, there is much research about the problem of planning a path which can be the complete coverage in an environment by a mobile robot. Commonly, the methods are spiral path and straight rows path with backtracking to assure the whole network is visited. These navigation methods are very simple but not very efficient. In [26], a number of mobile collectors, called data mules, traverse the sensed field along parallel straight lines and gather data from sensors. However, in practice, data mules may not always be able to move along straight lines, for example, obstacles or boundaries may block the moving paths of data mules. Moreover, the performance and the cost of the data mule scheme depend on the number of data mules and the distribution of sensors. When only a small number of data mules are available and not all sensors are connected, data mules may not cover all the sensors in the network if they only move along straight lines. Another problem is that the existed mobile data-gathering scheme doesn't consider the worst-case delay and time-limited data for entire data-gathering. This can further cause buffer overflow and delay in the agents and reducing the reliability of data collection.

In this paper, the intelligent mobile data collector has been explored to collect data for improving the networking facilities in the system. To improve the efficiency of data collecting, an efficient energy-aware distributed intelligent data-gathering algorithm (DIDGA) is proposed to plan the data-gathering path for the mobile collector in WSNs. DIDGA will reduce high energy expenditure in multihop routings and increase the efficiency of the mobile collector to gather data. The main contributions of DIDGA may be summarized as follows:

(i) DIDGA creates and constructs a minimum connected dominating set (MCDS) based on maximal independent sets (MISs) in distributed and localized manner, which will reduce high energy expenditure in multihop routing with the cluster head, and it will select the node with more power to be the cluster head in turn to prolong the network lifetime. Then DIDGA restricts to the hop counts of the sensed data transmission by communicating with the cluster head in one hop. Sensor nodes' data transmission can cooperate with mobile collector's data-gathering path, which will increase the efficiency of the mobile collector to collect data. DIDGA disperses routing hot spots in WSNs. So it extends the network's lifetime.

(ii) DIDGA considers the path formation NP problem with dynamic requirements under dynamic network environment. The characteristics of the problem are described in terms of sink, the high priority cluster header with urgent event, and the time-limited data between requirement nodes. A path formation optimized algorithm (PFOA) is proposed which combines ant colony algorithm and evolutionary algorithm to satisfy the constraints. Then DIDGA uses the cluster head relay mechanism for planning the data-gathering path. For the delay constraint event and time critical data, the collector will report it to sink directly. The path formation scheme will meet the requirements of cluster headers with less-power and real-time urgent events in practical applications.

The remainder of this paper is organized as follows. In Section 2, we discuss some related studies. Section 3 describes the details of the proposed data-gathering algorithm. Section 4 presents and assesses the simulation results. Finally, Section 5 concludes the paper.

2. Related Work

In practice, the network performance is degraded by the complex monitoring terrain, multihop, and interference and time-varying property of the wireless channel [21]. To make effective use of the gigantic amount of individual sensor readings, it is essential to equip WSNs with scalable and energy-efficient data-gathering mechanisms. Some distinct characteristics of WSNs, such as large node density, unattended operation mode, high dynamicity and severe resource constraints, pose a number of design challenges on sensor data-gathering schemes. Many research activities have been carried out on the research issue. Since the fundamental task of WSN is to gather data efficiently with less resource consumption, to address the problem, there are two threads of research to improve the performance of data collecting: optimized data-gathering schemes and mobile collector assisted data-gathering in WSNs.

For the first thread, most data-gathering algorithms aim to prolong lifetime with some optimized schemes. The balance energy consumption problem was formulated as an optimal transmitting data distribution problem [9] and minimal aggregation time (MAT) problem [10] are formulated as optimal problems. In [27], the construction of a data-gathering tree to maximize the network lifetime was studied, and the problem is also shown to be NP-complete. To balance load within each cluster, an even energy dissipation protocol (EEDP) was proposed for efficient cluster-based data-gathering in WSNs [24]. The method proposed in [28] gathers data in high-density WSNs in real-time, which determines network topology by hierarchical clustering to avoid radio collision and enables to gather data with minimum data latency from numerous high-density sensor nodes. To address the problem of gathering information in WSNs, the work in [29] took into account the fact that interference can occur at the reception of a message at the receiver sensor. However it assumes the distribution of sources are known. Another way to save energy is to decrease data transmitting with some schemes. A new distributed framework to achieve minimum energy data-gathering was proposed in [11]. To minimize the total energy for compressing and transporting information, the problem of constructing a data-gathering tree over a WSN was studied in [12]. And a tunable data compression technique was proposed in [30]. In fact, the schemes mentioned above have some defects for actual environment. To some extent, all those schemes require the node has extra computation to optimize the data transmission or compress and decompress data.

For the second thread, nodes in WSNs are in multihop and mobile environment in general. The characteristic of each link will change timely. In the content of the WSNs where each node only has a partial view of the network, it is very important for each node to estimate the system status by a simple and accurate method [21, 23]. Especially for data transmission with less power consumption, a mobile data collector is more perfectly suited to such applications, for the collector can be equipped with a powerful transceiver and battery. Instead, it is effective to collect data by assisted mobile collector which can achieve better power saving performance [22, 24, 26]. A new data-gathering mechanism called M-collector for large-scale wireless sensor networks was proposed in [31] by introducing mobility into the network. However, it just considers the single-hop data-gathering problem. An adaptive data-gathering protocol was proposed in [32] that employs multiple mobile collectors (instead of sinks) to help an existing WSN achieve such requirements, which adopts a virtual elastic-force model to help mobile collectors adjust their moving speed and direction while adapting to changes within the network. However, the number of collectors can not be predefined, for the irregularity of the information generation rate as well as the cost of mobile collectors. A well-planned adaptive moving strategy (AMS) for a mobile sink in large-scale, hierarchical sensor networks was presented in [33]. The mobile sink traverses the entire network uploading the sensed data from cluster heads in time-driven scenarios. However, it just tries to minimize the whole tour length to save energy. An efficient hybrid method for message relaying and load balancing was proposed in low-mobility wireless sensor networks in [34]. The system uses either a single-hop transmission to a nearby mobile sink or a multihop transmission to a far-away fixed node depending on the predicted sink mobility pattern. Another problem is that the existed mobile data-gathering scheme didn't consider the worst-case delay and time-limited data for entire data-gathering in practical dynamic monitoring environment.

3. Proposed Scheme

3.1. System Model. A distributed WSN is modeled as an undirected graph $G(V, E)$, where V is the set of sensor nodes and E is the set of communication links. The location of each sensor node v_i is denoted by (x_i, y_j) and assumed to be known in advance. The Euclidean distance between any two nodes v_i and v_j is denoted by $d(v_i, v_j)$. We assume that the WSN is partitioned into multiple node-disjoint or node-joint minimum connected dominating sets (MCDSs) MC_i, for the limitation of terrain. A MCDS MC_i will include many maximal independent sets (MISs) $MIS_{i,j}$. In each MIS $MIS_{i,j}$, a special node will act as the cluster header CH_{ij}. And a sink is the final destination of all data aggregations in a WSN. A mobile collector is mounted on a moving entity such as

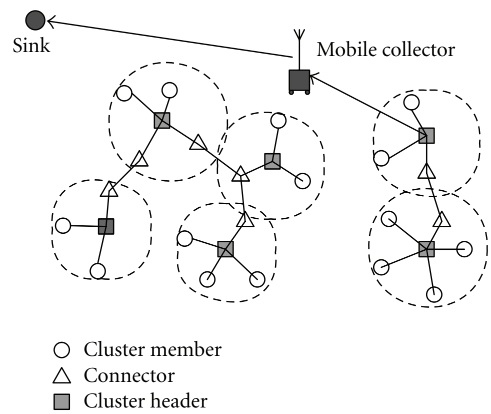


Figure 1: An example of the system.

an unmanned aerial mobile collector whose motion can be controlled and with sufficient energy supplies, which collects data directly from cluster headers and relays data to the sink. Hence, the system will form many clusters such that data is aggregated at cluster heads before mobile collector reach them, which will result in saving effective time for data-gathering and also reduce the overall cost of the system.

In a distributed WSN, resource-constrained sensor node v_i will transmit or forward the data to its cluster header CH_{ij} with more energy and communication capabilities. The intercluster connectivity constraint is slightly relaxed in our system, in that the nodes in the cluster are not required to be connected to each other. Considering that it is difficult to make the mobile collector visit every node in the network, so the mobile collector only needs to communicate with cluster headers to gather sensed data. In our scheme, the mobile collector has two channels. One channel is used for communication with sensors, while the other is used to relay data to the sink. We assume that the communication range of a mobile collector is large enough to sustain the direct communication between the mobile collector and the sink. This relaxation in connectivity and communication gives the system better fault tolerance and its distributed nature. In addition, such data-gathering method will overcome the terrain movement limitations, and decrease medium access control (MAC) collisions and congestion when the mobile collector is within communication range of a group of sensor nodes. Figure 1 shows an example of the system.

In order to gather data effectively, the proposed data-gathering algorithm should address two important fundamental problems: cluster formation and path formation.

Cluster formation one of the crucial challenges in the data-gathering of WSN, is energy efficiency. Cluster-based network organizations are considered to be the most favorable approach in terms of energy efficiency. In this approach, sensor nodes are organized into clusters, and one sensor node v_i in each cluster is selected as the cluster head CH_{ij} which then plays a special role as a transfer point. Additionally, each cluster header CH_{ij} creates a transmission schedule for the sensor nodes within the cluster. This schedule allows the radio components of each noncluster header node to be powered down except during scheduled transmit times. Hence, the proposed intelligent data-gathering algorithm should adopt an energyefficient distributed clustering scheme to form a coverage-efficient WSN, before the mobile collector gathering data. The cluster formation scheme should be distributed and localized.

Path formation: From time to time, the mobile collector needs to conduct data-gathering in the WSN by traversing each cluster. The mobile collector will leave from one node, and visit any cluster header CH_{ij} in each MIS $MIS_{i,j}$. A clustering-based data-gathering method is a path planning scheme over a clustered network, where any moving path contains only cluster headers. It is known that this path planning scheme is energy-efficient in mobile wireless networks since it suffers less from collision and reduces the amount of both control messages and routing related information. When the mobile collector is within the communication range of a cluster header, the latter can collect all sensed data in its cluster and forward to the collector. For some important and time-limited event data, the mobile collector should transmit the data to the sink immediately. When returning to the sink, the mobile collector will relay all other sensed data to the sink. After cluster formation phase, the path formation scheme for the mobile collector is equal to resolve a travel salesman problem (TSP) with some constraints, which is a typical NP problem for the combination optimization. The major disadvantage of such a solution is that there is considerable delay in acquiring sensed data, which depends on tour length and mobile collector's speed. So, the proposed intelligent data-gathering algorithm should minimize the total traversal distance of the mobile collector with some conditions, such as the cluster headers CH_{ij} with higher priority events should be visited first.

3.2. Cluster Formation. In the proposed DIDGA, we assume that the sensor nodes are randomly scattered over the network. The cluster formation scheme will be done using the distributed manner, as explained below.

In order to decrease the path length of the mobile collector, the sensor nodes first probe the network and MCDS will be constructed in a distributed way. The proposed cluster formation method exploits the localized network structure and the remaining energy of neighboring nodes in order to define a new way for estimating dynamically the cluster heads.

Definition 1. A subset S of V is a dominating set (DS) if each node u_i in V is either in S or adjacent to some node v_i in S. Nodes from S are called cluster headers, while nodes not from S are called cluster members.

Definition 2. A subset C of V is a connected dominating set (CDS) if C is a dominating set and C induces a connected subgraph.

In the CDS, the nodes in C can communicate with any other node in the same set without using nodes in $V - C$. A DS with the minimum number of nodes is called a minimum dominating set (MDS). A CDS with minimum number is denoted by MCDS.

```
Input: All sensors v_i, i ∈ [1, N]
Output: All MIS_(i, j)
for each node v_i do
    if v_i.nLower ≡ 0 then
        v_i.status ← cluster header;
        BroadcastMessage(HEADER);
    end
    if receive HEADER message then
        v_i.status ← cluster header;
        BroadcastMessage(HEADER);
    end
    if receive HEADER message then
        v_i.nLower ← v_i.nLower-1;
        if v_i.nLower ≡ 0 then
            v_i.status ← cluster header;
            BroadcastMessage(HEADER);
        end
    end
end
```

Algorithm 1: MIS construction algorithm.

Definition 3. A subset V' of vertices V in a graph G is an independent set (IS) if, for any pair of vertices in V', there is no edge between them.

An MIS is a maximum cardinality subset V' of V so that there is no edge between any two vertices in V'. And the MIS construction algorithm is shown in Algorithm 1. Each node v_i is in one of four states: candidate, cluster header, cluster member, and connector. Each node v_i is initialized as candidate state and subsequently enters either the cluster member state or the cluster header state. The connector state can only be entered from the cluster member state. There is a local variable *nLower* in each node v_i. It stores the number of the current candidate neighbors with lower identities (IDs) and is initially equal to the total number of neighbors with lower IDs. A candidate node with $nLower = 0$ changes its own state to cluster header and then broadcasts a HEADER message. Upon receiving a HEADER message, a candidate node changes its own state to cluster member and then broadcasts a MEMBER message. Upon receiving a MEMBER message, a candidate node decreases *nLower* by one if the sender has a lower ID. If *nLower* is equal to 0 after the updating, it changes its own state to cluster header, and then broadcasts a HEADER message.

After the MIS construction, the MCDS construction algorithm is shown as in Algorithm 2. Then each cluster header generates a REQ_HEADER message to find all other cluster headers within three hops. This message is broadcasted at most three hops before it arrives at a cluster header. When a cluster member receives this message, it appends its ID into the node list included in the REQ_HEADER message and then broadcasts this message. In this way, when a REQ_HEADER message arrives at a cluster header, it has already recorded the IDs of all nodes in its node list which form the path from the cluster header originating this message to the cluster header receiving this message. When a

Input: All sensors v_i, $i \in [1, N]$
Output: All MC_i

```
for each node v_i do
    if v_i is cluster header then
        BroadcastMessage(REQ_HEADER);
        if cluster member v_j receive REQ_HEADER
        then
            appends ID of v_j in
            REQ_HEADER.nodeList;
            BroadcastMessage(REQ HEADER);
        end
        else if cluster header v_j receive
        REQ HEADER then
            REPLY_CON.nodeList ← reverse order
            of REQ_HEADER.nodeList;
            SendMessage(REPLY_CON);
            if cluster member v_k receive
            REQ_HEADER AND ID of v_k ∈
            REPLY_CON.nodeList then
                v_k.state ← connector;
                Send(REPLY_CON) to the next-hop
                node in REPLY_CON.nodeList;
            end
        end
    end
end
```

Algorithm 2: MCDS construction algorithm.

cluster header receives a REQ_HEADER message for the first time from another cluster header, it generates a REPLY_CON message including the path that this message should visit and sends this message. This path is the reverse order of the one in the REQ_HEADER message that it has received before. When a cluster member's ID that is included in the path of the REPLY_CON message receives this REPLY_CON message, it changes its state to connector and sends this message to the next-hop node according to the path in this message.

Lemma 1. *For every cluster member node v_i, it can be connected to at most five cluster header nodes in the MCDS algorithm.*

Proof. Let S denote a set of cluster header nodes in one-hop neighbors of node v_i and CH_j denote each cluster header which is adjacent to v_i. Suppose $|S| \geq 6$. We know in the unit disk centered at v_i; there must be two cluster header neighbors CH_j and CH_k such that the angle $\angle CH_j v_i CH_k$ is at most 60°. So, the distance between CH_j and CH_k is at most one unit, which implies that there is an edge between CH_j and CH_k. This is a contradiction with the definition of MIS. □

Lemma 2. *Let D^* be a MCDS for G and let D be any MIS for G. Then $|D| \leq 5|D^*|$.*

Proof. Since D is an independent set, by Lemma 1, no vertex in D^* can dominate more than five vertices in D. Hence, $|D| \leq 5|D^*|$ and the theorem follows. □

Theorem 1. *The distributed algorithm for constructing an MCDS has a constant approximation factor of MCDS in G.*

Proof. From Lemmas 1 and 2, we can demonstrate this theorem instantly. □

The MCDS algorithm formulates a certain number of clusters. As being a cluster head is much more energy intensive than being a cluster member node, this requires that each node take its turn as cluster head to prolong network lifetime. Assumption that each sensor node v_i elects itself to be a cluster head at the beginning of round $r + 1$ with probability $P_i(t)$, which is chosen such that the expected number of cluster heads for this round is k. Thus if there are N nodes in the network, we have

$$\sum_{i=1}^{N} P_i(t) = k. \tag{1}$$

Ensuring that all nodes are cluster heads the same number of times requires each node to be a cluster head once in N/k rounds on average. We induce a parameter $C_i(t)$ indicator function whether or not node v_i has the chance to be a cluster head. If $C_i(t) = 1$, it means that node v_i has not been a cluster head in the most recent $(r \bmod (N/K))$ rounds. The node v_i becomes a cluster head at round r with probability $P_i(t)$:

$$P_i(t) = \frac{k}{N - k \cdot (r \bmod (N/K))}, \quad \text{if } C_i(t) = 1. \tag{2}$$

If $C_i(t) = 0$, it means that node v_i has been a cluster head in the most recent r rounds. The node v_i is not eligible to be a cluster head at round r.

$$P_i(t) = 0, \quad \text{if } C_i(t) = 0. \tag{3}$$

The expected number of nodes that have not been cluster heads in the first r rounds is $N - k * r$. After N/k rounds, all nodes are expected to have been cluster head once, following which they are all eligible to perform this task in the next sequence of N/k rounds. Since $C_i(t) = 1$, if node v_i is eligible to be a cluster head at time t and $C_i(t) = 0$; otherwise, the total number of nodes that are eligible to be a cluster head at time t:

$$E\left[\sum_{i=1}^{N} C_i(t)\right] = N - k \cdot \left(r \bmod \left(\frac{N}{K}\right)\right). \tag{4}$$

This ensures that the energy at all nodes is approximately equal to each other after every N/k rounds. The expected number of cluster heads E_{number} per round is

$$E_{\text{number}} = \sum_{i=1}^{N} P_i(t) \cdot C_i(t). \tag{5}$$

Assume that the energy of each sensor equipped is E. To transmit l bits message with distance d, the radio expands

$$E_{tx} = l \cdot E + l \cdot \varepsilon_{\text{amp}} \cdot d^2, \tag{6}$$

where ε_{amp} is an amplifier parameter.

To receive l bits message with distance d, the radio expands

$$E_{rx} = l \cdot E. \quad (7)$$

Assume that there are N nodes distributed uniformly in an $M \times M$ region. If there are W clusters, there are on average N/W nodes per cluster including one cluster head node and $(N/W-1)$ cluster member nodes. Each cluster head dissipates energy E_{ch} receiving signals from the nodes

$$\begin{aligned} E_{\text{ch}} &= \left[l \cdot E \cdot \left(\frac{N}{W} - 1\right)\right] + \left[l \cdot E_{\text{DA}} \cdot \frac{N}{W}\right] \\ &\quad + \left[l \cdot E + l \cdot \varepsilon_{\text{amp}} \cdot d^2\right], \end{aligned} \quad (8)$$

where E_{DA} is the energy for data aggregation per signal per bit.

Each cluster member node just needs to transmit its data to the cluster head with energy E_{member}

$$E_{\text{member}} = \left[l \cdot E + l \cdot \varepsilon_{\text{amp}} \cdot d_1^2\right], \quad (9)$$

where d_1 is the distance between the cluster member node and the cluster head.

Assume that the cluster head is at the center of mass of the cluster and the radium is R, then we can get the energy E_{member} and the total energy of the cluster E_{cluster}

$$\begin{aligned} E_{\text{member}} &= \left[l \cdot E + l \cdot \varepsilon_{\text{amp}} \cdot \frac{M^2}{2\pi W}\right] \\ E_{\text{cluster}} &= E_{\text{ch}} + \left(\frac{N}{W} - 1\right) E_{\text{member}}. \end{aligned} \quad (10)$$

Then the total energy E_{total} is

$$E_{\text{total}} = W \cdot E_{\text{cluster}}. \quad (11)$$

Then we can get the optimum number W' of clusters by setting the derivative of with respect to W to zero.

$$W' = \frac{M\sqrt{N}}{\sqrt{2\pi}d}. \quad (12)$$

3.3. Path Formation. The main objective of DIDGA is to improve the efficiency of the data-gathering by the mobile collector using the cluster header to relay sensed data. The data of a sensor node v_i only need to be relayed up to the cluster header CH_{ij}, and the data will be sent to the mobile collector by the cluster header CH_{ij}. With the relay mechanism, the mobile collector can gather data from clusters within its communication range. Therefore, the mobile robot can collect data from all sensor nodes in the network without visiting them individually. After cluster formation phase, the path formation scheme for the mobile collector is equal to resolve a TSP with some constraints. In order to gather data efficiently, the path formation solution should minimize the total traversal distance of the mobile collector with some conditions, such as the cluster headers CH_{ij} with higher priority events should be visited first.

In DIDGA, it is assumed that the network is composed of large number of nodes, which are uniformly deployed over a rectangular area and each sensor node has a unique ID and location information. Assume that the maximal range for the direct communication of the sensor node's radio signal is R_c. In practice, the distance between two sensor nodes that can directly communicate with each other, usually unequal to the communication range R_c. In order to let the mobile collector gather all data in the network so that no sensor node could be unattended, we consider the worst case in wireless communication and define a most suitable value of the hop distance. The hop distance d is

$$d = \frac{R_c}{2}, \quad (13)$$

where R_c is the communication range.

In DIDGA, we assume that the data-gathering path of the mobile collector is random. Before executing the proposed DIDGA algorithm, the mobile collector will perform initial phase. If the locations of sensor nodes are not known in advance, the mobile collector has to traverse the whole sensed field along an exploring path to discover nodes' locations. This procedure has to be executed in the initial deployment and whenever the sink detects that some part of the network has been accidentally destroyed. When exploring the sensed field and whenever a sensor node is encountered, the mobile collector can instruct the sensor node to communicate with the other nodes inside the same MCDS to discover these nodes. In the initial phase, the nodes with the range of the mobile collector will receive the control message directly. Then the sensor nodes will send the location and IDs information of the cluster headers to the collector. After receiving the information, the collector can move to one of the cluster header to gather data and to form the moving path.

With the location information obtained from the initial phase, the mobile collector has to form an optimized path to gather sensed data. In such a scenario, if sensors with huge data attract the mobile collector for a long-time period, data will be dropped from the cache by sensors at the end of the tour in order to accommodate new data. And sensors with urgent event should be visited first. Although network energy is saved in such schemes, quality of services is reduced in terms of response time. Thus, in order to avoid high latency and lost high priority data, an optimized data collection method with the lowest possible latency requirement are necessary to keep the network working at an acceptable quality of service level.

In some applications, sensed data should be delivered to the users according to specific requirements given such as data reporting intervals. The property of such requirements can be either dynamic or static. For the dynamic case, the user will control the sensors' behavior by sending some information depending on the environmental situation or the analysis of sensed data already delivered. For the static case, the sensors may have to decide on the importance of sensed data based on the requirements initially given. For a time critical situation, the sensor network should focus on meeting a delay constraint even though energy consumption is relatively high. Therefore, DIDGA should be

Table 1: Notations.

Notation	Description
x_{ij}^t	binary decision variable
(c_{ij})	distance matrix of all cluster headers in V
r_{ij}^t	running time of path $\langle i, j\rangle$ at time t
w_i	waiting time of mobile collector arrived at cluster header CH_i early than the lower bound of CH_i
d_i	waiting time of mobile collector arrived at CH_i later than the upper bound CH_i
d_{0i}	time to depart from the sink to CH_i
s_i	service time at node i
e_i	beginning boundary of the time window at CH_i
l_i	ending boundary of the time window at CH_i
c_{ij}	minimum travel time among all possible discrete states between CH_i and CH_j
χ	weight associated with mobile collector waiting for cluster header
δ	weight associated with cluster header waiting for mobile collector
q_i	demand at node v_i
N	number of nodes
N_0	union of set N and the sink

able to flexibly adapt to the varying users' requirements while maximizing energy conservation for the network longevity. DIDGA helps sensors to maximize energy conservation in reporting their sensed data by providing two types of data forwarding link: data relay link and direct link.

In this paper, we consider the NP path formation problem with dynamic requirements under dynamic network environment. The characteristics of the problem can be described in terms of sink, the high priority cluster header with urgent event, and the time dependent between requirement nodes. The mobile collector has to send the time-limited data to the sink before the end of the event expired. Every requirement node has its own time window. The mobile collector is allowed to arrive at a requirement node outside of the time interval defined for service. However, there would be a penalty when the arriving time of the mobile collector violates the time window. On one hand, the running time $f_1(x)$ of the mobile collector is related to cluster headers' waiting time $f_2(x)$ and collector's waiting time $f_3(x)$. On the other hand, the running time is also determined by path and the priority of cluster header with time-limited data. So the objective of the path formation is to minimize the weighted sum of $f_1(x)$, $f_2(x)$ and $f_3(x)$ for the given constrained conditions, which will meet the requirements of cluster headers with less power and real-time urgent events. The notations used to describe the scheme are shown as in Table 1.

The value of x_{ij}^t is 1 if the mobile collector can gather data from cluster header i to j between time window $[T_t, T_{t+1}]$, otherwise 0. χ and δ are two weights in $[0,\ 1]$.

The proposed path formation scheme is formulated as follows:

$$\min f(\vec{x}) = \min\Big(f_1(x) + \chi f_2(x) + \delta f_3(x)\Big), \quad (14)$$

where $f_1(x)$ is the running time of mobile collectors, $f_2(x)$ is the waiting time of mobile collectors at cluster headers, and $f_3(x)$ is the waiting time of cluster headers. And $f_1(x)$, $f_2(x)$, and $f_3(x)$ are delineated by the following constraints:

$$\begin{gathered} f_1(x) = \sum_{t=0}^{T}\sum_{i\in V}\sum_{j\in V} c_{ij} r_{ij}^t x_{ij}^t, \\ f_2(x) = \sum_{i\in V} w_i, \\ f_3(x) = \sum_{i\in V} d_i\ , \\ \sum_{j\in N_0, j\neq i} x_{ij}^t = 1, \quad \forall i \in N, \\ \sum_{j\in N_0, j\neq i} x_{ij}^t = 1, \quad \forall j \in N, \\ \sum_{j\in N} x_{0j}^t \le f_1(\vec{\mathbf{x}})\ , \\ s_j + \max\Big(d_i + c_{ij}, e_j\Big) \le d_j, \quad \forall i,\ j \in N\ , \\ d_{0i} \ge e_0\ , \end{gathered} \quad (15)$$

where the waiting times and excess duration caused by the violation of time windows or departure time plan are defined in the following and can be easily calculated once the first-stage decision is determined and the stochastic travel times are realized. Note that symbol c_{ij} represents the minimum travel time among all states of the stochastic travel time between cluster headers CH_i and CH_j, which forms a route among cluster headers without the sink in it. Note that the logical expression (15) can easily be transformed into linear expressions by introducing a sufficiently large constant. The stochastic travel time c_{ij} could be either continuous or discrete in nature. Since the continuous stochastic programming model is difficult to solve, we suppose c_{ij} is a random variable defined by m discrete states of a stochastic travel time.

The objective (15) is to minimize the weighted sum of expected travel time, expected waiting time, and expected penalties, which will meet the requirements of cluster headers with less power and high priority data. Considering the ant colony algorithms have the characteristic of good local searching capability while the evolutionary algorithms have fairly good global searching performance and, to solve (15), the proposed path formation optimized algorithm (PFOA) combines ant colony algorithm and evolutionary algorithm so as to satisfy the constraints. Herein, the solution PFOA algorithm is formally proposed as in Algorithm 3.

From the Algorithm 3, we know that the best path is determined by the current pheromone matrix. So the update way of the pheromone matrix will affect the efficiency of

Input: All sensors v_i, $i \in [1, N]$
Output: optimized pheromone matrix C
Initialization;
Set the maximum iteration number $I_{\max}$;
Set the maximum iterative number $I_{\max,\text{ant}}$ of ant colony algorithm, and the population size P_{ant} of ant colony algorithm;
Set the maximum iteration number $I_{\max,\text{evo}}$ of evolutionary algorithm, the population size P_{evo} of evolutionary algorithm;
Create the pheromone matrix;
Initialize the pheromone matrix by evolutionary algorithm, update Pareto candidate solution set;
while *not satisfying the stopping criterion* **do**
  Update the pheromone matrix by ant colony algorithm;
  Optimize pheromone matrix by evolutionary algorithm;
end
return the optimized pheromone matrix C;

Algorithm 3: Path formation optimized algorithm.

Input: All sensors v_i, $i \in [1, N]$
Output: optimized pheromone matrix C
Randomly generate $E_M - 1$ individuals and set the current generation $G_C = 0$;
Evaluate the $E_M - 1$ individuals;
while $G_C \leq I_{\max}$ **do**
  Select E_M individuals;
  Encoding the E_M individuals;
  Crossover the E_M individuals, and evaluate the $E_M - 1$ individuals;
  Mutate the E_M individuals, and evaluate the $E_M - 1$ individuals;
  Select the best E_M individuals from the two generations as the new population;
  $G_C = G_C + 1$;
end
return optimized pheromone matrix C;

Algorithm 4: Pheromone matrix optimized algorithm.

PFOA. Pheromone will be updated by a process of global update given as follows:

$$\tau_{ij}(t+1) = \max\left(\rho \cdot \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}, \tau_{\min}\right), \qquad (16)$$

where $\rho \in [0, 1]$ is the trail persistence, m is the number of ants, and $\Delta\tau_{ij}^{k}$ is the amount of pheromone laid by the k-th ant on edge (ij). The pheromone matrix is optimized by Algorithm 4.

When evaluating individual in the algorithm, we generate ants and calculate the pareto-dominate relationship between ants and the set of pareto candidate solution. When an ant is generated, no matter generated by evolutionary algorithm or generated during the iterating process of ant colony algorithms, the updating strategy of pareto candidate solution set remains the same. That is, if the ant is not dominated by any individual in the set, and the pareto candidate solution set is not full, add it into the set; otherwise, if the ant is not dominated but the set is full, it will be replaced with the closest candidate solution from this ant by Hamming distance.

Input: All sensors v_i, $i \in [1, N]$
Output: Sensed data set *Data*
Initialization;
Create MISs by Algorithm 1;
Construct MCDSs by 2;
Perform PFOA by Algorithm 3 and pheromone matrix optimized Algorithm 4;
for *each cluster head* CH_i **do**
  if *Data is critical data* **then**
    Sort *Data* in ascending order by $t_{\exp} - Data.t_w$;
    Send *Data* to sink directly;
  end
  else
    Save *Data*;
  end
end
return Sensed data set *Data*;

Algorithm 5: Distributed intelligent data-gathering algorithm.

First, we defined a relation sequence R_{ij}, representing the relationships mentioned above, which has m elements if there exist m travel time states between a pair of consecutive nodes v_i and v_j in the route. For example, if the relationships between the departure times in the pair of consecutive cluster headers CH_i and CH_j are characterized by 3 different travel time states t_1, t_2, and t_3, then the relation sequence can be written as $R_{ij} = \{t_1, t_2, t_3\}$. What we intend here is to prove the convergence of the proposed PFOA algorithm by showing the following facts:

For path $i \leftrightarrow j$, the parameters in (14) and (15) can be determined according to R_{ij} of the current solution. If the R_{ij} associated with the current solution appears for the first time during the solution process, an optimal cut corresponding to R_{ij} is added to the constraint set. For path $i \leftrightarrow j$, if the relation sequence R_{ij} associated with the current solution is the same as any previous solution, the optimal cut corresponding to this relation sequence must have been added already and the second/third item in the original objective will be equal to its lower bound subject to cluster header CH_i or CH_j. As a result, there is no need to add the same cut again.

After determining the data-gathering path for the mobile collector with the help of the sensor nodes with distributed manner, the mobile collector performs the proposed DIDGA that is shown as in Algorithm 5.

In the system, the sensed data is classified in two types: critical and noncritical data. For critical data, an expiration time $t_{\exp}$ is preset manually. And the critical data has higher priority than the noncritical data in data-gathering in CH_i. For the delay constraint event and time critical data, CH_i will sort the *Data* in ascending order according to the value $t_{\exp} - Data.t_w$, where $Data.E_t$ denotes the actual waiting time. And the collector will report it to sink directly with higher priority. The proposed DIDGA guarantees a uniform level in term of expiration time for all critical data. There are two criteria applied to the proposed scheme. And the collector will transmit other sensed data to sink after visiting all cluster headers. Then the mobile collector will start the next round data-gathering.

4. Simulation Results

In order to evaluate the performance of the proposed DIDGA algorithm, we implemented the DIDGA in the well-known simulation tool NS-2 [35], the well-planned adaptive moving strategy (AMS) in [33] and the algorithm without mobile collector (WMC) are simulated as discussed here. Our simulation is performed considering the deployed region as a square of fixed area of 1000 m × 1000 m. In order to avoid the communication holes among the nodes in the network, the distance between any two nodes are maintained as R_c, which is equal to the communication range between two nodes. Besides, the distance between any two layers of nodes is considered as $\sqrt{3}R_c/2$, so that each three nodes in the network can be vertices of an equilateral triangle. Though minimum 725 nodes deployed regularly as described above, some more nodes are deployed randomly as per requirement. Thus, the deployment strategy totally ensures the assumption that there is no communication hole in the network. And the system generates critical and noncritical events randomly. The performance analysis has been done by deploying variable number of nodes on the fixed squared area, to verify the effect of node densities on the data-gathering path length and average number of hop counts. All nodes use the CSMA protocol for channel access and broadcasting the control packets.

In the experimental system, we assume the dynamic requirement of sensor node follows the Poisson process. The dynamic requirement R of customers includes three types: $R = \{R_1, R_2, R_3\}$, where R_1, R_2, and R_3 denote the $edod\_TW \in [0, 0.4]$, $edod\_TW \in [0.4, 0.6]$, and $edod\_TW \in [0.6, 0.8]$, respectively. The velocity W of the mobile collector also includes three kinds: $W = \{W_1, W_2, W_3\}$, where W_1, W_2, and W_3 denote low, middle and high speed, respectively. We set $I_{\max} = 100$, $E_M = 5$, $\alpha = 3$, $\rho = 0.8$, $\tau_{\min} = 0.001$, $p_c = 0.5$, $p_m = 0.1$. The problem set includes three kinds $S = \{S_1, S_2, S_3\}$, where S_1, S_2, and S_3 denote the case $N \in [400, 800]$, $\lambda \in [1, 2]$, $N \in [800, 1200]$, $\lambda = 2$, and $N \in [1200, 1600]$, $\lambda = 3$, respectively. The scheduling time is 10. DIDGA will randomly generate the traffic graph with nodes in the square field. The average value of service time is 0.3, and the square error is 0.2. Our simulation results are all from the average of 1000 runs.

4.1. Validation of Path Formation. In order to explore the weights of χ and δ of influence on path length, Figure 2 shows the results of total path length with varying weights χ and δ, where the number of nodes is 1200. We can see that χ and δ have some influence on the path length, because mobile collector will change the path when the cluster headers' waiting time and collector's waiting time vary, and the collector has to take the priority of cluster header with time-limited data and less power into consideration. And it is difficult to obtain the optimized path length. Hence, we can only get the suboptimal path length with the appropriate χ and δ. In the following scenarios, we set the values of χ and δ as 0.34 and 0.67, respectively.

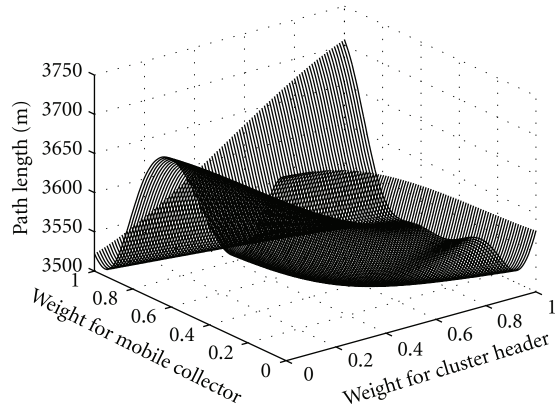


Figure 2: Path length with varying weights.

With the number of nodes increasing, the path length of two schemes will increase obviously. When the number of nodes increases from 800 to 1600, the path length of AMS increases to 4367 m, which is slightly lower than that of DIDGA. We can see that the path length of DIDGA always is slightly higher than that of AMS. The reason is that the objective of the path formation is to minimize the weighted sum of collector's running time, cluster headers' waiting time, and collector's waiting time for the given constrained conditions. So in path formation scheme, DIDGA considers the dynamic requirements such as high priority urgent event, data reporting intervals, and the time critical situation into consideration. Such constrains slightly increase the total length of mobile collector to gather data.

In Figure 3, the total path length of different algorithms is shown. With the number of nodes increasing, the path length of two schemes will increase obviously. When the number of nodes increases from 800 to 1600, the path length of AMS increases to 4367 m, which is slightly lower than that of DIDGA. We can see that the path length of DIDGA always is slightly higher than that of AMS. The reason is that the objective of the path formation is to minimize the weighted sum of collector's running time, cluster headers' waiting time, and collector's waiting time for the given constrained conditions. So in path formation scheme, DIDGA considers the dynamic requirements such as high priority urgent event, data reporting intervals, and the time critical situation into consideration. Such constrains slightly increase the total length of mobile collector to gather data.

Figure 4 shows the data-gathering time by different schemes when the number of nodes varies from 800 to

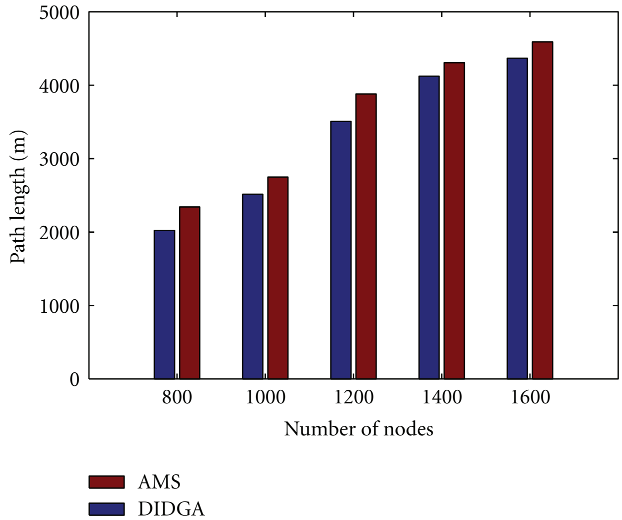


Figure 3: Path length of different algorithms.

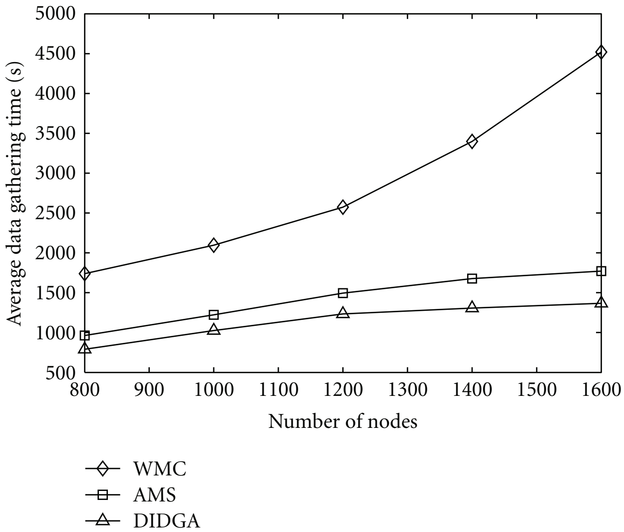


Figure 4: Average data-gathering time of different algorithms.

1600. We compare three schemes: without mobile collector (WMC), the adaptive moving strategy (AMS), and DIDGA. It can be seen that data-gathering time of all the schemes increases as number of nodes increases. For WMC, the average data-gathering time increases greatly, from 1740 s (800 nodes) to 4521 s (1600 nodes). The reason is that the sink must wait the relayed data by nodes as the number of nodes increasing, which will increase the data-gathering time obviously. However, DIDGA always outperforms other schemes due to the concurrent use of the mobile collector and simultaneous data uploading among sensors with the support of cluster headers. For instance, it achieves 73% time saving compared with WMC scheme when number of nodes is 1600. Shorter data-gathering time leads to longer network lifetime since sensors can turn to power-saving mode once the data-gathering in their region is done. We also notice that the advantage of DIDGA over AMS becomes more evident

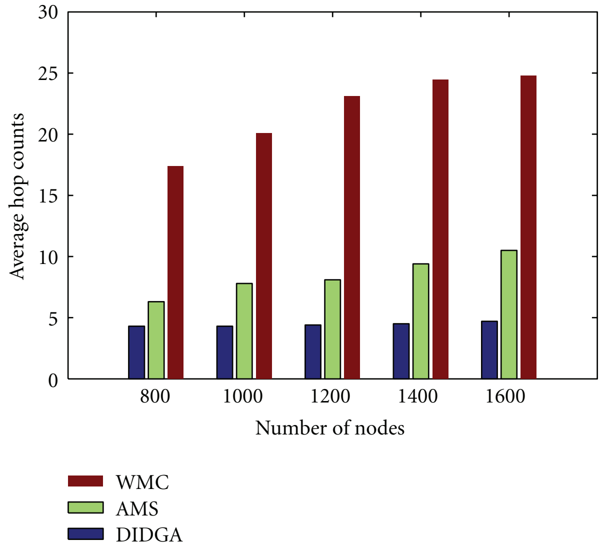


Figure 5: Average hop counts with different algorithms.

when the network becomes denser with more sensors. This is reasonable because more sensors would provide more opportunities to utilize cluster headers for concurrent data uploading.

4.2. Validation of DIDGA. Figure 5 shows the results of average hop counts for different schemes with different node numbers. As shown in Figure 5, it is observed that DIDGA outperforms in terms of average hop counts irrespective of the position of the sink. For DIDGA, the average hop count keeps about 4.7, while for WMC and AMS, the average hop count is 20.1, 8.7. It is noticed that the number of nodes has a great impact on the average hop counts for WMC. It is reasonable since more nodes mean that the sensed data will be relayed by more hops. On the contrary, when the number of nodes is large, the impact of node number on average hop counts is not obvious for DIDGA. For example, when the number of nodes increases to 1600, the average hop count of DIDGA just increases to 4.7, which is smaller than that of AMS about 5.8. The reason is that all sensor nodes will form clusters and just need to relay the sensed data to the cluster headers, so the average hop counts are not affected by node numbers.

In order to show the overall trend of energy consumption, we sorted the remaining energy levels collected from each cluster member and cluster head. Figures 6 and 7 show the comparison results of remaining energy level of each sensor and cluster head for AMS, WMC, and the proposed DIDGA, respectively. As shown in Figures 6 and 7, it is observed that the proposed DIDGA outperforms in terms of remaining energy irrespective of the number of cluster nodes or cluster headers. The remaining energy level of WMC is obviously lower than that of DIDGA. Because that the nodes have to relay sensed data to the sink. For the algorithm AMS, the remaining energy increases to 0.92 when the number of cluster members increases to 1200, while for DIDGA, the remaining energy of DIDGA is 0.952. The reason is that the

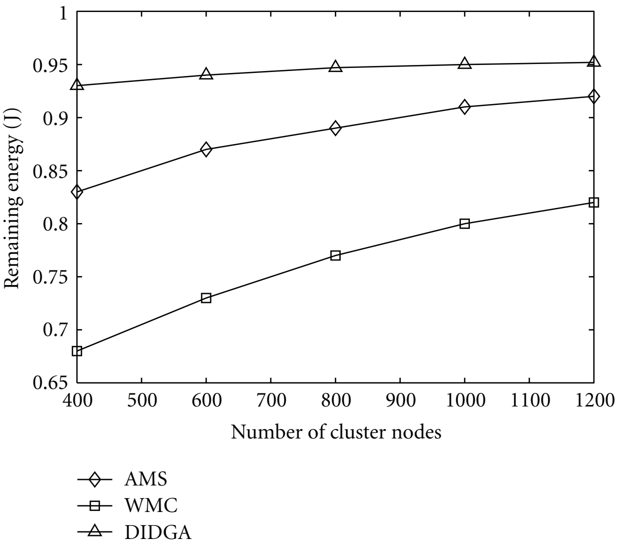


Figure 6: Remaining energy of cluster members.

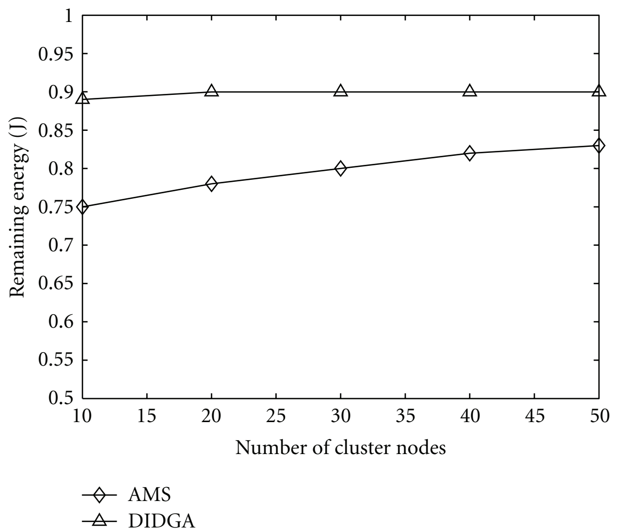


Figure 7: Remaining energy of cluster heads.

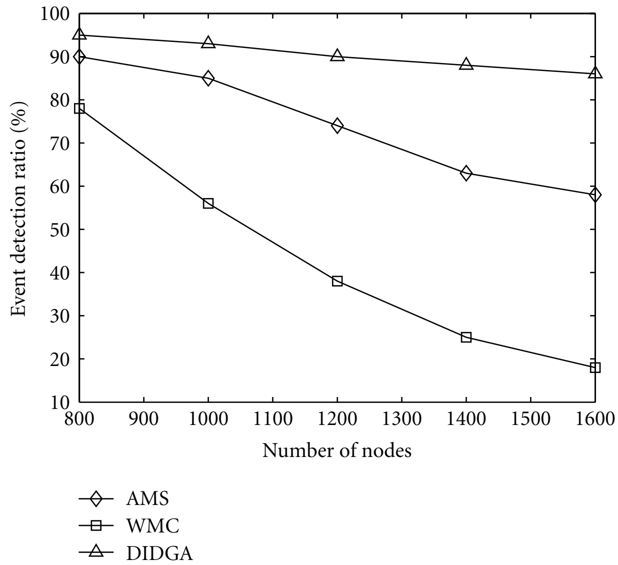


Figure 8: Event detection ratio of different algorithms.

sensor nodes just need to send their packets to its neighbor cluster header to save energy. And the proposed scheme not only considers nodes topology, energy information but also hop distance to known nodes, and it uses mobile collector to gather data. As mentioned previously, the radio range is a factor affecting the transmission distance which is proportional to the energy consumption. Thus, we can observe that Figure 6 shows a much higher energy-savings compared to the one in Figure 5. In order to meet the delay constraint, some cluster members have to use more energy using the direct link even though they have a data relay point.

Figure 8 shows the results of event detection ratio of different algorithm, where the expiration time $t_{\exp}$ of critical data is 3600 s. The event detection ratio is the ratio of sink that successfully received the real-time event and time-limited event data before the data expired and all events. As shown in Figure 8, it is observed that DIDGA outperforms in term of event detection ratio irrespective of the number of nodes. As the number of nodes increasing, the error between DIDGA and other algorithm will increase greatly. For example, the event detection ratio of DIDGA is 86%, which is higher than AMS about 28%. For the WMC, the ratio greatly decreases to 18%. The reason is that the sink is located any location, the sensor nodes have to route data in several hops, thereby leads to more delay and energy when the number of nodes increases. And relay scheme also leads to the energy exhaustion of some nodes, which makes some events cannot be detected and relayed, especially for time-sensitive that data. For WMC and DIDGA, the reason is that more nodes will lead to longer path length, which will increase the data-gathering time and result in the expiration of some time-limited events. For the proposed DIDGA, the event detection ratio decreases slightly when the number of nodes increases. The reasons are as follows: firstly, DIDGA will gather and report the delay constraint event and time critical data with higher priority, and it also sends the *Data* in ascending order by $t_{\exp} - Data.t_w$, which also guarantees a uniform level in term of expiration time for all critical data; secondly, DIDGA can reduce the power consumption of nodes with the cluster head relay mechanism, which will prolong the lifetime of all sensor nodes to monitor environment and detect events; finally, DIDGA can minimize the weighted sum of collector's running time, cluster headers' waiting time, and collector's waiting time for the given constrained conditions, which not only meets the requirements of real-time urgent events but also decreases the data-gathering time. The results also prove that DIDGA can greatly improve event monitor and detection performance in WSN.

The results of event detection ratio of different algorithms with the varying expiration time are shown in Figure 9, where the node number is 1200. As the expiration

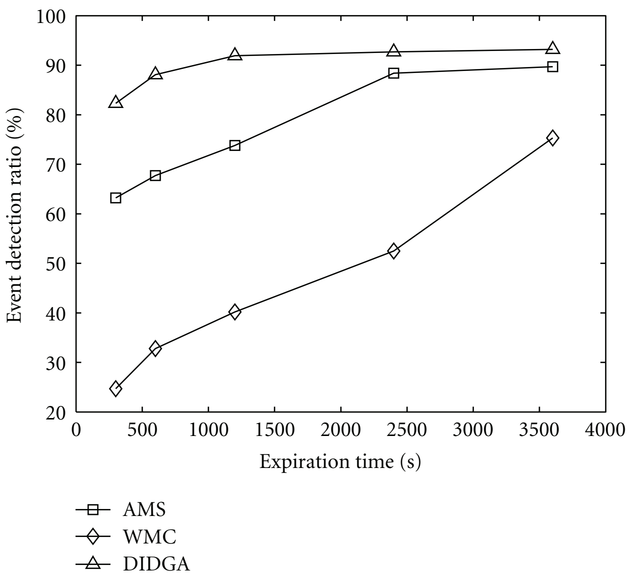


Figure 9: Event detection ratio of different expiration time.

time increasing, the event detection ratio of different algorithms will increase obviously. Especially for WMC, the event detection ratio increases to 75.3%. The main reason is that WMC collects by nodes relay, which leads to longer delay time for remote nodes to transmit the real-time urgent events to the sink, and increasing expiration time will decrease the overdue data. So the expiration time has a great influence on WMC. The event detection ratio of AMC is higher than that of WMC clearly. Because AMS employs the mobile sink to collect sensed data, which will decrease time of data-gathering and collect more real-time event and sensed data. For the proposed DIDGA, the event detection ratio is higher than that of AMS. For example, when the expiration time is 300 s, the event detection ratio of DIDGA is 82.3%, which is higher than that of AMS about 20.9%. When the expiration time increases to 3600 s, the event detection ratio of DIDGA increases to 93.2%. Compared with AMC, DIDGA distinguishes the critical and noncritical data with different priorities and gathering schemes and reports the delay constraint event and time-critical data with higher priority. And the power saving and path formation schemes of DIDGA also help to prolong the lifetime of all sensor nodes to detect events successfully.

5. Conclusion and Future Work

In order to reduce the high energy expenditure in multihop routing and extend lifetime in distributed WSNs, an efficient distributed intelligent data-gathering algorithm called DIDGA is also proposed for the mobile collector. A mobile collector is employed to gather the sensed data from nodes by dividing the whole network into certain MCDS to minimize it by reducing the number of hops in the network. A path formation optimized algorithm (PFOA) is also proposed which combines ant colony algorithm and evolutionary algorithm to satisfy the time-limited constraints. Detailed simulation results and comparisons with other algorithms show that DIDGA not only decreases average hop counts, average data-gathering time, but also improves event detection ratio, saves energy consumption of sensor nodes, and greatly extends the network lifetime.

Future work will address collaborative in-network processing to provide the required processing power not available in standalone sensor nodes. With this approach, the communication scheduling will choose reliable links and balance communication load among cluster nodes, which will increase the communication reliability and the network lifetime. Considering that the values of χ and δ have some influence on the path length, we'll try to find optimized weights χ and δ to decrease the path length.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 60902053), the Science and Technology Research Planning of Educational Commission of Hubei Province of China (No. B20110803), and the Natural Science Foundation of Hubei Province (No. 2008CDB339). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers.

References

[1] C. Y. Chong and S. P. Kumar, “Sensor networks: evolution, opportunities, and challenges,” *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[3] K. Kalpakis, “Everywhere sparse approximately optimal minimum energy data gathering and aggregation in sensor networks,” *ACM Transactions on Sensor Networks*, vol. 7, no. 1, 2010.

[4] L. Shu, Y. Zhang, Z. Zhou, M. Hauswirth, Z. Yu, and G. Hynes, “Transmitting and gathering streaming data in wireless multimedia sensor networks within expected network lifetime,” *Mobile Networks and Applications*, vol. 13, no. 3-4, pp. 306–322, 2008.

[5] F. Wang, J. Liu, and L. Sun, “Ambient data collection with wireless sensor networks,” *Eurasip Journal on Wireless Communications and Networking*, vol. 2010, 2010.

[6] J. Wang, Y. Liu, and S. K. Das, “Energy-efficient data gathering in wireless sensor networks with asynchronous sampling,” *ACM Transactions on Sensor Networks*, vol. 6, no. 3, article 22, 2010.

[7] P. Santi, “On the data gathering capacity and latency in wireless sensor networks,” *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 7, pp. 1211–1221, 2010.

[8] S.-H. Hong and B.-K. Kim, “An efficient data gathering routing protocol in sensor networks using the integrated gateway node,” *IEEE Transactions on Consumer Electronics*, vol. 56, no. 2, pp. 627–632, 2010.

[9] H. Zhang and H. Shen, “Balancing energy consumption to maximize network lifetime in data-gathering sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 10, pp. 1526–1539, 2009.

[10] X. Chen, X. Hu, and J. Zhu, "Data gathering schedule for minimal aggregation time in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 5, no. 4, pp. 321–337, 2009.

[11] K. Yuen, B. Liang, and B. Li, "A distributed framework for correlated data gathering in sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 1, pp. 578–593, 2008.

[12] Y. Yu, B. Krishnamachari, and V. K. Prasanna, "Data gathering with tunable compression in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 2, pp. 276–287, 2008.

[13] Y. Revah and M. Segal, "Improved bounds for data-gathering time in sensor networks," *Computer Communications*, vol. 31, no. 17, pp. 4026–4034, 2008.

[14] H.-C. Lin, F.-J. Li, and K.-Y. Wang, "Constructing maximum-lifetime data gathering trees in sensor networks with data aggregation," in *Proceedings of the IEEE International Conference on Communications (ICC '10)*, 2010.

[15] J. C. Dagher, M. W. Marcellin, and M. A. Neifeld, "A theory for maximizing the lifetime of sensor networks," *IEEE Transactions on Communications*, vol. 55, no. 2, pp. 323–332, 2007.

[16] R. Zhu, "Efficient fault-tolerant event query algorithm in distributed wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2010, Article ID 593849, 7 pages, 2010.

[17] M. Zhao and Y. Yang, "An optimization based distributed algorithm for mobile data gathering in wireless sensor networks," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '10)*, March 2010.

[18] F. J. Wu, C. F. Huang, and Y. C. Tseng, "Data gathering by mobile mules in a spatially separated wireless sensor network," in *Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware (MDM '09)*, pp. 293–298, May 2009.

[19] C. L. Hwang and N. W. Chang, "Fuzzy decentralized sliding-mode control of a car-like mobile robot in distributed sensor-network spaces," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 1, pp. 97–109, 2008.

[20] F. El-Moukaddem, E. Torng, and G. Xing, "Maximizing data gathering capacity of wireless sensor networks using mobile relays," in *Proceedings of the IEEE 7th International Conference on Mobile Adhoc and Sensor Systems (MASS '10)*, pp. 312–321, 2010.

[21] S. Jain, R. C. Shah, and W. Brunette, "Exploiting mobility for energy efficient data collection in wireless sensor networks," in *Proceedings of the IEEE Workshop on Modeling and Optimization in Mobile Ad hoc and Wireless Networks*, 2004.

[22] S. Susca, F. Bullo, and S. Martinez, "Monitoring environmental boundaries with a robotic sensor network," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 288–296, 2008.

[23] J. Culpepper, L. Dung, and M. Moh, "Hybrid indirect transmissions (HIT) for data gathering in wireless micro sensor networks with biomedical applications," in *Proceedings of the IEEE Annual Workshop on Computer Communications*, pp. 124–133, 2003.

[24] D. Mandala, X. Du, F. Dai, and C. You, "Load balance and energy efficient data gathering in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 8, no. 5, pp. 645–659, 2008.

[25] J. P. Sheu, P. K. Sahoo, C. H. Su, and W. K. Hu, "Efficient path planning and data gathering protocols for the wireless sensor network," *Computer Communications*, vol. 33, no. 3, pp. 398–408, 2010.

[26] D. Jea, A. A. Somasundara, and M. B. Srivastava, "Multiple controlled mobile elements (data mules) for data collection in Sensor Networks," in *International Conference on Distributed Computing in Sensor Systems (DCOSS '05)*, June 2005.

[27] Y. Wu, S. Fahmy, and N. B. Shroff, "On the construction of a maximum-lifetime data gathering tree in sensor networks: NP-completeness and approximation algorithm," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 1013–1021, April 2008.

[28] Y. Kishino, Y. Sakurai, K. Kamei, Y. Yanagisawa, T. Maekawa, and T. Okadome, "Data gathering in high-density wireless sensor networks using hierarchical clustering," in *Proceedings of the IEEE International Symposium on Wireless Communication Systems (ISWCS '08)*, pp. 547–551, October 2008.

[29] Y. Revah and M. Segal, "Improved algorithms for data-gathering time in sensor networks II: Ring, Tree, and Grid topologies," *International Journal of Distributed Sensor Networks*, vol. 5, no. 5, pp. 463–479, 2009.

[30] C. Luo, F. Wu, J. Sun, and C. W. Chen, "Compressive data gathering for large-scale wireless sensor networks," in *Proceedings of the 15th Annual ACM International Conference on Mobile Computing and Networking (MobiCom '09)*, pp. 145–156, September 2009.

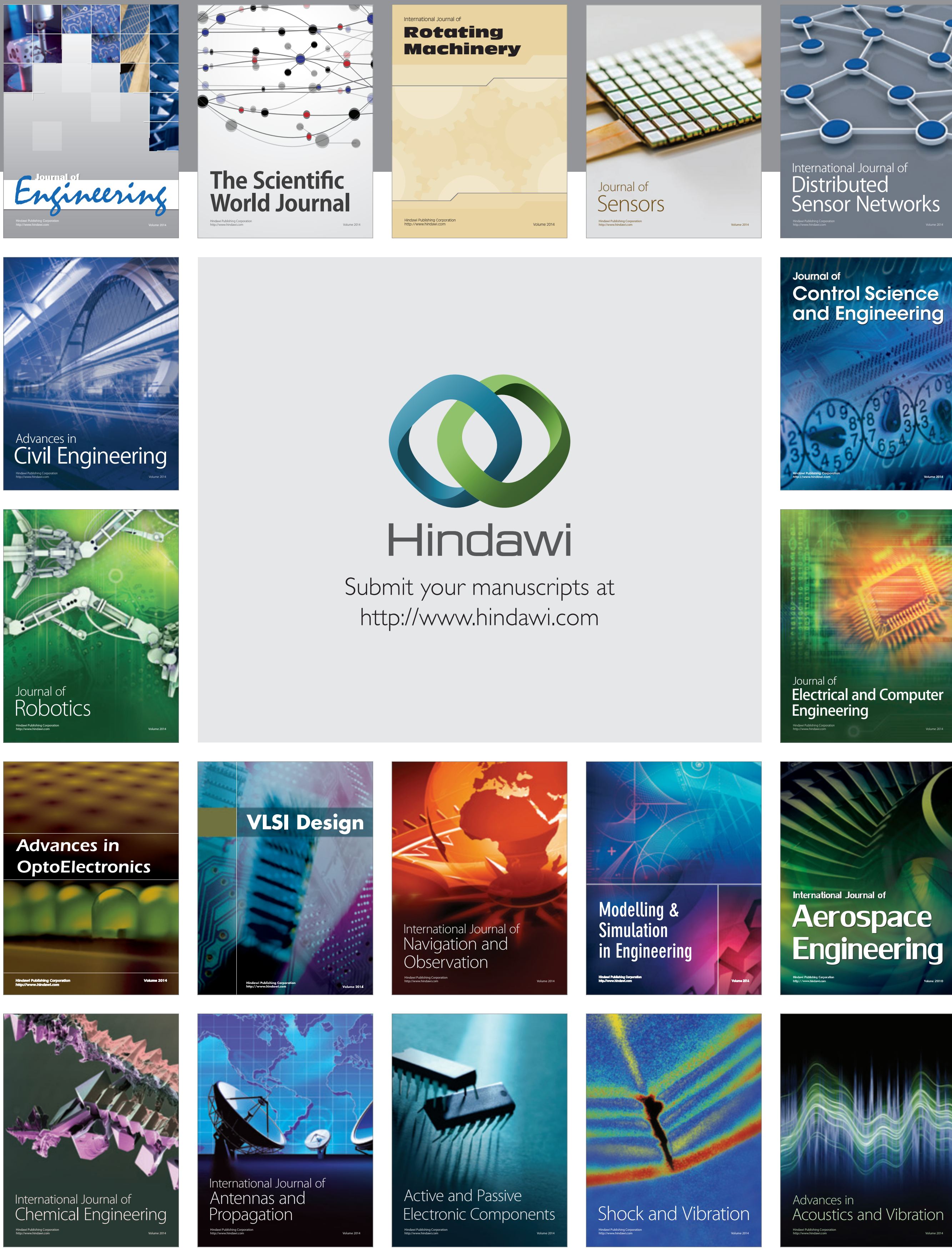
[31] M. Ma and Y. Yang, "Data gathering in wireless sensor networks with mobile collectors," in *Proceedings of the 22nd IEEE International Parallel and Distributed Processing Symposium (IPDPS '08)*, April 2008.

[32] A. Boukerche and X. Fei, "Adaptive data-gathering protocols with mobile collectors for vehicular ad-hoc and sensor networks," in *Proceedings of the 4th IEEE International Conference on Wireless and Mobile Computing, Networking and Communication (WiMob '08)*, pp. 7–12, 2008.

[33] E. M. Saad, M. H. Awadalla, and R. R. Darwish, "A data gathering algorithm for a mobile sink in large-scale sensor networks," in *Proceedings of the 4th International Conference on Wireless and Mobile Communications (ICWMC '08)*, pp. 207–213, August 2008.

[34] G. F. Zaki, H. M. Elsayed, H. H. Amer, and M. S. El-Soudani, "Energy balanced model for data gathering in wireless sensor networks with fixed and mobile sinks," in *Proceedings of the 18th International Conference on Computer Communications and Networks (ICCCN '09)*, August 2009.

[35] The Network Simulator. NS-2, http://www.isi.edu/nsnam/ns/.


Journal of Engineering
The Scientific World Journal
International Journal of Rotating Machinery
Journal of Sensors
International Journal of Distributed Sensor Networks
Advances in Civil Engineering
Journal of Control Science and Engineering
Hindawi
Submit your manuscripts at
http://www.hindawi.com
Journal of Robotics
Journal of Electrical and Computer Engineering
Advances in OptoElectronics
VLSI Design
International Journal of Navigation and Observation
Modelling & Simulation in Engineering
International Journal of Aerospace Engineering
International Journal of Chemical Engineering
International Journal of Antennas and Propagation
Active and Passive Electronic Components
Shock and Vibration
Advances in Acoustics and Vibration