Research Article

Constructing a CDS-Based Network Backbone for Data Collection in Wireless Sensor Networks

Xiaoyan Kui, 1 Yu Sheng, 1 Huakun Du, 2 and Junbin Liang 3

1 School of Information Science and Engineering, Central South University, Changsha 410083, Hunan, China
2 School of Geoscience and Info-Physics, Central South University, Changsha 410083, Hunan, China
3 School of Computer and Electronic Information, Guangxi University, Nanning 530004, Guangxi, China

Correspondence should be addressed to Yu Sheng; shengyu@csu.edu.cn

Received 2 January 2013; Accepted 19 March 2013

Academic Editor: Mianxiong Dong

Copyright © 2013 Xiaoyan Kui et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Data collection is one of the most important operations in wireless sensor networks. Currently, many researches focus on using a connected dominating set to construct a virtual backbone for data collection in WSNs. Most researchers concentrate on how to construct a minimum connected dominating set because a small virtual backbone incurs less maintenance. Unfortunately, computing a minimum size CDS is NP-hard, and the minimum connected dominating sets may result in unbalanced energy consumption among nodes. In this paper, we investigate the problem of constructing an energy-balanced CDS to effectively preserve the energy of nodes in order to extend the network lifetime in data collection. An energy-balanced connected dominating set scheme named DGA-EBCDS is proposed, and each node in the network can effectively transmit its data to the sink through the virtual backbone. When constructing the virtual backbone in DGA-EBCDS, we prioritize selecting those nodes with higher energy and larger degree. This method makes the energy consumption among nodes more balanced. Furthermore, the routing decision in DGA-EBCDS considers both the path length and the remaining energy of nodes in the path; it further prolongs the lifetime of nodes in the backbone. Our conclusions are verified by extensive simulation results.

1. Introduction

Wireless sensor network (WSN) is a new hot spot in current research, and it has broad application foreground [1, 2]. A wireless sensor network consists of a large number of nodes that collaborate together to monitor various phenomena. WSN can be used in various fields, such as battlefield, environment monitoring, disaster forecasting, business, and traffic control.

However, the communication ability, computational ability, storage ability, and energy of sensor nodes are limited [3–6], and the nodes are difficult to be replaced because they are often deployed in remote or inaccessible environments; the limitations of wireless sensor networks also shorten the performance of the network. Because of the limited energy, the signal generated by a source can only reach the nodes that are within its transmission range. If two nodes are within the transmission range of each other, they can exchange messages directly; otherwise, the messages have to be relayed through other intermediate nodes [7, 8]. This problem motivates us to construct a virtual backbone in wireless sensor network, and through this virtual backbone each node of the network can effectively transmit its data to the sink.

Wireless sensor networks (WSNs) are data oriented and are usually densely deployed in a monitor environment to process a great deal of data. Data collection is one of the most important operations in wireless sensor networks [9–13], which means that the data sensed by nodes should be transmitted to the sink for further processing. How to conserve the limited energy of nodes and extend the lifetime of the network is an important issue in data collection [14, 15]. The network lifetime is usually defined as the duration of the network until the first node depletes its energy. So, the network lifetime effectively ends with the first node death (FND). Due to incomplete data, the remaining energy in the surviving nodes is of no use after FND. In order to effectively extend the lifetime of a WSN (delay the death of the first node), many algorithms construct a virtual backbone of the
network and only use the nodes in the virtual backbone to receive and transmit data for data collection. Nodes that are not in the backbone can go to sleep mode periodically to conserve the limited energy. Because of the number of nodes in the virtual backbone is relatively small, the impact of the broadcast storm problem can be greatly reduced, and the routing path search space can also be limited to the nodes that are in the set of the backbone. So, the virtual backbone can bring several benefits to network management.

Connected dominating set (CDS) plays an important role in the construction of virtual backbone in wireless sensor networks [16]. A dominating set (DS) of a network is a subset of all its nodes, which makes every node out of the set adjacent to at least one node in the set. A dominating set is called a connected dominating set when the nodes in the dominating set can form a connected graph. A CDS can be used as a virtual backbone to help each node transfer its data to the sink. Many researches focus on finding the minimum connected dominating set (MCDS) to construct the network virtual backbone [10]. However, MCDS-based algorithms may result in too much energy consumption of nodes in the backbone, and may cause unbalanced energy consumption among all the nodes, which will shorten the network lifetime and narrow their applications.

Motivated by the above problem, an energy-balanced connected dominating set scheme (DGA-EBCDS) is proposed in this paper, which well balances energy consumption among nodes that are in the backbone and consequently prolongs the lifetime of the network. In DGA-EBCDS, we prioritize selecting nodes that have higher energy and larger degree to form the backbone. By transmitting data through the backbone with small routing cost, each node can preserve its energy effectively. Moreover, the nodes in the virtual backbone would not die quickly because of their high energy level. The proposed algorithm creates a small-size connected dominating set, which performs well in energy saving and efficiently affords more numbers of rounds in data collection.

The main contributions of this paper are summarized as follows.

First, we introduce the concept of energy-balanced connected dominating set in Section 3. Using this scheme, DGA-EBCDS can well balance energy consumption among nodes and consequently prolong the lifetime of the network.

Second, we theoretically analyze DGA-EBCDS algorithm in Section 4. Theoretical analyses prove that the set of the dominators is a maximal independent set (MIS) when the first stage for DGA-EBCDS terminates and the message complexity of DGA-EBCDS is \( O(n \log n) \).

Third, we support our algorithm analysis with extensive simulations in Section 5. Compared with another dominating set-based algorithm mr-CDS [17], the average energy consumption of DGA-EBCDS is reduced by 33.3% in the formation of CDS, and the network lifetime is prolonged by 57.8%.

The rest of the paper is organized as follows. In Section 2, we briefly introduce the related work. Section 3 presents the system model and problem statement. In Section 4, we describe and analyze the proposed DGA-EBCDS protocol in detail. The simulation results and corresponding discussions are given in Section 5. Finally, Section 6 concludes the paper and presents the further research work.

2. Related Work

The construction of connected dominating set in wireless networks has been extensively studied for many years. In order to measure the quality of a connected dominating set, the size of the CDS is usually chosen as the main concern factor. A smaller CDS can suffer less interference from its neighbors because each node in a WSN shares the communication channel with its neighbors. Moreover, a smaller CDS can perform well in reducing the number and the cost of control messages in routing and transmission. On the other hand, the smaller CDS can also make the management of the virtual backbone easier. Based on the above reasons, many researches focused on reducing the size of a CDS [18–31]. Dai and Wu [18] firstly introduced two greedy algorithms to compute the MCDS in a general graph, both of them were polynomial time algorithms. Some works [18–20, 22] focused on constructing \( k \)-connected \( m \)-dominating sets for fault tolerance, and most of these works used a UDG as their network model. Wu et al. [23] proposed a CDS construction algorithm using an energy model which was also adopted in other works [24, 25]. Cardei et al. [26] and Funke et al. [27] proposed another two distributed algorithms, which could achieve better performance. Authors in [28–30] presented a distributed algorithm for constructing CDS in UDGs, which consisted of two phases. Thai et al. [31] addressed the problem of constructing a CDS in a heterogeneous network and presented two approximation methods to obtain MCDS. Reference [21] considered constructing CDS in heterogeneous network, which formed an energy-efficient virtual network by using directional antennas. These algorithms effectively reduced the backbone size and improved the performance of the network.

Besides these algorithms, there were many other CDS-based algorithms aiming at the optimization goal of reducing the energy consumption of nodes in order to make the network live longer. CDS-BD-D [32] considered how to construct diameter-restricted minimum connected dominating set. The diameter referred to the maximum length of the shortest path between any two nodes in the CDS. Data collection could be finished in limited delays by constructing diameter-restricted MCDS. GOC-MCDS-D [33] considered the routing cost of nodes and achieved higher energy efficiency than CDS-BD-D. Wu_h [34] aimed at creating the minimum connected dominating set and ignored the diameter size of the CDS. Another classic protocol named mr-CDS [17] considered the residual energy of nodes and effectively improved the energy efficiency of Wu_h [34]. Firstly, the node with higher residual energy than its neighboring nodes had a higher priority to become the dominator node. The selected dominator broadcasted its dominant message to its neighbors. The dominatee nodes could infer the number of the dominators among their neighbors from these dominant messages. Then, each node that was not in the dominators set would broadcast its neighbors’ dominating set, which
enabled the 2-hop neighbors of the dominators to know the dominators. By comparing the set of the 2-hops neighbors, each dominatee node could judge if there were any paths for the 2-hop dominator nodes. If no path existed to connect the dominator nodes, the node itself would become a dominator node. Finally, all dominator nodes would form a connected dominating set. However, the process of a dominatee node changing to a dominator node could not consider the energy factor, and each dominatee node autonomously converted to a dominator node, which made too many dominatee nodes change to dominator nodes at the same time. This caused the nodes consuming too much energy to run the pruning algorithm so as to reduce the number of nodes in the CDS. The message complexity of mr-CDS is \( O(n \log n) \). Since mr-CDS is very easy and effective, we will compare this classic protocol with our protocol in this paper.

Moreover, there were some other algorithms aiming at the optimization goal of ensuring the reliability of the network. That is to say, the data could still be transmitted to the sink after the death of some nodes. LDA [35] focused on how to structure \( k \)-connected \( m \)-dominating set, and each dominator node constructed a local \( k \) vertex-connected subgraph according to the neighbor information. Then, each dominator node noticed the dominate nodes in this subgraph to join the CDS. FT-CDS-CA [36] studied the problem of constructing fault-tolerant CDS and proposed a constant factor polynomial time approximation algorithm to compute \((3, m)\)-CDS, which was similar to the method used in [22]. SSC [37] was the first self-stabilizing distributed \((k, r)\) algorithm, which used synchronous multiple paths to improve security, availability, and fault tolerance of the algorithm.

3. System Model and Problem Statement

3.1. Network Model. Assume that there are \( n \) sensor nodes in the network that are labeled as \( v_1, v_2, \ldots, v_n \), respectively. Denote the sink by \( v_0 \). All the nodes are randomly deployed in a \( M \times M \) field to continuously monitor the environment. Denote the transmission range of sensor nodes as \( r \). We assume that \( r \ll M \). The sink and all the sensor nodes form a connected network. Let \( V \) be the set of nodes. \(|V|\) represents the number of nodes in set \( V \), so \(|V| = n + 1 \). The network has the following characteristics:

1. The network is static; that is, all nodes and the sink are stationary after deployment.
2. Nodes may have different initial energy. The sink is assumed to have infinite power supply and powerful computation ability.
3. Nodes are not aware of their geographic information.
4. Data are highly correlated; the nodes in the virtual backbone (the nodes in the CDS) can aggregate the data into one packet of \( k \) bits for transmission no matter how much data are arriving. The data aggregation makes the node able to merge its own data with the received data from its neighbors, leading to significant energy and bandwidth savings.

In this paper, we consider a wireless sensor network where data is periodically reported from the sensor nodes to the sink. Data collection and transmission proceeds in rounds, the data from the sensor nodes are collected and aggregated with the packets of peer sensor nodes, and only one data is sent per round from the sensor network to the sink, so, each node has only one packet of information to communicate in each round. A round is defined as the process of collecting all the data from nodes to the sink, regardless of how much time it takes [38].

3.2. Energy Model. DGA-EBCDS uses the same energy consumption model which is widely adopted in previous works [39, 40]. According to this model, the energy dissipated to deliver a packet of \( k \) bits from the source to the destination is defined as

\[
E_s(k, r) = kE_{elec} + ke_{tera}r^2,
\]

where \( E_{elec} \) is the energy dissipated in operating the transmitter radio, \( r \) is the transmission range of the node and \( e_{tera}r^2 \) represents the energy dissipated by transmitter amplifier which varies with the distance \( r \) between the two nodes. The energy dissipated in operating the receiver radio is expressed as

\[
E_r(k) = kE_{elec}.
\]

3.3. Problem Statement. The minimum connected dominating set (MCDS) is an NP-hard problem for general graphs [41]. The MCDS can simplify network abstracted topology and reduce routing cost. However, when the nodes in the MCDS are without enough energy, these nodes will die earlier because of consuming energy too quickly.

Consider the graph shown in Figure 1(a). There are 7 nodes and one sink in the network, and each node starts with different amount of energy. A (1) represents node A having one unit of energy, and so forth.

In Figure 1(b), network generates CDS only by node degree. B has the largest node degree, so it becomes a dominator (to be blacked) firstly. And the neighboring nodes of B (A, C, D, E, and F) become the dominatees. The left node G has no other nodes to compare with, so it becomes the dominator naturally. In order to connect two dominators G and B, F becomes the connector (to be shaded). So, CDS contains three nodes (B, F, and G), accounting for 43% of the total number of nodes, and there is only one path from the CDS to the sink; this path transmits all the data flows of the network; thus, the energy consumption of this path is very large.

In Figure 1(c), network generates CDS by node energy. E and G have larger energy, so they become dominators firstly, and then, A and F become the connectors to connect G and the sink, and C and D become the connectors to connect E and the sink. Thus, CDS contains six nodes, accounting for 86% of the total number of nodes. CDS as the virtual backbone has to work long hours and should not go to the sleep mode, so this part of the nodes will run out of energy quickly and die earlier.
In Figure 1(d), network generates CDS by two factors: both the energy and the degree of a node. F and D both have larger degree and higher energy, so they become dominators firstly, and then, A becomes the connector to connect F and the sink, and C becomes the connector to connect D and the sink. Thus, CDS contains four nodes, accounting for 57% of the total number of nodes. Meanwhile, there are two paths from the CDS to the sink; more nodes outside the CDS can go to the sleep mode, which can not only reduce the size of the CDS, but also balance the energy consumption among nodes and extend the network lifetime.

So, constructing an energy-balanced CDS is a good method to solve the problems of MCDS, which takes both the energy and the degree of a node into consideration so as to effectively prolong the network lifetime in wireless sensor network.

4. The Design and Analysis of DGA-EBCDS

For the sake of brevity in describing our algorithm in the following, we give some definitions and notations here. Let \(N(v_i)\) be the neighbors set of a node \(i\). Dominators represent the nodes in the dominating set. Dominatees represent the nodes that are not in the dominating set. Connectors refer to the nodes that can connect two different dominators that are not adjacent to each other, and all the dominators and connectors can form a connected dominating set. Independent set (IS) is a subset of \(V\) such that no two vertices within the set are adjacent in \(V\). Maximal independent set (MIS) is an independent set such that adding any vertex not in the set breaks the independence property of the set; thus, the number of the IS nodes in the MIS is the largest. Let \(P(v_i)\) be the set of neighboring nodes of node \(i\) that cannot become any kind of dominators, dominates, or connectors. These definitions and notations will be used throughout this paper.

DGA-EBCDS is a distributed algorithm it depends only on the local information of nodes. The algorithm does not require the global information. DGA-EBCDS includes two stages: in the first stage, DGA-EBCDS selects the dominators; in the second stage, DGA-EBCDS selects the connectors, and finally, all the dominators and connectors form a CDS.

In order to construct an energy-balanced CDS, we use a weight \(W_i\) to measure the capability of a node, \(W_i = E_i \cdot \text{Degree}_i\), where \(E_i\) is the energy of node \(v_i\) and \(\text{Degree}_i\) is the degree of node \(v_i\). The higher the weight is, the more is its chance to become a dominator. By combining energy and degree together to measure the capability of a node, the node with a higher weight has higher priority to become a
In this section, we will describe the details of DGA-EBCDS backbone node. That is to say, the node with more energy and larger degree has more chance to become a CDS node. In this section, we will describe the details of DGA-EBCDS as follows.

4.1. Construction of Energy-Balanced CDS. Sensor nodes in the DGA-EBCDS can have three different colors: "white," "gray," and "black." Initially, all nodes are "white." During the execution of our algorithm, all nodes change their color to either "black" or "gray." Based on the weight comparison among neighbors, some suitable nodes are selected as dominators; nodes that are not in the dominating set remain as dominatees. All the black nodes form the dominating set (network backbone), whereas the gray nodes remain as dominatees, and they use neighboring dominators as next hops to transmit the data. Each node runs DGA-EBCDS distributedly and the algorithm includes the two following stages:

(1) Weight comparison among neighbors in the first stage, and some suitable nodes are selected as the dominators. During weight comparison, node $i$ is more suitable to be a dominator than its neighboring $j$ when node $i$ has a higher weight than $j$, or the weights of node $i$ and $j$ are the same, but the ID of node $i$ is smaller than that of node $j$. When a node $i$ wins the weight comparison over its neighbors, the node $i$ becomes the dominator and turns "black." When its neighboring nodes receive the black message from node $i$, they will become dominatees and turn "gray." Algorithm 1 gives detailed description of the selection of dominators by comparing the weights among nodes.

(2) In the second stage, the dominators select the suitable neighbors to become the connectors, and then the dominating nodes can be connected by the connectors. The selection is decided on the energy level of their neighbors as well as the connection condition between the neighbors and other dominators. When node $i$ becomes a connector, it will turn "black" too. Finally, all the dominators and connectors will form a connected dominating set (network backbone). We use two lists (List1 and List2) to save the path information. Algorithm 2 gives detailed description of the selection of connectors.

In Algorithm 2, at the beginning of the second stage, if the color of node $v_i$ is "Gray," $v_i$ will broadcast a message dominatee($v_j, v_k$) for each dominator $v_k$ that is one of the neighbors of $v_i$; this message will help $v_i$ to search for other dominators that $v_i$ can connect them through $v_k$.

When $v_i$ receives a dominatee($v_j, v_k$) message, $v_i$ checks the color of itself; if the color of node $v_i$ is "Gray" and if there is a dominator $v_m$ that is one of the neighbors of $v_i$ and $v_m < v_k$, $v_i$ will broadcast a message dominatee2($v_j, v_k, v_k$); the message will make $v_m$ know that it can connect $v_i$ through $v_j$ and $v_k$. If the color of node $v_i$ is "Black," $v_i$ will wait to receive the dominatee1 and the dominatee2 messages being sent from other neighbors; otherwise, if there is a record $[v_j, v_k]$ in List1 and $E_j < E_k$, delete $[v_j, v_k]$ from List1 and add $[v_j, v_k]$ to List2.

When $v_i$ receives a dominatee2($v_m, v_j, v_k$) message, if the color of node $v_j$ is "Black," and if List2 is empty or $[v_m, v_j, v_k]$ cannot be found in List2, add the record $[v_m, v_j, v_k]$ to List2; $v_i$ waits to receive the dominatee1 and the dominatee2 messages being sent from other neighbors; otherwise, if there is a record $[v_m, v_j, v_k]$ in List2 and $min(E_m, E_j) < min(E_m, E_k)$, delete $[v_m, v_j, v_k]$ from List2 and then add the record $[v_m, v_j, v_k]$ to List2.

When the timer expires, if $v_i$ can connect with $v_k$ through $v_j$, add $v_i$ to a set $A$ and delete the record $[v_m, v_j, v_k]$ from List2; if $v_i$ $\in$ $A$ and the color of node $v_i$ is "Gray," $v_i$ will change itself from a dominatee to a connector, and then the color of $v_i$ turns to "Black."

Each node runs DGA-EBCDS distributedly; when a node turns to "Black," it will never change its color again. At the end of the second stage, the set of "Black" nodes (dominators and connectors) forms a CDS.

---

**Algorithm 1:** The first stage for DGA-EBCDS.

At the beginning of the first stage:

$P(v_i) = N(v_i);
\text{if}(W_j > max \{W_j | v_j \in P(v_j)\}) \&\&(W_j = max \{W_j | v_j \in P(v_j)\})
\{ v_j \text{ color } = \"Black\};
\text{if}(v_j \text{ color } = \"Black\})
\text{v_j broadcasts a message dominate(v_j)};
\text{When v_j receives a dominate (v_i) message:}
\{ v_j \text{ color } = \"White\}
\{ delete v_j from P(v_i);\}
\text{if}(W_j > max \{W_j | v_j \in P(v_j)\}) \&\&(W_j = max \{W_j | v_j \in P(v_j)\})
\{ v_j \text{ color } = \"Black\};
\text{v_j broadcasts a message dominate(v_j)};\}

---

**Algorithm 2:** The second stage for DGA-EBCDS.

In Algorithm 2, at the beginning of the second stage, if the color of node $v_j$ is "Gray," $v_j$ will broadcast a message dominatee($v_j, v_k$) for each dominator $v_k$ that is one of the neighbors of $v_j$; this message will help $v_j$ to search for other dominators that $v_j$ can connect them through $v_k$.

When $v_j$ receives a dominatee($v_j, v_k$) message, $v_j$ checks the color of itself; if the color of node $v_j$ is "Gray" and if there is a dominator $v_m$ that is one of the neighbors of $v_j$ and $v_m < v_k$, $v_j$ will broadcast a message dominatee2($v_j, v_k, v_k$); the message will make $v_m$ know that it can connect $v_j$ through $v_j$ and $v_k$. If the color of node $v_j$ is "Black," $v_j$ will wait to receive the dominatee1 and the dominatee2 messages being sent from other neighbors; otherwise, if there is a record $[v_j, v_k]$ in List1 and $E_j < E_k$, delete $[v_j, v_k]$ from List1 and then add $[v_j, v_k]$ to List2.

When $v_j$ receives a dominatee2($v_m, v_j, v_k$) message, if the color of node $v_j$ is "Black," and if List2 is empty or $[v_m, v_j, v_k]$ cannot be found in List2, add the record $[v_m, v_j, v_k]$ to List2; $v_j$ waits to receive the dominatee1 and the dominatee2 messages being sent from other neighbors; otherwise, if there is a record $[v_m, v_j, v_k]$ in List2 and $min(E_m, E_j) < min(E_m, E_k)$, delete $[v_m, v_j, v_k]$ from List2 and then add the record $[v_m, v_j, v_k]$ to List2.

When the timer expires, if $v_j$ can connect with $v_k$ through $v_j$, add $v_j$ to a set $A$ and delete the record $[v_m, v_j, v_k]$ from List2; if $v_j$ $\in$ $A$ and the color of node $v_j$ is "Gray," $v_j$ will change itself from a dominatee to a connector, and then the color of $v_j$ turns to "Black."

Each node runs DGA-EBCDS distributedly; when a node turns to "Black," it will never change its color again. At the end of the second stage, the set of "Black" nodes (dominators and connectors) forms a CDS.
4.2. Analysis of the Distributed Algorithm

**Theorem 1.** The set of the dominators is a maximal independent set (MIS) when the first stage for DGA-EBCDS terminates.

*Proof.* From the detailed description of the algorithm shown in Algorithm 1, all the nodes will make a weight comparison among neighbors in the first stage of DGA-EBCDS. During weight comparison, \( v_i \) will be selected as a dominator when \( v_i \) has a higher weight than its neighboring \( v_j \), or the weights of \( v_i \) and \( v_j \) are the same, but the ID of \( v_i \) is smaller than that of the ID of \( v_j \). When \( v_i \) wins the weight comparison over its neighbors, it will become a dominator.

When \( v_i \) becomes a dominator, all the neighbors of \( v_i \) will become dominates; thus, each two dominators cannot be connected directly. From the above analysis, the set of all the dominators is an independent set (IS). For any node \( v_i \in V \), according to DGA-EBCDS, there are only two possible cases for \( v_i \): if \( v_i \) is not a dominator, it must be a dominatee. For each dominatee, there is certainly at least one dominator in its neighbors. Therefore, any dominatee turning to a dominator will break the independence property of the current set (produce two dominators that can be connected directly). Thus, the set of the dominators is a maximal independent set (MIS) when the first stage for DGA-EBCDS terminates.

**Theorem 2.** The message complexity of DGA-EBCDS is \( O(n \log n) \).

*Proof.* In order to guarantee that the network is connected, the transmission radius of \( r \) should satisfy the following formula [42]:

\[
r \geq \frac{\sqrt{2}}{2} M \sqrt{\frac{1}{n} \log \left( \frac{n}{\delta} \right)},
\]
where $\delta$ represents the probability that the network is not connected. Certainly, the number of the neighboring nodes must satisfy

$$N_{\text{neighborhood}} \approx \frac{n r^2}{M^2} \geq \frac{\pi}{2} \log \left( \frac{n}{\delta} \right) = O \left( \log n \right). \quad (4)$$

From the detailed description of the algorithms shown in Algorithms 1 and 2, for each node $v_i$, if $v_i$ is a dominator, it will send two messages: one message for declaring itself being selected as the dominator and the other message for broadcasting the set of the connectors.

If $v_i$ is a dominatee, it will send at most $O(\log n)$ messages: one message for broadcasting dominatee($v_i$), $O(\log n)$ messages for broadcasting dominatee($v_j$, $v_k$), and $O(\log n)$ messages for broadcasting dominatee($v_j$, $v_p$, $v_k$), where $O(\log n)$ is decided by the number of the neighboring nodes of $v_i$.

If $v_i$ turns out to be a connector, it will broadcast connector ($A$, $a$) message only once. Thus, a node will send at most $O(\log n)$ messages in DGA-EBCDS. There are $n$ nodes in the network; we can deduce that the total message complexity of the network is $O(n \log n)$, which is the same as the message complexity of the classic DS-based algorithm mr-CDS.

5. Performance Evaluation

We conduct extensive simulations to evaluate the performance of DGA-EBCDS. The experiments are assumed to be performed in a square field of $M \times M$, in which nodes are randomly dispersed. The parameters are as follows.

Each node in the field is assigned a randomly-generated initial energy level between 1 joule($J$) and 1.5$J$, $E_{\text{elec}} = 50 \text{ pJ}/\text{bit}$, $\epsilon_{\text{se}} = 13 \text{ pl}/\text{bit}/\text{m}^2$, and each node will generate only one packet of 1000 bits per round. We compare DGA-EBCDS with mr-CDS and use the same assumption of mr-CDS. Sink is located at the center of the field; its coordinate is $(0.5M, 0.5M)$. All experiments will be performed 20 times, and their average values are taken as the final results. Our target is to balance the energy among nodes and delay the death of the first node. Network density is defined as the number of nodes lying in each unit of the field, which can be described as $n/M^2$. Average node degree represents the number of nodes in its transmission range, which can be described as $n r^2/M^2$; we will use them to set our experiment scenes.

5.1. Backbone Size. We simulate a network of 480 static nodes placed randomly in a $1000 \text{ m} \times 1000 \text{ m}$ area to observe the distribution of the CDS. The transmission range of nodes is set to be 100 m. The simulation results are shown in Figure 2.

In Figure 2, the sink is marked with a pentagram, the nodes in the CDS (including dominators and connectors) are marked with circles adding a plus sign, and dominatees are marked with circles. We can observe that the CDS produced by DGA-EBCDS and mr-CDS can both cover the whole network, which can effectively guarantee the network coverage.

In order to measure the validity of these two algorithms, we have used two different environments to randomly deploy the sensor nodes. Figures 2(a) and 2(b) are the first deployment environment, and Figures 2(c) and 2(d) are the second deployment environment.

We can see from Figures 2(a) and 2(b), in the first experiment, on average, 142 nodes constitute the mr-CDS backbone, whereas in case of DGA-EBCDS, there are only 79 nodes generated in the backbone; the backbone size of DGA-EBCDS is reduced by 44.4% on average.

Meanwhile, we can see from Figures 2(c) and 2(d), in the second experiment, on average, 146 nodes constitute the mr-CDS backbone, whereas in case of DGA-EBCDS, there are only 86 nodes generated in the backbone; the backbone size of DGA-EBCDS is reduced by 41.1% on average.

As backbone nodes must remain active during the network operating period, with fewer nodes in the backbone, DGA-EBCDS consumes less energy to keep the network in an operational state, and a smaller backbone in DGA-EBCDS can make more nodes (dominatees) go to a periodic sleep mode and effectively save the energy of the network. Moreover, no matter how the randomized sensor placements change, DGA-EBCDS always achieves better performance than mr-CDS.

5.2. Energy Consumption in the Formation of CDS. In order to compare the energy efficiency of DGA-EBCDS and mr-CDS, we have examined the average energy consumption per node in the formation phase of CDS. The transmission range of nodes is set to 100 m, and the message packets size of nodes is 32 bits. We have investigated both algorithms under varying conditions like average node degree and the scale of the network. Two experimental scenes are used:

(1) Scene 1: the area is fixed to $1000 \text{ m} \times 1000 \text{ m}$. Assume that there are 320, 480, 640, 800, and 960 nodes randomly distributed in the field, respectively. The average node degree is 10, 15, 20, 25, and 30 under this scene, respectively.

(2) Scene 2: the average node degree is fixed to 15. The area is $800 \text{ m} \times 800 \text{ m}$, $1000 \text{ m} \times 1000 \text{ m}$, $1200 \text{ m} \times 1200 \text{ m}$, and $1500 \text{ m} \times 1500 \text{ m}$, respectively. There are 306, 480, 688, and 1075 nodes in this scene, respectively. The energy consumption in the formation phase of CDS is shown in Figure 3.

We can see from Figure 3(a), in the formation of CDS, the energy consumption of nodes increases when the average node degree grows. The reason is that nodes need to exchange messages with their neighbors. As the average node degree increases, the node needs to exchange its messages with more neighbors, which exhausts too much energy. However, DGA-EBCDS consumes less energy than mr-CDS; the average energy consumption has been reduced by 33.3%.

In Figure 3(b), when average node degree is fixed, the energy consumption for DGA-EBCDS and mr-CDS will be similar. Because the numbers of average neighbors of nodes will remain unchanged when average node degree is fixed,
the energy consumption will be unchanged. However, DGA-EBCDS consumes less energy than mr-CDS in different scales of network. Based on the observation, we can get the conclusion that DGA-EBCDS considers the energy balancing among nodes, as a result, DGA-EBCDS wastes less energy and acquires better energy efficiency than mr-CDS.

5.3. Network Lifetime. We set the same scenes of section B to evaluate the network lifetime. The energy of node will be reducing in data collection and the energy is changing per round. Node broadcasts its energy message at the beginning of each round. The message of energy can be delivered by some small packets; the energy consumption of this part can be neglected in the network.

In wireless sensor networks, the energy of node is very limited, if the network lifetime (FND) expires, remaining energy of the alive nodes cannot come to any use. So our algorithm tries to maximize the use of overall network energy before a network expires and extend network lifetime by delaying the death of the first node. In the process of data collection, the node cannot acquire the path information, so it does not know from which path it can transmit its data to sink with less energy consumption. There are many algorithms using the nodes in the CDS to construct a data collection tree and set the sink as the root of the tree. We no longer specially discuss this method here, and we use the distance vector (DV) routing algorithm [43] to construct a data collection tree after forming a CDS-based backbone. The collection tree is also used by mr-CDS. The sink acts as the root node and initiates tree formation, and only the dominators take part in the construction of the tree. Furthermore, we can see from Figure 2 that the number of dominators of DGA-EBCDS is quite lower when compared to the network size, so running
DV algorithm only over the dominators can reduce the cost of the network.

In the data collection process, each dominatee will transmit its data to the neighbor that has the highest energy in the data collection tree, and then it goes to the sleep mode to save its energy. The nodes in the data collection tree will aggregate the received data into one packet of 1000 bits and then transmit the packet to their father. A round of data collection terminates when the sink receives all the data from its children. Network lifetime achieved by these two algorithms is shown in Figure 4.

In Figure 4(a), the network lifetime is decreasing as the average node degree increases. DGA-EBCDS achieves longer network lifetime than mr-CDS no matter how the average node degree changes.

In Figure 4(b), when the average node degree is fixed, the network lifetime in large-scale network is shorter than that in small-scale network. As the transmission range is the same, the data need to be transmitted more times to get to the sink in large-scale network than in small-scale network, so the network lifetime in large-scale network will be reduced. However, DGA-EBCDS achieves longer network lifetime than mr-CDS. On average, the network lifetime is prolonged by 57.8%.

5.4. Delay. We set the same scenes of Section 5.2 to evaluate the data dissemination delays of these two algorithms. Since data dissemination delay are mainly affected by the maximum number of hops from the node to the sink, that is, the larger the number of hops, the longer the delay, we use the maximum number of hops between sink and the node to measure the data dissemination delay. The comparison results of data dissemination delay are shown in Figure 5.

In Figure 5, we can see that the data dissemination delays are mostly influenced by the size of the network; however, the average node degree produces very little influence on data dissemination delays.

In Figure 5(a), the data dissemination delays of these two algorithms remain unchanged as the average node degree increases. The average delay of DGA-EBCDS is 9.63 hops, and the average delay of mr-CDS is 8.36 hops. DGA-EBCDS incurs a larger delay (just only one hop on average) than that of mr-CDS.

In Figure 5(b), when the average node degree is fixed, the data dissemination delays of these two algorithms in large-scale network are larger than that in small-scale network. As the transmission range is the same, the data need to be relayed by more nodes to get to the sink in large-scale network than in small-scale network, so the data dissemination delays in large-scale network will be enlarged. The average delay of DGA-EBCDS is 10.49 hops, and the average delay of mr-CDS is 9.29 hops; mr-CDS shortens the delay by only one hop on average. Moreover, from Figure 2, we can see that the backbone size of DGA-EBCDS is smaller than that of mr-CDS. In some cases, there are fewer paths between the nodes and the sink, so some CDS nodes of DGA-EBCDS need to be transmitted more times to get to the sink; thus, DGA-EBCDS incurs larger delays than that of mr-CDS.

However, DGA-EBCDS considers the energy balancing among nodes. The algorithm can significantly delay the death of the first node, and the dominators can take more rounds for data collection. Based on the observation, we can draw the conclusion that DGA-EBCDS gains better performance than mr-CDS.

6. Conclusion and Future Work

Selecting proper nodes to construct the CDS in order to maximize the network lifetime is an important issue when
designing connected dominating set protocols and algorithms for data collection in wireless sensor networks. Most existing algorithms cannot consider the energy balancing among nodes, which limits their applications in large scale WSNs. The high contribution of this paper is the construction of an energy-balanced connected dominating set (DGA-EBCDS) for data collection in wireless sensor network. DGA-EBCDS can effectively increase the number of rounds before the first node failure by reducing the energy consumption per round and only choosing the nodes with a relatively higher weight to construct the CDS. Theoretical analyses show that the total message complexity of the algorithm is $O(n \log n)$, and the simulation results show that DGA-EBCDS can reduce energy consumption in the formation of CDS as well as effectively prolong the lifetime of the network.

In the next step, how to switch the role of dominators when designing the CDS backbone is a new research direction for us. If we always use the same set of dominators, those nodes must relay network traffic all the time, and they will easily run out of energy when compared with the...
dominatee nodes. We can consider that once the energy of a dominator goes below a threshold, a new node can take up the responsibility of this dominator. Therefore, researches that aim at delaying the death of the first node and switching the role of dominators can be useful in the future work.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant nos. 61202494, 61103245), the Scientific Research Fund of Hunan Provincial Education Department (Grant no. 11CI099), the Natural Science Foundation of Guangxi (Grant no. 2012GXNSFBA053163), Hunan Provincial Science and Technology Department projects (2010FJ4052), and the Freedom Explore Program of Central South University (2012QNZNTO68).

References

[1] P. J. Chuang, Y. C. Yu, and C. S. Lin, “Reliable and congestion-controlling transport in wireless sensor networks,” Journal of the Chinese Institute of Engineers, vol. 36, no. 1, pp. 2–16, 2013.

[2] A. F. Liu, Z. M. Zheng, C. Zhang, Z. Chen, and X. Shen, “Secure and energy-efficient disjoint multi-path routing for WSNs,” IEEE Transactions on Vehicular Technology, vol. 61, no. 7, pp. 3255–3265, 2012.

[3] N. Kimura, V. Jolly, and S. Latifi, “Energy restrained data dissemination in wireless sensor networks,” International Journal of Distributed Sensor Networks, vol. 2, no. 3, pp. 251–265, 2006.

[4] A. F. Liu and Z. H. Liu, “An elaborate chronological and spatial analysis of energy hole for wireless sensor networks,” Computer Standards and Interfaces, vol. 35, no. 1, pp. 132–149, 2013.

[5] Y. Xu, W. C. Lee, J. Xu, and G. Mitchell, “Energy-aware and time-critical geo-routing in wireless sensor networks,” International Journal of Distributed Sensor Networks, vol. 4, no. 4, pp. 315–346, 2008.

[6] K. Ota, M. Dong, Z. Cheng, J. Wang, X. Li, and X. Shen, “ORACLE: mobility control in wireless sensor and actor networks,” Elsevier Computer Communications, vol. 35, no. 9, pp. 1029–1037, 2012.

[7] H. Zhu, X. Lin, R. Lu, Y. Fan, and X. Shen, “SMART: a secure multilayer credit-based incentive scheme for delay-tolerant networks,” IEEE Transactions on Vehicular Technology, vol. 58, no. 8, pp. 4628–4639, 2009.

[8] H. Zhu, S. Du, Z. Gao, M. Dong, and Z. Cao, “A probabilistic misbehavior detection scheme towards efficient trust establishment in delay tolerant networks,” IEEE Transactions on Parallel and Distributed Systems, 2013.

[9] X. Y. Kui, J. X. Wang, and S. G. Zhang, “Energy-balanced clustering protocol for data gathering in wireless sensor networks with unbalanced traffic load,” Journal of Central South University, vol. 19, no. 11, pp. 3180–3187, 2012.

[10] C. Zheng, S. X. Sun, and T. Y. Huang, “Constructing distributed connected dominating sets in wireless Ad Hoc and sensor networks,” Ruan Jian Xue Bao/Journal of Software, vol. 22, no. 5, pp. 1053–1066, 2011.

[11] S. Gao and H. K. Zhang, “Optimal path selection for mobile sink in delay-guaranteed sensor networks,” Tien Tzu Hsueh Pao/Acta Electronica Sinica, vol. 39, no. 4, pp. 742–747, 2011.

[12] S. Ji and Z. Cai, “Distributed data collection in large-scale asynchronous wireless sensor networks under the generalized physical interference model,” IEEE/ACM Transactions on Networking, 2012.

[13] S. Ji and Z. Cai, “Distributed data collection and its capacity in asynchronous wireless sensor networks,” in Proceedings of the 31st Annual IEEE International Conference on Computer Communications (INFOCOM ’12), pp. 2113–2121, Orlando, Fl, USA, March 2012.

[14] A. F. Liu, P. H. Zhang, and Z. G. Chen, “Theoretical analysis of the lifetime and energy hole in cluster based Wireless Sensor Networks,” Journal of Parallel and Distributed Computing, vol. 71, no. 10, pp. 1327–1355, 2011.

[15] A. F. Liu, X. Jin, G. H. Cui, and Z. G. Chen, “Deployment guidelines for achieving maximum lifetime and avoiding energy holes in sensor network,” Information Sciences, vol. 230, pp. 197–226, 2013.

[16] J. He, S. Ji, Y. Pan, and Z. Cai, “Approximation algorithms for load-balanced virtual backbone construction in wireless sensor networks,” Theoretical Computer Science, 2012.

[17] S. Hussain, M. I. Shahique, and L. T. Yang, “Constructing a CDS-based network backbone for energy efficiency in industrial wireless sensor network,” in Proceedings of the 12th IEEE International Conference on High Performance Computing and Communications (HPCC ’10), pp. 322–328, Melbourne, Australia, September 2010.

[18] F. Dai and J. Wu, “On constructing k-connected k-dominating in wireless networks,” in Proceedings of the 9th IEEE International Parallel and Distributed Processing Symposium (IPDPS ’05), Denver, Colo, USA, April 2005.

[19] F. Wang, M. T. Thai, and D. Z. Du, “On the construction of 2-connected virtual backbone in wireless networks,” IEEE Transactions on Wireless Communications, vol. 8, no. 3, pp. 1230–1237, 2009.

[20] Y. Wu, F. Wang, M. T. Thai, and Y. Li, “Constructing k-connected M-dominating sets in wireless sensor networks,” in Proceedings of the Military Communications Conference (MILCOM ’07), pp. 1–7, October 2007.

[21] S. Yang, J. Wu, and F. Dai, “Efficient backbone construction methods in MANETs using directional antennas,” in Proceedings of the 27th International Conference on Distributed Computing Systems (ICDCS ’07), pp. 45–50, June 2007.

[22] M. T. Thai, N. Zhang, R. Tiwari, and X. Xu, “On approximation algorithms of k-connected m-dominating sets in disk graphs,” Theoretical Computer Science, vol. 385, no. 1–3, pp. 49–59, 2007.

[23] J. Wu, M. Gao, and I. Stojmenovic, “On calculating power-aware connected dominating sets for efficient routing in Ad Hoc wireless networks,” in Proceedings of the IEEE International Conference on Parallel Processing (ICPP ‘01), pp. 346–354, September 2001.

[24] J. Wu, B. Wu, and I. Stojmenovic, “Power-aware broadcasting and activity scheduling in ad hoc wireless networks using connected dominating sets,” Wireless Communications and Mobile Computing, vol. 3, no. 4, pp. 425–438, 2003.

[25] Y. Zeng, X. Jia, and Y. He, “Energy efficient distributed connected dominating sets construction in wireless sensor networks,” in Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC ’06), pp. 797–802, July 2006.

[26] M. Cardei, X. Cheng, X. Cheng, and D. Z. Du, “Connected domination in multihop Ad Hoc wireless networks,” in Proceedings of the 6th International Conference on Computer Science and Informatics (IJCSI ’02), pp. 251–255, March 2002.
[27] S. Funke, A. Kesselman, U. Meyer, and M. Segal, “A simple improved distributed algorithm for minimum CDS in unit disk graphs,” ACM Transactions on Sensor Networks, vol. 2, no. 3, pp. 444–453, 2006.

[28] P. J. Wan, K. M. Alzoubi, and O. Frieder, “Distributed construction of connected dominating set in wireless ad hoc networks,” in Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’02), pp. 1597–1604, June 2002.

[29] K. M. Alzoubi, P. J. Wan, and O. Frieder, “New distributed algorithm for connected dominating set in wireless Ad Hoc networks,” in Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS ’02), pp. 3849–3855, 2002.

[30] K. M. Alzoubi, W. Peng-Jun, and O. Frieder, “Distributed heuristics for connected dominating sets in wireless ad hoc networks,” Journal of Communications and Networks, vol. 4, no. 1, pp. 22–29, 2002.

[31] M. T. Thai, F. Wang, D. Liu, S. Zhu, and D. Z. Du, “Connected dominating sets in wireless networks with different transmission ranges,” IEEE Transactions on Mobile Computing, vol. 6, no. 7, pp. 721–730, 2007.

[32] D. Kim, Y. Wu, Y. Li, F. Zou, and D. Z. Du, “Constructing minimum connected dominating sets with bounded diameters in wireless networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 20, no. 2, pp. 147–157, 2009.

[33] H. Du, Q. Ye, W. Wu et al., “Constant approximation for virtual backbone construction with Guaranteed Routing Cost in wireless sensor networks,” in Proceedings of the 30th IEEE International Conference on Computer Communications (INFOCOM ’11), pp. 1737–1744, Shanghai, China, April 2011.

[34] K. A. M. Almahorg, S. Naik, and X. Shen, “Efficient localized protocols to compute connected dominating sets for ad hoc networks,” in Proceedings of the 53rd IEEE Global Communications Conference (GLOBECOM ’10), pp. 1–5, Miami, Fla, USA, December 2010.

[35] Y. Wu and Y. Li, “Construction algorithms for k-connected m-dominating sets in wireless sensor networks,” in Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc ’08), pp. 83–90, HongKong, China, May 2008.

[36] D. Kim, W. Wang, X. Li, Z. Zhang, and W. Wu, “A new constant factor approximation for computing 3-connected m-dominating sets in homogeneous wireless networks,” in Proceedings of the 29th IEEE Conference on Computer Communications (INFOCOM ’10), pp. 1–9, San Diego, Calif, USA, March 2010.

[37] A. Larsson and P. Tsigas, “A self-stabilizing, (k, r)-clustering algorithm with multiple paths for wireless Ad-hoc networks,” in Proceedings of the 31st International Conference on Distributed Computing Systems (ICDCS ’11), pp. 353–362, Minneapolis, Minn, USA, June 2011.

[38] H. Ö. Tan and I. Körpeoğlu, “Power efficient data gathering and aggregation in wireless sensor networks,” SIGMOD Record, vol. 32, no. 4, pp. 66–71, 2003.

[39] X. Y. Kui, S. G. Zhang, J. X. Wang, and J. N. Cao, “An energy-balanced clustering protocol based on dominating set for data gathering in wireless sensor networks,” in Proceedings of the 2012 IEEE International Conference on Communications (ICC ’12), pp. 193–197, Ottawa, Canada, June, 2012.

[40] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” IEEE Transactions on Wireless Communications, vol. 1, no. 4, pp. 660–670, 2002.

[41] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, Calif, USA, 1978.

[42] V. Mhatre and C. Rosenberg, “Design guidelines for wireless sensor networks: communication, clustering and aggregation,” Ad Hoc Networks, vol. 2, no. 1, pp. 45–63, 2004.

[43] R. Malhotra, IP Routing, O’Reilly, 1st edition, 2002.
