

Research Article

Distributed Energy-Efficient Approaches for Connected Dominating Set Construction in Wireless Sensor Networks

Najla Al-Nabhan, Mznah Al-Rodhaan, and Abdullah Al-Dhelaan

Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

Correspondence should be addressed to Najla Al-Nabhan; nalnabhan@ksu.edu.sa

Received 29 December 2013; Revised 12 May 2014; Accepted 13 May 2014; Published 10 June 2014

Academic Editor: Li-Hsing Yen

Copyright © 2014 Najla Al-Nabhan 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.

Energy efficiency is one of the major issues in wireless sensor networks (WSNs) and their applications. Distributed techniques with low message and time complexities are expected in WSNs. Connected dominating sets (CDSs) have been widely used for virtual backbone construction in WSNs to control topology, facilitate routing, and extend network lifetime. Most of the existing CDS approaches suffer from a very poor approximation ratio, high time, and message complexities. This paper proposes two novel approaches for CDS distributed construction in WSNs. The proposed approaches are intended to construct a small CDS as well as allowing energy-efficient CDS construction and maintenance in WSNs. Simulation shows that our distributed approaches have an approximation factor of 7.5 to the optimal CDS. This approximation outperforms the existing distributed CDS construction algorithms.

1. Introduction

A wireless sensor network (WSN) is a wireless network that consists of thousands of very small stations called *sensor* nodes. The main function of sensor nodes is to monitor, record, and notify a specific condition at various locations to other stations and end users. WSNs are increasingly attractive as a means to provide more advanced, intelligent, and context-aware systems with implicit user interaction. They have a wide range of application areas such as geophysical monitoring, precision agriculture, habitat monitoring, transportation, health, military systems, and business processes [1, 2].

WSNs pose their unique challenges due to the lack of a central entity for organization, the sensors' limitation, and mobility of the participants, as well as the limited range of wireless communications. Furthermore, in most of their applications, sensors are flexibly and quickly deployed with minimal effort, eliminating the need for physical backbone infrastructure [1, 2]. As each sensor node is tightly power constrained and one-off, the lifetime of a WSN is limited. In order to prolong the network lifetime, energy-efficient protocols should be designed for the characteristic of WSNs

[2, 3]. An important research problem in wireless-sensing networking is to find a small set of nodes that can collaborate to form a self-organizing network to substitute for the absence of infrastructure and central control. This virtual backbone plays a significant role in enhancing the network efficiency, extending its lifetime, and supporting routing processes as well as all other network tasks and applications.

Connected dominating sets (CDSs) have been widely used for constructing virtual backbone in wireless networks. Using CDS, routing will be restricted to the reduced graph formed by the CDS. Every node in a CDS, called dominator, is considered as a gateway, and only a gateway needs to keep routing information. Other nodes, called dominatees, need not keep any routing information. Moreover, any dominatee can switch to sleep mode for energy saving without causing network partition. In such a scenario, valuable resources—for example, transmission energy, nodes' memory and computation time, and bandwidth—are saved [1, 2].

The minimum connected dominating set (MCDS) problem is to find a connected dominating set with the smallest possible cardinality among all connected dominating sets on a graph G . The MCDS problem is defined as follows. For a given connected graph (network) $G = (V, E)$, where V

is the set of vertices (sensors) and E is the set of edges that provides the available communications, a dominating set (DS) is a subset V' of V , where, for each vertex u of V , u is either in V' or has at least one neighbor vertex in V' . The minimum dominating set (MDS) problem is to find a dominating set with smallest cardinality. The decision version of the MDS is a classical NP-complete problem. The dominating set D is a connected dominating set if the subgraph induced by vertices in D is connected. Maximal independent set (MIS) is a subset M of V that satisfies the following conditions: (i) nodes in the MIS are pairwise nonadjacent, and (ii) no more nodes can be added to maintain the nonadjacency property of this set. Each node that is not in the MIS is adjacent to at least one node in the MIS. If we connect all nodes in the MIS through some nodes not in the MIS, a CDS is then constructed [1].

In the context of CDS construction, the approximation ratio of an algorithm A is defined as the largest (worst) ratio between the size of the obtained CDS using algorithm A and the optimal result that can be obtained by MCDS (opt). Most of the existing algorithms for CDS construction suffer from a very poor approximation ratio compared to the MCDS and from high time and message complexities. Recently, a new centralized algorithm for constructing a CDS was proposed in [4, 5], with a constant approximation ratio of 5. This approximation factor is the smallest in the literature. While most of the existing CDS construction algorithms are based on the conventional MIS that guarantees the distance between any complementary subsets is *exactly two hops* or at *most three hops*, the proposed approach in [4, 5] is based on constructing a *special MIS* which guarantees that the distance between any of its complementary subsets is *exactly three hops*. Unfortunately, the algorithm is centralized and based on constructing a spanning tree, which makes it very costly in terms of communication overhead to maintain the CDS in the case of mobility and topology changes. Moreover, most of the proposed approach does not pay attention to the residual energy in ranking nodes for CDS construction. Therefore, a distributed approach with energy efficiency is needed.

Centralized approaches usually achieve better performance than distributed approaches, but they utilize the global information of the network, so they are energy consuming and hard to be realized in the practical applications. On the contrary, distributed approaches can be realized by sensors with low complexity and have become a hot topic. Additionally, WSNs are usually exposed to challenging and dynamic environments. Therefore, it is possible for connectivity loss of individual nodes to occur. In these situations, the conventional centralized algorithms that need to operate with global knowledge of the whole network will potentially experience a serious protocol failure as a result of transmission errors or a failure of a critical node [3].

In this paper, we propose a distributed energy-efficient algorithm for CDS construction in wireless sensor networks. The algorithm extends the state-of-the-art centralized algorithm, proposed in [4, 5], in that it implements the exact-three-hop property in the distributed environment. Moreover, the algorithm employs a *new raking function* that is carefully implemented to ensure that the constructed CDS is really *connected*. The introduced ranking function is intended

to provide *energy awareness*. It attempts to prolong the lifetime of the constructed CDS by allowing nodes with higher residual energy to be more likely part of the constructed backbone. In addition, the paper proposes a nonspanning tree based CDS construction approach (approach II).

The rest of this paper is organized as follows. Next section discusses the related work. Section 3 presents the detailed design of the proposed approaches. Section 4 provides the performance discussion and simulation results. In Section 5, we conclude our findings.

2. Related Work

Various techniques were developed in the literature for CDS construction with different performance ratios and design perspectives. CDS construction methods are evaluated according to many factors. Some of these factors characterize the constructed CDS, including size, diameter, and approximation factor. Other criteria, including algorithm complexity, communication overhead, and information range, are related to the construction algorithm [2].

Generally speaking CDS construction algorithms are usually classified into centralized [4–9] and distributed algorithms [10–18]. For any CDS algorithm, the size of the constructed CDS is usually considered the most important performance factor. For distributed CDS construction algorithms, it is difficult to achieve a small approximation factor compared to the centralized methods. Distributed algorithms pay more attention for other metrics such as message complexity and information range. The *message complexity* metric refers to the number of exchanged messages between nodes during the CDS construction in the worst-case scenario. *Information range* shows the amount of neighborhood information that should be collected by a node to perform the CDS construction task. It is usually measured in number of hops. Information range has an influence on algorithm and message complexities. A good distributed algorithm is the one that has low message complexity and requires the least neighboring information to construct the CDS [2]. Table 1 shows the approximation factors of the state-of-the-art CDS construction algorithms.

In most of the CDS construction algorithms, a coloring mechanism is used for the purpose of illustration. All nodes are initially white, then dominators are colored black and dominatees are colored gray. A centralized algorithm, called S-MIS, was presented in [6]. The algorithm is greedy. It constructs a small CDS based on the classical MIS model. MIS-based algorithms are two-phase based realization algorithms. They construct the MIS at first and then find some optimal nodes to connect the MIS nodes together in order to obtain the final CDS. In this algorithm, MIS is first constructed and then a Steiner tree algorithm is employed to interconnect MIS nodes into CDS. The approximation factor achieved by S-MIS algorithm is $(5.8 + \ln 4)$ from the optimal solution. A novel centralized algorithm was proposed in [4, 5] for CDS construction in WSNs. The proposed algorithm is a four-phase-based realization algorithm with two different types of dominators and two different types of connectors.

TABLE I: The state-of-the-art CDS construction algorithms and their approximation factor.

CDS construction approach	Algorithm category	Construction method	Performance factor
(Al-Nabhan et al. 2012) [4]	Centralized	Spanning tree based	5
(Li et al. 2005) [6]	Centralized	Spanning tree based	$5.8 + \ln 4$
(Liu et al. 2010) [7]	Centralized	Spanning tree based	9.67
(Li et al. 2007) [8]	Centralized	Spanning tree based	11.4
(Zou et al. 2008) [9]	Centralized	Spanning tree based	15.3
Approach I (this paper)	Distributed	Spanning tree based	7.5
Approach II (this paper)	Distributed	Nonspanning tree based	7.5
(Alzoubi et al. 2002) [11]	Distributed	Spanning tree based	8
(Cardei et al. 2002) [13]	Distributed	Spanning tree based	8
(Butenko et al. 2003) [14]	Distributed	Spanning tree based	8
(Kim et al. 2010) [18]	Distributed	Nonspanning tree based	14.93
(Islam 2010) [17]	Distributed	Nonspanning tree based	50
(Cheng et al. 2006) [16]	Distributed	Nonspanning tree based	147
(Li et al. 2004) [15]	Distributed	Nonspanning tree based	172
(Alzoubi et al. 2002) [12]	Distributed	Nonspanning tree based	192

The algorithm introduces a new model for CDS construction based on the exact-three-hops property. In this model, a special independent set, called S_1 , is first constructed. This set satisfies the following condition. *The hop distance between any two complementary subsets of S_1 is exactly three.* Then, the algorithm obtains a small set of nodes to dominate the multiple disconnected regions (the yellow regions) resulted after constructing the special MIS in the first phase. Then, in two different steps, the obtained dominators in the first and second steps are connected to form the final CDS. The paper proposes three different approaches to dominate the multiple disconnected regions. The main approach dominates the connected components based on the geometrical properties of the disconnected regions and has an approximation factor of at most 5. To the best of our knowledge, this approximation is the smallest in the literature.

Figure 1 shows a constructed CDS using the classical MIS model in (a) and using the special independent set model with exact-three-hop property in (b). In Figure 1(a), the black nodes are the MIS nodes forming the dominators in the constructed CDS, the intermediate blue nodes (the square nodes) are their connectors, and the other gray nodes are the dominatees or nonbackbone nodes. Figure 1(b), the black nodes all together form the primary dominators or S_1 nodes, satisfying the exact-three-hop properties. Red nodes are the secondary dominators, blue nodes are the connectors, and gray nodes are the dominatees.

For distributed techniques, the first distributed algorithm guaranteeing a constant approximation factor was proposed in [11]. Their algorithm has an approximation factor of 8, $O(n)$ time complexity, and $O(n \log n)$ message complexity. Unfortunately, this algorithm suffers from the complexity of constructing and maintaining a spanning tree in WSNs. Therefore, a number of localized algorithms, that do not require spanning tree, were proposed in [12], [15], [16], [17], and [18] with 192, 172, 147, 50, and 14.93 approximation factors, respectively.

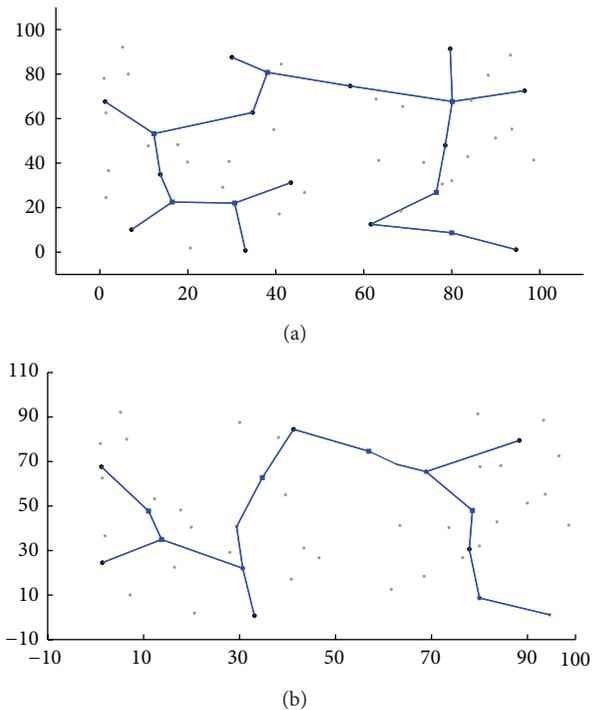


FIGURE 1: The classical MIS-based model with the exact-two-hop property versus the special independent set-based model with the exact-three-hop property. (a) The classical MIS-based model with the exact-two-hop property. (b) The special independent set-based model with the exact-three-hop property.

In this paper, we propose two distributed approximation approaches for CDS construction in WSNs. The proposed approaches extend the exact-three-hop model to the distributed setting. Our algorithms are intended to provide a small approximation factor when compared to their presented centralized version in [4, 5], lower construction cost

(in terms of message and time complexities), and allow efficient backbone maintenance after encountering topology changes. We provide the required performance discussion and experimental results that show the effectiveness of our proposed approaches and compare them to their centralized version.

3. Modeling and Methods

In a distributed environment, the execution of algorithms is event driven. Hence, the construction process of CDS is different from its centralized version in that it overlaps the construction phases. Based on the exchanged messages with its neighbors, a node is preprogrammed to change its state (color) or trigger an event if a specific condition is satisfied. For illustrative purposes, we employ a coloring scheme to differentiate node states during the construction process. The nodes of S_1 (dominators) are marked black. The nodes used to cover the disconnected regions (S_2 nodes) are marked red. Connectors are marked blue and dominatee nodes are marked gray. Other colors (white, orange, and yellow) are temporarily introduced to make the elaboration of the algorithm easier: white is used for initialization, orange to mark nodes at a certain distance to a black node, and yellow to mark disconnected components after S_1 construction. As the information range of this distributed algorithm is at most three, each node records the important changes, specifically the color and region ID, in its 1-hop neighbors. Additionally, each node i keeps the following three lists:

- (i) a *black list*: to store the IDs of its 1-, 2-, and 3-hop black neighbors, as well as their corresponding graph distance to the node i ,
- (ii) a *red list*: to store the IDs of its 1-hop red neighbors,
- (iii) a *region list*: to store the region ID of its 1-hop neighbors.

The described algorithm in the next subsection, *approach I*, is a distributed algorithm for CDS construction that implements the exact-three-hop property. Approach I is based on a spanning tree. Section 3.2 describes the distributed approach II, which is a nonspanning tree based CDS construction method with energy awareness.

3.1. Approach I: Distributed Algorithm for CDS Construction.

The scenario of CDS construction is described as follows: given an arbitrary rooted spanning tree T , we define the tree level of a node u as the number of hops in T between u itself and i , where i is the root of T . All nodes are initially undominated and are colored white. The region ID of each node is set to its own ID. Nodes will be eventually marked with different colors during the execution of the algorithm, and their region IDs might be changed.

The presented state diagram in Figure 2 shows the employed coloring mechanism to construct a CDS using approach I. First, the root node (i) initiates the CDS construction by coloring itself black. Then, the black node i broadcasts a “BLACK” message that includes its ID and the counter value $k = 3$. Upon receiving a “BLACK” message with $k = 3$,

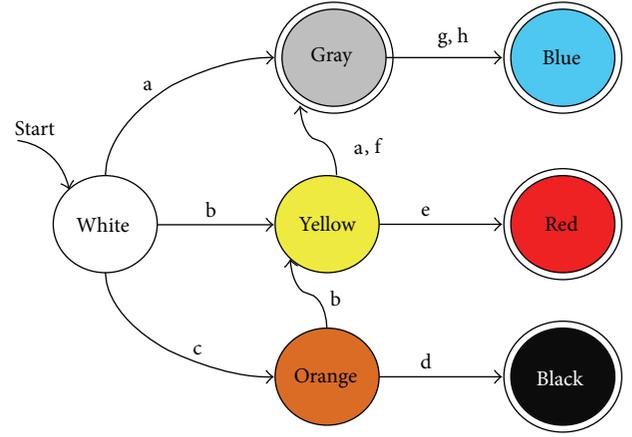


FIGURE 2: The state diagram for the color changes within the execution of approach I.

a white node colors itself gray, updates its black list, decrements k by 1, and rebroadcasts the message with the decremented k value, upon receiving a “BLACK” message with $k = 2$, a white node marks itself yellow, updates its black list, decrements k by 1, and rebroadcasts the message with the decremented k value, and upon receiving a “BLACK” message with $k = 1$, a white node marks itself orange and updates its black list. For the completion of this distributed algorithm, we declare the following events.

3.1.1. Event: Orange Bidding. Orange bidding is done by orange nodes and handled (*evaluated and acknowledged*) by their 1-hop yellow neighbors. The winning orange node is colored *black*. After it is colored orange, each orange node announces its bidding by broadcasting an *orange bidding message* that includes orange node ID, level, and the number of its 3-hop black neighbors. Then, it waits for the acknowledgment from all of its yellow neighbors. Upon receiving an orange bidding from an orange node by one of its neighboring yellow nodes, the receiving yellow node checks the received bidding from all of its 1-hop orange neighbors and selects the orange node that has the lowest level and/or maximum number of black neighbors, and then it acknowledges the winning orange node by broadcasting an *ACK message* with the winning node ID. Upon receiving *ACK messages* from all its yellow neighbors, an orange node colors itself black and broadcasts a “BLACK” message with its ID and $k = 3$. Similar to the previous processing of black messages, upon receiving a “BLACK” message with $k = 3$, a white/orange/yellow node marks itself gray, updates its black list, decrements k by 1, and rebroadcasts the message with the new k value. Upon receiving a “BLACK” message with $k = 2$, a white/orange node marks itself yellow, updates its black list, decrements k by 1 and rebroadcasts the message with new k value. Upon receiving a “BLACK” message with $k = 1$, a white node marks itself orange, updates its black list, and announces its bidding. Upon receiving a “BLACK” message with $k = 1$, an orange node updates its black list and announces its new bidding.

3.1.2. Event: Yellow Bidding. Yellow bidding is done and handled by yellow nodes. The winning yellow node is colored red. Each yellow node that has no orange/white neighbors announces its bidding by broadcasting a *yellow bidding message*. The *yellow bidding message* includes its ID and the number of its 1-hop yellow neighbors (called coverage factor). Then it waits for the acknowledgment from all of its 1-hop yellow neighbors. A yellow node that has no yellow/orange/white neighbors will color itself red without bidding. Upon receiving a *yellow bidding message* by a yellow neighbor, it evaluates and acknowledges the highest bidding (the yellow node that has the highest coverage factor) by broadcasting an *ACK message* with the winning node ID. Upon receiving *ACK messages* from all its 1-hop yellow neighbors, a yellow node colors itself red and broadcasts a “*RED*” *message*. Upon receiving the “*RED*” *message* by a yellow node, it colors itself gray and broadcasts an “*YGRAY*” *message* for its 1-hop neighbors. Upon receiving an “*YGRAY*” *message*, a yellow node recalculates and announces its new yellow bidding. The “*YGRAY*” *message* is very important for the execution of this algorithm in the distributed environment. It notifies yellow neighbors to (i) recalculate their bidding and (ii) to reevaluate and confirm the previously received bidding as required.

3.1.3. Event: Selecting Red Connectors. The distributed logic yields red nodes having the same geometrical properties of the red node in the centralized algorithm: each red node has at least one black node that is only two hops away. Therefore, we need to include at most one gray node to connect each red node to its nearest black node. After a node is colored red, it selects a connector of its 1-hop gray neighbors based on *number of their 1-hop backbone nodes*. As a result, the winning gray node is colored blue and its 1-hop backbone nodes (black, red, and blue) are merged into one region. If a red node already has a blue node within its 1-hop neighbors, it does not need to introduce a new connector. Instead, it links itself to that existing connector by sending a *request-to-connect message* and changes its region ID accordingly.

3.1.4. Event: Changing the Region ID. In this implementation, all nodes with the same region ID form a *connected component*. When a node changes its region ID, it notifies its 1-hop neighbors for this update by sending a *region-change message*. Upon receiving a *region-change message* by a region member (a backbone node that belongs to that region), it changes its region ID and broadcasts a *region-change message* for its 1-hop neighbors.

3.1.5. Event: Gray Bidding. Gray bidding is done by gray nodes and handled by their gray, blue, black, and red neighbors. As a result, the winning gray nodes are colored blue and their 1-hop backbone nodes (black, red, and blue) are merged into one region. Each gray node that has no yellow/orange neighbors and has more than one nearby region announces its gray bidding by broadcasting a *gray bidding message* that includes its ID, its dominator color and region ID, count, and a list of its 1-hop different regions. Upon receiving a *gray*

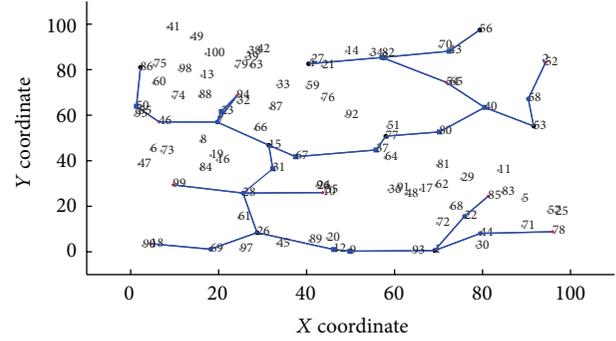


FIGURE 3: An exemplary graph for a network of 100 nodes after constructing a CDS using approach I.

bidding message by a backbone node (blue, red, and black), it decides if it needs to connect to the reported regions by comparing its current region ID with the received region list. If different regions are going to be merged through this gray node, the receiving backbone node sends an *ACK message* to the bidding gray node in order to color it blue. As a result, the acknowledged gray node colors itself blue, calculates the new region ID, and announces it by broadcasting a “*BLUE*” *message*, to its 1-hop backbone nodes in order to merge them all into one region. Upon receiving a *gray bidding message* by a neighboring gray node, it checks whether or not their dominators are black and have different region IDs. If yes, the receiving gray node sends an *ACK message* to the bidding gray node. The bidding gray nodes are evaluated by their neighboring backbone nodes based on their exchanges region list. Upon receiving an *ACK message* from a neighboring gray node, the receiving gray node colors itself blue, calculates the new region ID, and broadcasts a “*BLUE*” *message* with the new region ID. Upon receiving a “*BLUE*” *message* by a neighboring backbone node, a backbone node changes its region ID and broadcasts a *region change message* for its 1-hop neighbors. Upon receiving a “*BLUE*” *message* by a neighboring gray node, it updates its region list. The final CDS is obtained after executing all the triggered events. At that time, all white, orange, and yellow nodes are colored black, red, blue, or gray. The union of the black, red, and blue sets forms the final CDS.

Figure 2 shows the state diagram for the color changes within the execution of approach I. The transition conditions are described as follows: (a) is receiving a “*BLACK*” *message* with $k = 3$, (b) is receiving a “*BLACK*” *message* with $k = 2$, (c) is receiving a “*BLACK*” *message* with $k = 1$, (d) is receiving *ACK* to orange bidding from all yellow neighbors, (e) is receiving *ACK* to yellow bidding from all yellow neighbors, (f) is receiving a “*RED*” *message*, (g) is receiving a “*Request-to-connect*” *message*, and (h) is receiving *ACK* to gray bidding.

Figure 3 shows an exemplary graph for a network of 100 nodes, deployed in a $100\text{ m} \times 100\text{ m}$ square field, after constructing a CDS using approach I. The transmission range of nodes is uniform and assumed to be 20 m. The black, red, and blue nodes with edges between them form the backbone.

The other gray nodes are the dominatees. The gray nodes are dominated by their black or red neighbors. The blue nodes are used to connect the red and black nodes to form the final backbone.

3.2. Approach II: Energy-Efficient CDS Construction. This section discusses the design of approach II, which is an improvement of approach I. The discussed approach in the previous sections (approach I) presents a distributed implementation of the presented centralized algorithm in [4, 5]. These algorithms rely on the construction of a rooted spanning tree, which makes it very costly in terms of communication overhead to maintain the CDS in the case of mobility and topology changes.

In order to eliminate the cost of calculating and maintaining the spanning tree and also to allow efficient maintenance to the constructed CDS, we simplify the ranking function by including the bidding nodes' ID instead of their level. Moreover, we attempt to prolong the lifetime of the constructed CDS by allowing nodes with higher residual energy for being more likely part of the constructed backbone.

In addition to employing a distributed logic to construct the CDS, the energy efficiency of approach II comes from three sources: (i) eliminating the need to *construct* and *maintain* a spanning tree in WSN which affects the time and message complexities of the algorithm (ii) allowing nodes with higher residual energy to be more likely part of the constructed backbone and (iii) allowing efficient maintenance of the backbone in the case of mobility and topology changes; for example, new sensors are introduced or connections to some backbone nodes are lost, as a result of mobility or energy dissipation. In similar scenarios, the improved algorithm does not require maintaining or recalculating the spanning tree to incorporate the encountered topological changes.

Considering approach I as our basic distributed approach, we summarize the following differences between constructing a CDS using approach I and approach II. For orange bidding and evaluation, in approach II, *orange bidding messages* are changed to include orange node ID, *residual energy value*, and a number of its 3-hop black neighbors. Upon receiving an orange bidding message from an orange node by one of its neighboring yellow nodes, a yellow node checks the received bidding from all of its 1-hop orange neighbors, selects the orange node that has the maximum number of black neighbors, residual energy, and/or the lowest ID, and then it acknowledges the winning orange node by broadcasting an *ACK message* with the winning node ID.

For yellow bidding and evaluation, *yellow bidding messages* are changed to include node ID, the value of its residual energy, and the number of its 1-hop yellow neighbors (called coverage factor). The yellow node that has the highest coverage factor and residual energy is selected and acknowledged by its yellow neighbors.

For gray bidding and evaluation, the *gray bidding messages* are changed to include: node ID, its residual energy value, its dominator color and region ID, count and a list of its 1-hop different regions. The bidding gray nodes are evaluated by their neighboring backbone nodes according to their region list and their residual energy value.

4. Results and Discussion

In this section, we discuss the performance of our proposed approaches and report simulation results. We first discuss the approximation factor and analyze the message and time complexities of our proposed approaches. Then, we describe the design of our simulation experiment, including environment setting, simulation input, deployment models, and energy model. Finally, we show simulation results, with each representing an average of 50 runs.

4.1. Performance Discussion. For the approximation factor of approach I, compared to the proposed centralized algorithm in [4, 5], the constructed S_1 and S_2 sets in approach I form *independent sets* in the network and their size is bounded by the size of the MCDS (1 *opt*). The S_2 connectors are chosen by S_2 nodes and are also bounded by the size of the S_2 set, which is 1 *opt*. For S_1 connectors, these connectors are 2-hop connectors. Unlike the centralized algorithm, the upper bound of these connectors is difficult to predict in the distributed environment. As a result of making decisions distributed and the lack of information, some redundant S_1 connectors might be introduced by the algorithm. However, the implemented ranking function tries to minimize this duplication. Under different settings, simulation shows that approach I generates a CDS of a size bounded by 1.5 times than the size of the constructed CDS by its centralized algorithm in [4, 5]. As the approximation factor of this centralized algorithm is bounded by 5 *opt*, these extensive experimental results establish a 7.5 *opt* as an upper bound for the size of the constructed CDS by approach I. Details on experimental results are presented in the next subsection.

The information range shows the amount of neighborhood information that should be collected by a node to perform the CDS construction task. This factor has an influence on algorithm and message complexities. The *information range* of approaches I and II is three.

For time and message complexities, the construction of the spanning tree T depends on the employed algorithm. Spanning tree algorithms are expected to use linear messages and take either a linear or a linearithmic time. After a rooted spanning tree T is constructed, the construction of S_1 and S_2 dominators and their connectors uses linear messages and takes at most linear time. In what follows, we discuss the number of exchanged messages and the number of executed instructions in our approaches as a function of input size.

To analyze the message complexity of our proposed approaches, we consider the different types of exchanged messages and their frequency in worst case scenario. For coloring announcements, some colors (black, red, and blue) are permanent colors and cannot be altered during the execution of the algorithm. Other colors (gray, yellow, and orange) are altered and need to be announced. Hence, during the execution of the algorithm, a node which experiences the different combinations of color changes will broadcast four coloring announcements at most.

For bidding announcements and evaluation, during the execution of approaches I and II, a yellow, orange, and gray

node broadcasts its bidding in a single message which is confirmed by its yellow neighbors using one acknowledgment message. A bidding node x may need to rebid as a response to changes in the state of its neighbors. Assuming d is the maximum node's degree in the network (maximum number of nodes per unit of area), then a node x broadcasts d bidding messages at most. Accordingly, in worst case scenario, a node needs to broadcast: 4 color change announcements, $3 \times d$ bidding messages, and $3 \times d$ acknowledgement messages. As a result, the maximum number of messages broadcasted by a node using our approach is $(4 + 6 \times d) = O(d)$, where d is the maximum degree. As d is bounded by n (the total number of nodes in the network), the maximum number of transferred messages by a node during the execution of our algorithm is bounded by $O(n)$. Therefore, our approach uses linear messages.

For time complexity, each node processes and responds once to $O(d)$ messages and actions in the neighboring area. Assuming each response takes a unit of time, then the complexity of our algorithm is $O(d)$, which is bounded by $O(n)$. Therefore, besides the construction of the spanning tree T , approaches I and II use $O(n)$ messages and take $O(n)$ time. The presented simulation results in the next section confirm this complexity analysis.

4.2. Simulation Setup

4.2.1. Simulation Variables and Setting. In order to compare the size of the generated CDS by the distributed approach (approach I) to the centralized approach that extends it [4, 5], we apply both approaches to the same network topology (the same input). Similarly, in order to study the impact of simplifying the ranking function, we applied approaches I and II to the generated networks. For each generated topology, we ran approaches I and II as well as their centralized version, in order to construct a CDS as well as to collect the total number of messages and the total energy dissipated.

For each reported simulation result, we generated 50 different network topologies. We investigated the performance of all algorithms with different input values for the number of nodes N , the transmission range R , and the deployment area A . We studied the performance with different input values for the numbers of nodes $N \in [100-1000]$ nodes, the transmission range $R \in [10-50]$ m, and the deployment area $A \in [40 \times 40-100 \times 100]$ m².

We used UDG to model the network. Hence, the transmission range for all nodes is unified and equals R . We consider R values that keep the network connected. For the deployment area, the choice of different field sizes for the same input size allows the generation of relatively sparse (for larger squares) and dense (for smaller squares) graphs [17]. For a set of N nodes deployed in a field of area A , with each node having a transmission range R , we define a constant node density $\rho = N/A$, which denotes the expected number of nodes per unit area. We also define the expected number of nodes per transmission unit (or average node degree) $\gamma = \pi R^2 \rho$, where ρ is the expected node density [19].

4.2.2. Deployment Model. In our simulation, we mainly consider the *uniform random deployment model* [20]. For a given 2D square field of area A , we generate a total of N nodes. In a uniform random deployment, each of the N sensors has equal probability of being placed at any point inside the given deployment field. Consequently, the nodes are scattered onto locations that are not known with certainty. For example, such a deployment can result from throwing sensor nodes from an airplane. In general, a uniform random deployment is assumed to be both easy and cost-effective. Because random node deployment is expected in most WSN applications, we mainly included it to assess the performance of our algorithms [20].

4.2.3. Energy Model. In order to calculate the total dissipated energy by the distributed algorithm, we assigned the amount of energy dissipated by transmitter and receiver electronics to process a bit (E_{elec}) to be 50 nJ/bit, the energy dissipated by transmitter amplifier to transmit a bit (E_{ampl}) to be 0.1 nJ/bit, and the maximum size for the control packets to be 64 bits. The energy dissipated to receive (k bits) is calculated as follows [3]:

$$\text{Energy}_{\text{receive}}(k) = (E_{\text{elec}} \times k). \quad (1)$$

The dissipated energy to transmit (k bits) for a distance d is calculated as follows [3]:

$$\text{Energy}_{\text{transmit}}(k, d) = (E_{\text{elec}} \times k) + E_{\text{ampl}} \times k \times d^2. \quad (2)$$

We evaluated the lifetime of the constructed CDS, as a number of rounds, in a network that dissipates a total of E_{round} per round to aggregate data when the size of data packets is 2000 bits. The value of E_{round} is calculated as a function of the size of the constructed CDS, the network size, the transmission cost, and receiving cost of a data packet [3].

In addition to the conventional homogeneous energy model, we implemented a multilevel heterogeneous energy model for the underlying network [3]. In this model, the energy level that is assigned to each sensor is randomly distributed over the close set $[E_0, E_{\text{max}}]$, where E_0 is the lower bound and E_{max} is the upper bound of the initial energy. We consider a fraction m of advanced nodes in a network of N nodes. These advanced nodes are equipped with E_{max} as an initial energy. Hence, the number of advanced nodes in the network equals mN . The value of E_{max} is a times more than the value of the lower bound of the initial energy E_0 . In this simulation we considered the fraction $m = 0.2$, $a = 3$, and $E_0 = 1$ J (Joule).

4.3. Simulation Results. Simulation shows that the implementation of our distributed approaches performs similarly and appropriately in constructing CDS for all network types and densities. It also shows that they satisfy the exact-three-hop property between each black node and its nearest black node. Moreover, red nodes are originally yellow nodes that are marked yellow using one of their 2-hop black neighbors (after receiving a "BLACK" message with $k = 2$). Therefore, this implementation ensures that all red nodes are exactly 2

TABLE 2: The average performance and error margin of approaches I and II and the centralized approach when the number of nodes varies from 100 to 1000 nodes.

Approach	Measure	Number of nodes									
		100	200	300	400	500	600	700	800	900	1000
The Centralized version	Average	27.22	26.52	27.36	26.74	27.04	27.26	26.9	27.06	26.72	27.08
	error	± 0.56	± 0.56	± 0.55	± 0.45	± 0.43	± 0.47	± 0.36	± 0.40	± 0.46	± 0.44
Approach I	Average	33.9	36.02	37.4	39	38.68	39.26	40.14	40	40.54	40.14
	error	± 0.87	± 0.95	± 0.85	± 0.87	± 0.76	± 0.83	± 0.96	± 0.90	± 0.94	± 0.89
Approach II	Average	34.22	35.36	36.54	38.1	38.2	38.1	39.24	38.44	39.02	38.76
	error	± 0.74	± 0.98	± 0.86	± 0.88	± 0.78	± 0.86	± 1.01	± 0.82	± 0.90	± 0.95

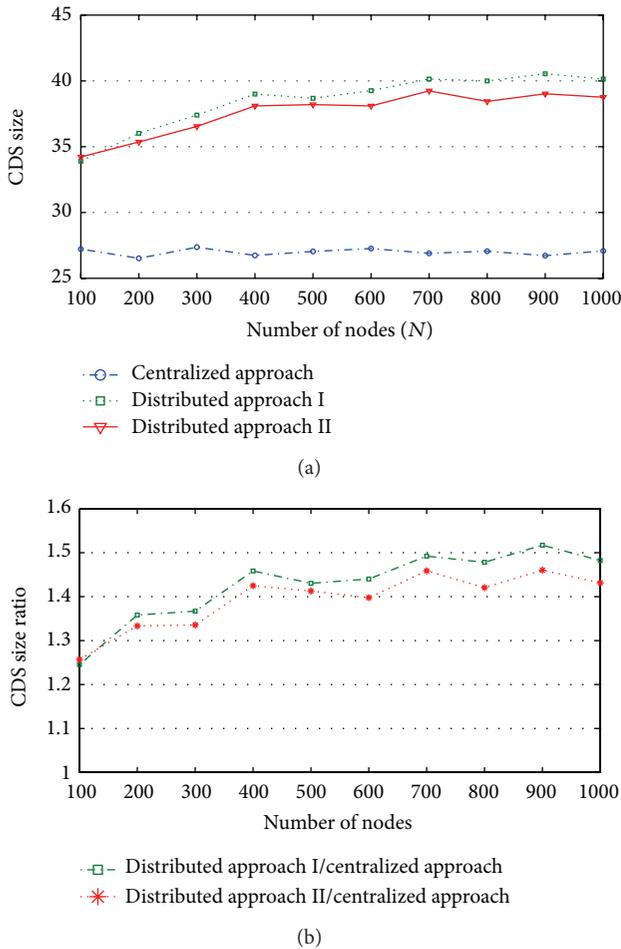


FIGURE 4: The average performance of approaches I and II versus the centralized approach when N varies from 100 to 1000 nodes. (a) Showing the average CDS size. (b) Showing the ratio.

hops from at least one black node. This implementation also satisfies that the count of red connectors is at most equal to the count of red nodes, and the count of black connectors is at most twice the count of black nodes.

Figure 4(a) shows the average CDS size in approaches I and II and the centralized approach when the number of nodes N varies from 100 to 1000 nodes, the transmission range, and deployment area are fixed and equal to

$R = 20$ m and $100 \text{ m} \times 100 \text{ m}$, respectively. Obviously, distributed algorithms produce CDSs of larger sizes because decisions are not made based on the global view of the network. This increase in CDS size is almost entirely as a result of having a larger number of black connectors than in the centralized version. As the figure shows, the two curves of approaches I and II are very close to each other. Therefore, there is no performance loss for the added the energy-efficiency feature. Instead, approach II slightly outperforms approach I. Figure 4(b) shows the ratio between CDS size in approaches I and II to the centralized approach. The corresponding margin of error for approach II against approach I and the centralized version is presented in Table 2.

Considering the energy efficiency and communication complexity of our distributed approaches I and II and the same simulation setting, Figure 5(a) shows the total dissipated energy in nanojoule (nJ) per number of nodes (the average energy dissipation per node). It is shown that energy consumption increases while the number of nodes increases due to the increase of nodes' degrees. Energy consumption is proportional to the number of exchanged messages. Therefore, the increase in nodes' degree increases the number of transferred/received control messages (color announcements, ACK, and bidding) by nodes. Each arrival/transmission of a message requires a certain amount of energy to be dissipated by the receiving/transmitting node. For message complexity, the corresponding average message complexity per number of nodes to above results is shown in Figure 5(b). The corresponding average node per unit of area (node degree) for the presented variation in number of nodes from 100 to 1000 is 12.5664, 25.1328, 37.6992, 50.2656, 62.8320, 75.3984, 87.9648, 100.5312, 113.0976, and 125.6640 nodes, respectively. Comparing the average number of transmitted messages per node to the average node degree in different setting, we identify that the average number of transmitted messages per node is bounded by the average node degree for the different variations of network setting. This comparison is based on the average number of exchanged messages by the algorithm and the average node degree. The average node degree is bounded by the number of nodes in the network (N). Therefore, our analysis for the message complexity of approach I as $O(n)$ is correct.

For varying the transmission range, Figure 6 shows the impact of changing the transmission range R from 10 m to

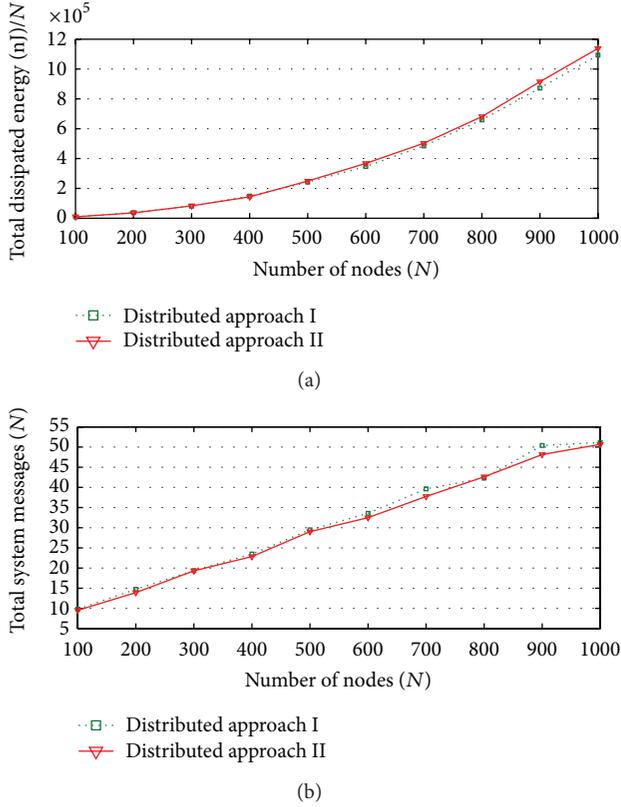


FIGURE 5: Energy efficiency and communication overhead of approaches I and II. (a) The average energy dissipation per node in (n). (b) The average number of transmitted messages per node.

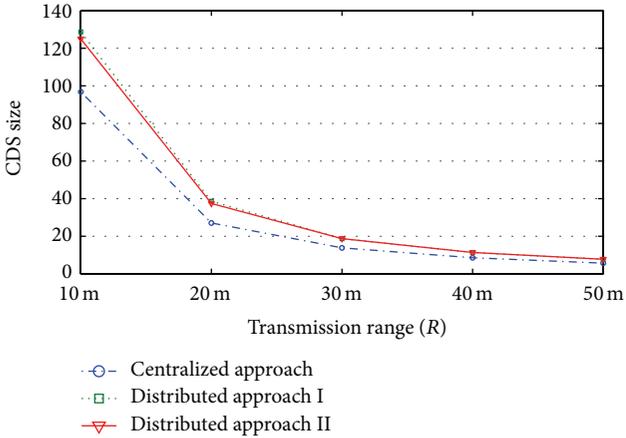


FIGURE 6: The average performance when the transmission range R changes from 10 m to 50 m.

50 m, when $N = 400$ and $A = 100 \text{ m} \times 100 \text{ m}$. Each time we increment R value, we move from sparse network toward dense network. The CDS size decreases each time we increase R value. The transmission range is inversely proportional to the size of CDS. The increase in transmission range allows a dominator node to dominate farther nodes. It also allows a connector node to connect many dominators at the same

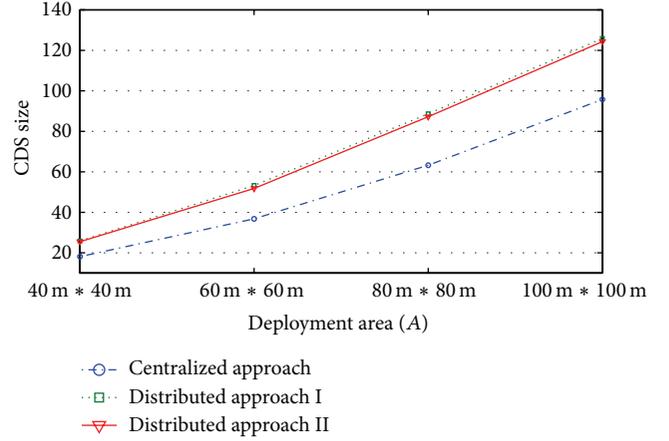


FIGURE 7: The average performance when the deployment area changes from $40 \text{ m} \times 40 \text{ m}$ to $100 \text{ m} \times 100 \text{ m}$.

time. Hence, the CDS size is reduced as a result of having fewer dominators and connectors. This relationship is valid for any CDS construction algorithm. Simulation also shows that there is no performance loss, in terms of CDS size, in approach II when compared to approach I. Their curves are almost identical.

For the deployment area, for $N = 400$ nodes and $R = 10 \text{ m}$, we changed the size of the deployment field between $40 \text{ m} \times 40 \text{ m}$ and $100 \text{ m} \times 100 \text{ m}$. The choice of different field sizes for the same N and R allows generating a relatively sparse (for larger squares) and dense networks (for smaller squares). In this particular example, the corresponding average number of nodes per unit area is 78.5, 34.8, 19.6, and 12.5 for the field sizes, respectively. Results are shown in Figure 7.

Considering approach II and the multilevel heterogeneous network, Figure 8 shows the estimated lifetime (in number of round): (a) when N varies between 200 and 1000 nodes, $R = 10 \text{ m}$ and $A = 100 \text{ m} \times 100 \text{ m}$ and (b) when R varies between 10 m and 50 m, $N = 1000$ nodes, and $A = 100 \text{ m} \times 100 \text{ m}$. It compares the lifetime of nodes in the constructed CDS compared to the lifetime of other nodes in the network (non-CDS nodes). It uses the first node dies (FND) metric to express the lifetime in a number of rounds. As the figure shows, the constructed CDS using approach II obtains at least 50% extra rounds than other nodes in the network.

5. Conclusions

In this paper, we propose two distributed approaches for CDS construction in wireless sensor networks. Approach I aims to construct a small CDS in distributed setting. It extends the exact-three-hop property construction model that was introduced and implemented in centralized setting in [4, 5] to the distributed environment. Approach II aims to prolong the lifetime of the constructed CDS and achieve energy-efficient construction and maintenance. It implements a nonspanning tree-based CDS construction approach with energy awareness properties.

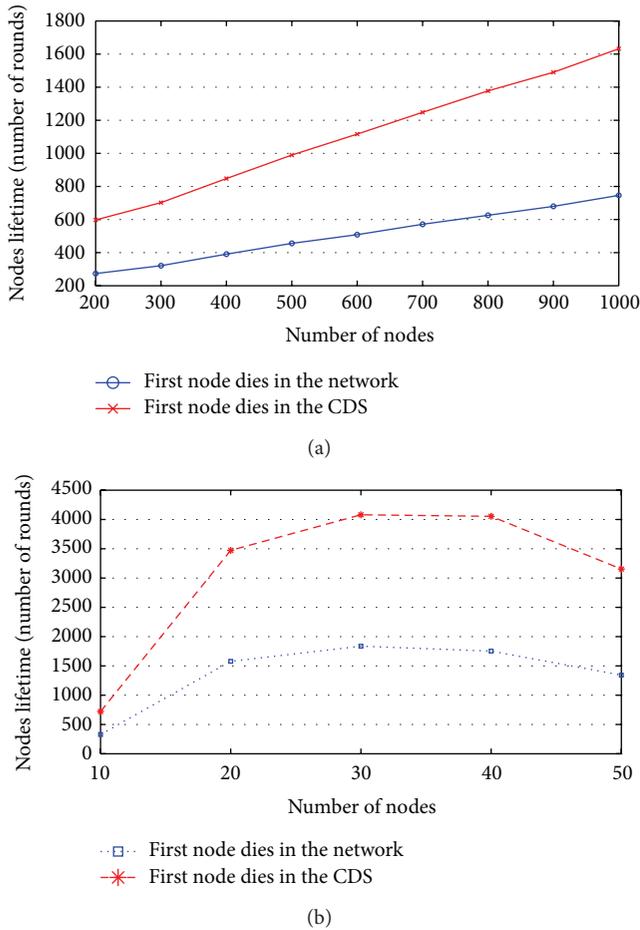


FIGURE 8: The average lifetime (as number of rounds) of the first node dies (FND) in the constructed CDS using approach V versus the FND of the other non-CDS nodes (the rest of network nodes) in the multilevel heterogeneous networks.

Throughout the simulation, we study the performance of our proposed distributed approaches and compare them to their centralized version [4, 5]. Simulation shows that our distributed construction approaches have a maximum size ratio of 1.5 to the centralized approach, and that they satisfy all the geometrical properties of the exact-three-hop property construction model. Based on this ratio, our distributed approaches achieve an approximation factor of 7.5 from the optimal CDS. To the best of our knowledge, this approximation outperforms the existing distributed CDS construction algorithms.

For the lifetime of the constructed CDS, considering a multilevel heterogeneous network, simulation shows that the constructed CDS using approach II obtains at least 50% extra rounds than the first node dies of other nodes (non-CDS nodes) in the network. Simulation also shows that approach II does not incur any increase in CDS size or communication overhead as a result of simplifying its ranking function to introduce energy efficiency.

In future, we consider applying a pruning algorithm to remove the redundant connectors from the final backbone. Furthermore, we plan to consider another source for

heterogeneity by assuming sensor nodes to have variable transmission ranges.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through research group no. RGP-VPP-264.

References

- [1] J. Blum, M. Ding, A. Thaler, and X. Cheng, "Connected dominating set in sensor networks and MANETs," in *Handbook of Combinatorial Optimization*, D. Z. Du and P. Pardalos, Eds., vol. 5, pp. 329–369, Springer, New York, NY, USA, 2005.
- [2] N. Al-Nabhan, M. Al-Rodhaan, A. Al-Dhelaan, and X. Cheng, "Connected dominating set algorithms for wireless sensor networks," *International Journal of Sensor Networks*, vol. 13, no. 2, pp. 121–134, 2013.
- [3] L. Qing, Q. Zhu, and M. Wang, "Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks," *Computer Communications*, vol. 29, no. 12, pp. 2230–2237, 2006.
- [4] N. Al-Nabhan, B. Zhang, M. Al-Rodhaan, and A. Al-Dhelaan, "Two connected dominating set algorithms for wireless sensor networks," in *Wireless Algorithms, Systems, and Applications*, vol. 7405 of *Lecture Notes in Computer Science*, pp. 705–713, Springer, Berlin, Germany, 2012.
- [5] N. Al-Nabhan, B. Zhang, X. Cheng, M. Al-Rodhaan, and A. Al-Dhelaan, "Three connected dominating set algorithms for wireless sensor networks," *International Journal of Sensor Networks*. In press.
- [6] Y. Li, M. T. Thai, F. Wang, C.-W. Yi, P.-J. Wan, and D.-Z. Du, "On greedy construction of connected dominating sets in wireless networks," *Wireless Communications & Mobile Computing*, vol. 5, no. 8, pp. 927–932, 2005.
- [7] Z. Liu, B. Wang, and Q. Tang, "Approximation two independent sets based connected dominating set construction algorithm for wireless sensor networks," *Information Technology Journal*, vol. 9, no. 5, pp. 864–876, 2010.
- [8] Y. Li, D. Kim, F. Zou, and D.-Z. Du, "Constructing connected dominating sets with bounded diameters in wireless networks," in *Proceedings of the 2nd Annual International Conference on Wireless Algorithms, Systems, and Applications (WASA '07)*, pp. 89–94, August 2007.
- [9] F. Zou, X. Li, D. Kim, and W. Wu, "Construction of minimum connected dominating set in 3-dimensional wireless network," in *Wireless Algorithms, Systems, and Applications*, Y. Li, D. Huynh, S. Das, and D. Z. Du, Eds., vol. 5258 of *Lecture Notes in Computer Science*, pp. 134–140, Springer, Berlin, Germany, 2008.
- [10] P.-J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," *Mobile Networks and Applications*, vol. 9, no. 2, pp. 141–149, 2004.

- [11] K. 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, Big Island, Hawaii, USA, 2002.
- [12] K. M. Alzoubi, P.-J. Wan, and O. Frieder, "Message-optimal connected dominating sets in mobile ad hoc networks," in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '02)*, pp. 157–164, Lausanne, Switzerland, June 2002.
- [13] 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*, pp. 251–255, Durham, NC, USA, March 2002.
- [14] S. Butenko, X. Cheng, D. Z. Du, and P. Pardalos, "On the construction of virtual backbone for ad hoc wireless networks," *Cooperative Control: Models, Applications and Algorithms*, vol. 1, pp. 43–54, 2003.
- [15] Y. Li, S. Zhu, M. T. Thai, and D. Z. Du, "Localized construction of connected dominating set in wireless networks," in *Proceedings of the NSF International Workshop on Theoretical Aspects of Wireless Ad Hoc, Sensor and Peer-to-Peer Networks (TAWN '04)*, pp. 11–19, Chicago, Ill, USA, 2004.
- [16] X. Cheng, M. Ding, D. H. Du, and X. Jia, "Virtual backbone construction in multihop ad hoc wireless networks," *Wireless Communications & Mobile Computing*, vol. 6, no. 2, pp. 183–190, 2006.
- [17] K. Islam, *Energy aware techniques for certain problems in wireless sensor networks [Ph.D. thesis]*, Queen's University, Kingston, Canada, 2010.
- [18] D. Kim, Z. Zhang, X. Li, W. Wang, W. Wu, and D.-Z. Du, "A better approximation algorithm for computing connected dominating sets in unit ball graphs," *IEEE Transactions on Mobile Computing*, vol. 9, no. 8, pp. 1108–1118, 2010.
- [19] C. Bettstetter, "On the minimum node degree and connectivity of a wireless multihop network," in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '02)*, pp. 80–91, Lausanne, Switzerland, June 2002.
- [20] W. Y. Poe and J. B. Schmitt, "Node deployment in large wireless sensor networks: coverage, energy consumption, and worst-case delay," in *Proceedings of the Asian Internet Engineering Conference (AINTEC '09)*, pp. 77–84, Bangkok, Thailand, November 2009.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

