A Distributed Energy Aware Connected Dominating Set Technique for Wireless Sensor Networks

Azrina Abd Aziz*† Y. Ahmet Şekercioğlu*
*Department of Electrical and Computer Systems Engineering, Monash University, Australia
†Department of Electrical and Electronic Engineering, Universiti Teknologi Petronas, Malaysia

Abstract—**This paper presents an energy-efficient algorithm based on the connected dominating set (CDS) for a wireless sensor networks (WSN). A CDS has been widely used for a virtual backbone construction to support routing and minimize the communication overheads in a network. Our algorithm, a single-phase single initiator (SPSI), only requires a single phase to construct a CDS in a distributed manner using localized information. It generates a small CDS with low message overhead and lower energy consumption. Furthermore, it does not rely on a global positioning system (GPS) to operate and can cope with the presence of unidirectional links. Therefore, it is simple and practical to implement. The performance of our algorithm is confirmed through simulations and validated against two leading CDS algorithms. The simulation results revealed that our algorithm is effective in reducing the CDS size, message overhead and energy consumption.**

I. Introduction

A wireless sensor network (WSN) is a dynamic network consisting of tiny sensor nodes typically designed for event monitoring. The sensor nodes work cooperatively to self-organize and self-configure the network without relying on a fixed infrastructure. This flexible feature makes WSNs attractive for numerous applications ranging from military, civil, industrial to health.

Due to the lack of physical infrastructure, sensor nodes have to form a temporary infrastructure to support various tasks including routing. One way to do this is to use a connected dominating set (CDS), which is derived from the graph theory concept [1]. The CDS forms a virtual backbone in the network and organizes the nodes into a hierarchical structure. It provides scalability, simplifies network management and guarantees network connectivity.

In a network, a set of nodes is defined as a dominating set (DS) if all nodes are either in the set or have a neighboring node in the set. To create the network backbone, the dominating set must remain connected in the network. This connected DS is referred to a CDS.

Constructing and maintaining the CDS are challenging tasks because of the limited energy resources available in the networks [2]. Particularly, sensor nodes usually operate using lightweight batteries that are difficult to be replaced or recharged once deployed in a field. Therefore, CDS algorithms must be energy-efficient to preserve the network lifetime. Efficient algorithms should generate a small CDS size, consume low message overhead, operate in a distributed approach and use localized information gathered through exchanged messages.

Various CDS algorithms [3]–[6] have been developed to tackle the energy conservation issue in WSNs. However, these algorithms have several shortcomings. First, they operate based on unrealistic assumptions such as nodes must be equipped with a global positioning system (GPS) to compute their exact positions or nodes can always communicate with neighbors without any interruption. Second, although these works are proposed to prolong the network lifetime, nodes are required to exchange a significant number of messages to form the CDS. This operation consumes high energy which can drain the energy resources of nodes, thus shorten the lifetime of the entire network.

In this paper, we present a novel energy-efficient CDS algorithm for WSNs called a single-phase single initiator (SPSI). SPSI creates a small backbone and significantly reduces the number of exchanged messages required for constructing the CDS. This feature contributes to the low energy consumption in the network. Other great features of our algorithm are: (1) it does not require nodes to know their accurate positions, by means of a GPS, (2) it does not operate on the unit disk graph (UDG), hence nodes are not homogeneous and (3) the physical layer effects are accurately modeled using a realistic simulation tool. Therefore, our algorithm is practical to implement in a realistic environment.

The rest of this paper is organized as follows. Section II briefly discusses the state-of-the-art of the CDS techniques proposed so far. Section III presents definitions used for the SPSI algorithm. Section IV discusses the detailed implementation of our algorithm. Section V provides the theoretical performance of the algorithm that includes the time complexity, message complexity and approximation factor. Section VI describes the simulation results. Finally, Section VII concludes our work and discusses our future work.

II. Related CDS Algorithms

Numerous algorithms for constructing a CDS have been surveyed in [7]. In general, existing CDS algorithms can be categorized into distributed or centralized approaches. Distributed approaches are preferable than the centralized approaches due to their ability to deal with the dynamic nature of the networks, in which the links between nodes are constantly updated when nodes move or die. Centralized approaches on the other hand operate based on the assumption that the global information is available. In these approaches, a centralized node is responsible for gathering information from

978-1-4577-1967-7/12/$26.00 ©2011 IEEE

all nodes in the network. Thus, they usually incur a significant amount of message overhead. For this reason, this section focuses on the distributed CDS algorithms.

A simple distributed and localized algorithm is proposed in [8]. The algorithm constructs a CDS in two phases using a marking process. Initially all nodes are unmarked. During the first phase, a node that has two unconnected neighbors is marked as a dominator. The marked dominators become a CDS and the size of the generated CDS is usually not small. In the second phase, two rules are used to prune the redundant CDS. The pruning rules state that if node u has a neighbor with higher ID which can cover all of its neighbors or if u has two connected neighbors with higher ID which can cover all of its neighbors, u can be eliminated from the CDS. The authors state that the advantages of this algorithm are its simplicity and quick computation of the CDS. It outperforms some of the existing algorithms such as [9]. Wu et al. later propose a power aware CDS (PACDS) algorithm [10] to address the energy limitation of WSNS. It aims to achieve two goals. The first goal is to construct a smaller size CDS while the second goal is to prolong the node's lifetime. From the authors' observation, the CDS nodes are typically overloaded with various tasks and they are the first to drain energy. To overcome this problem the role of the CDS is alternated between nodes with higher residual energy. The formation of the CDS also involves two phases. During the first phase, a marking rule similar to the one in [8] is used in which a node that has two unconnected neighbors will become a CDS. During the second phase, additional pruning rules that consider the node degree and energy level are introduced to calculate the CDS. However, the above algorithms require a significant amount of exchanged messages to construct the CDS, which could drain the energy reserves in the network.

Yuanyuan et al. [6] present an energy-efficient CDS (ECDS) algorithm to solve the energy constraints in wireless sensor networks and minimize the size of a CDS. Contrary to the algorithm in [10], the ECDS does not use any pruning rule. Instead, it first constructs a dominating set called a maximal independent set (MIS) and then finds gateway nodes to connect the MIS. The ECDS algorithm is based on a coloring technique and uses a weight metric to select the CDS and connector nodes. The weight is calculated based on the node's residual energy and the node degree. The drawback of the ECDS is its significant message complexity in acquiring neighbor's weight and updating nodes' status.

III. PRELIMINARIES

In our algorithm, each node periodically broadcasts beacons to gather one-hop and two-hop neighbors. We assume that all nodes in the network have a unique identifier (ID) and they are static nodes. We present some definitions related to our work below.

A. *Network Model*

An $undirected\ graph$ $G = (V, E)$ is used to represent the WSN, where V is a set of sensor nodes in the network, called vertices and E is a set of a communication link between a pair of sensor nodes, called edges usually denoted as $(u, v) \in E$. Two vertices u and v are neighbors if (1) they are within their maximum transmission range R_{max} and (2) the communication links between them are symmetrical.

B. *Dominating Set (DS) and Connected Dominating Set (CDS)*

In the graph $G = (V, E)$, a dominating set (DS) is a subset $S \subseteq V$ if and only if every node is either in S or has at least one neighbor in S. A connected dominating set (CDS) is a dominating set of G which induces a connected subgraph of G.

IV. SPSI ALGORITHM DESCRIPTION

The SPSI requires a single phase to generate a CDS. Nodes in the CDS are called dominators while the nonCDS nodes are referred to dominatees. The aim of the SPSI is to generate a small set of dominators while keeping the message overhead low. Each dominator finds its local set of connectors among its one-hop neighbors to add into the CDS. The chosen connectors then become dominators and continue the connectors selection. This process is performed greedily until all nodes are either in the CDS or covered by a dominator in the CDS.

A. *Connector Set Selection*

To minimize the CDS size, it is essential to keep the number of connectors small. To achieve this, we enhance the greedy MPR algorithm in [11] to reduce the CDS size and conserve energy. The idea is to exploit the two-hop neighborhood information. Each dominator u has a knowledge of its one-hop neighbors and two-hop neighbors. It will choose the connectors among its one-hop neighbors that can cover the largest number of two-hop neighbors. This rule generates a minimum number of connectors.

The connector set for a given node u is computed as in Algorithm 1. Let u denote the dominator initiating the connector selection, $N_1(u)$ denote the one-hop neighbors of u, $N_2(u)$ denote the two-hop neighbors of u and $C(u)$ denote the set of chosen connector of node u. The term uncovered node refers to node in $N_2(u)$ that is not covered by $C(u)$. Let *span*(v) denote the quality of a potential connector v used in case of a tie. The *span*(v) is computed by

$$span(v) = N_{\text{total}} E_r(v), \quad (1)$$

where N_{total} is the total number of one-hop neighbors of node v and $E_r(v)$ is the residual energy of node v.

Figure 1 illustrates the process of finding a connector set for dominator 4. Dominator 4 has to find connector nodes that can cover its two-hop neighbor nodes 6, 7 and 8. Therefore, it chooses node 1 as the connector for the two-hop neighbor node 6 and node 0 as the connector for the two-hop neighbor nodes 7 and 8. Node 2 is not chosen because node 8 is already covered by node 0. Hence, the connector set of dominator 4 is Connector(4)= {0,1}.

Algorithm 1 The enhanced MPR Algorithm

1. Add $v \in N_1(u)$ to $C(u)$, if v is the only node that covers node $w \in N_2(u)$. Remove connector v from $N_1(u)$ and also the nodes covered by v from $N_2(u)$.
2. Add $v \in N_1(u)$ to $C(u)$, if v covers the largest number of uncovered nodes in $N_2(u)$. If there is a tie, choose the node with the largest *span(v)* to break the tie. Use node ID to break another tie.

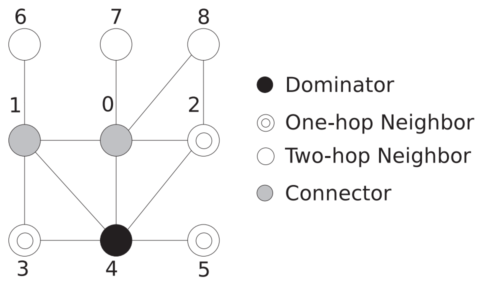


Fig. 1. An example of finding a connector set for a given dominator 4.

B. CDS Construction

The CDS construction begins with a neighbor discovery process. Each node u sends a beacon to gather symmetrical neighbors. All nodes are initially set to an uncovered state. At the end of the CDS construction, they will remain either in a dominatee or a dominator state. Figure 2 shows the state transition diagram of a node during the construction of the CDS.

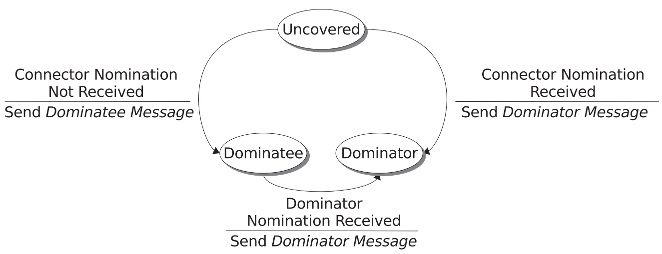


Fig. 2. The state transition diagram of a node in the SPSI algorithm.

After collecting the neighborhood information, the initiator changes its state to a dominator and sends a *Dominator Message*. For simplicity, we choose the node with the smallest node identifier (id) as the initiator. The dominator then computes its connector set using the rules described in Algorithm 1. It then broadcasts the connector selection to all neighbors. The uncovered or dominatee node which is chosen as a connector changes its state to a dominator, then computes the connectors and finally broadcasts the connector selection along with its updated state to neighbors. The uncovered node which is not chosen as a connector changes its state to a dominatee and broadcasts a *Dominatee Message* to neighbors. The dominator nodes will repeat the search of the connectors until there is no uncovered node left in the network.

V. Performance Evaluation

Theorem 1. *SPSI has $O(3\Delta C + 3\Delta)$ time complexity and $O(n)$ message complexity, where n is the total number of nodes in the network, C is the number of chosen connectors and Δ is the maximum node degree in the network.*

Proof. The time complexity of the SPSI algorithm is the time taken for computing the connector set in the network. Since the computation of connectors is based on the MPR algorithm [11], thus the time complexity of the SPSI algorithm is $O(3\Delta C + 3\Delta)$, which can be proven similarly as in [12].

The overall message complexity of SPSI is $O(n)$ since each node u exchanges exactly one message, either a *Dominatee Message* or a *Dominator Message* to build the CDS.

Theorem 2. *SPSI algorithm has an approximation factor of at most $O(log(\triangle))$, where $\triangle$ is the maximum number of neighbors in the network. Under sparse networks, the approximation factor is within a small constant factor.*

Proof. The SPSI algorithm uses the enhanced MPR rule to find a connector set for building the backbone in a network. This process involves finding the two-hop neighbors covered by each node in the neighbor set. As proven in [13], the algorithm that computes a connector set using this approach has an approximation factor bounded by $log(\triangle)$, where $\triangle$ is the maximum number of neighbors in the network. Since, the SPSI algorithm uses the same approach as in [11], it approximates the CDS size of at most $O(log(\triangle))$ times the CDS size of MCDS. Although it does not guarantee a small CDS size, the finding in [11] shows otherwise. Under sparse networks, the approximation factor is within a small constant factor. For example, for a network consisting of 100 nodes, the approximation factor of the proposed algorithm is below a constant value of 4.7 [11]. This approximation factor is comparable to the approximation factor of the TPSI algorithm.

VI. Simulations and Discussions

In this section, we present our simulation results. We evaluated the performance of our algorithm in terms of the following metrics: (1) the size of the CDS, (2) the number of messages required for constructing the CDS and (3) the energy consumption involved for building the CDS. We compared the SPSI algorithm with the ECDS [6] and PACDS [5] algorithms.

For each metric, we investigated the performance of our algorithm against the ECDS and PACDS algorithms in two aspects: the network size and the network density (node degree). The network density metric measures the scalability of the algorithm.

The simulations were conducted using the discrete event simulator OMNeT++ (version 4.1) [14]. We used the MiXiM framework [15] to incorporate the detailed modeling of physical layer effects such as path loss (value of 3) and signal interference to ensure our algorithm works in practice. For the simulations, we generated realistic topologies that are extended from a small network calibrated using the testbed in our department. Three scenarios of topologies were created to represent sparse network (node degree 4), medium network (node degree 8) and dense network (node degree 12). The number of nodes N in each scenario varies from 100 to 500 with an interval of 100, and 10 runs were conducted for each N to gain a 95% confidence interval. The nodes were deployed in a 2D dimensional space and their transmission range varies.

A. The Size of the CDS

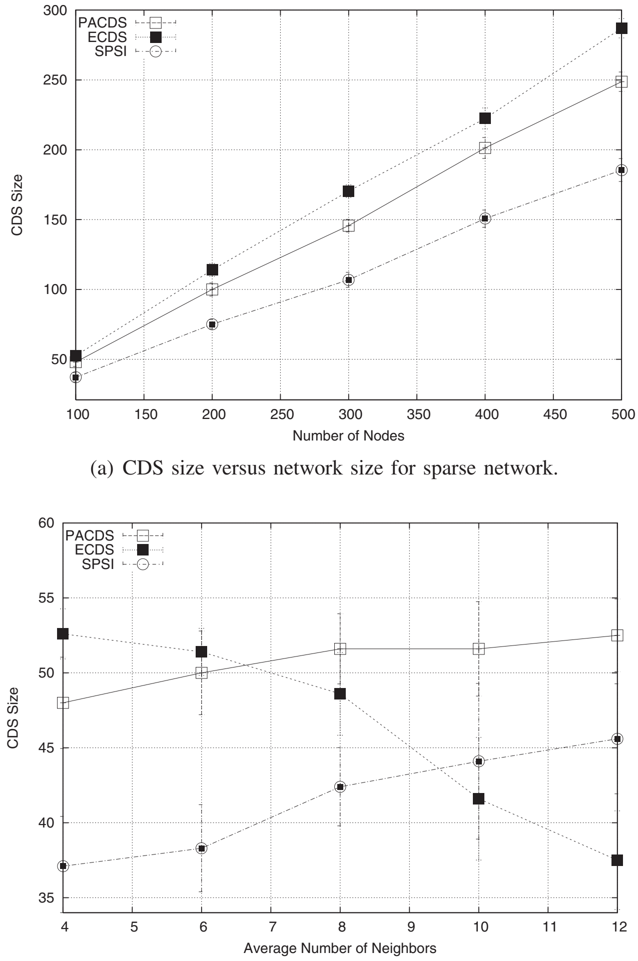
Figure 3(a) shows the efficiency of the algorithms in terms of the number of nodes generated for the CDS as the network size increases from 100 to 500 for a sparse network. It is clear that SPSI generated the smallest CDS size for sparse and medium networks compared to the ECDS and PACDS. However, in dense network, the ECDS created the lowest CDS size.

The performance of SPSI with respect to node degree is shown in Figure 3(b). SPSI has the best performance over the ECDS and PACDS algorithms when the network is sparse. In dense network, SPSI generates a smaller CDS than the one formed by the PACDS but at slightly higher number of CDS compared to ECDS. The PACDS algorithm always creates a larger CDS size because it uses a simple but less efficient rule for finding a CDS. Therefore, it requires a self-pruning method to minimize the CDS size. SPSI and ECDS both adopt a node degree rule for choosing the CDS. Since nodes with more neighbors (larger node degree) are chosen as a CDS, more nodes can be covered by these nodes, thus less nodes are required for building the CDS.

B. Message Overhead

Figure 4(a) shows the message overhead of the SPSI with respect to network size for a sparse network. The message overhead was evaluated based on the number of messages sent by nodes during the CDS construction. SPSI requires a significantly lower message overhead compared to ECDS and PACDS. The efficiency of SPSI is clearly shown in Figure 4(b). Its message overhead remains constant although the node degree increases. The message overhead of the ECDS and PACDS on the other hand, increases linearly as the node degree increases. In dense network, SPSI significantly reduces the number of messages in ECDS and PACDS up to 99.6% and 99.5% respectively.

The efficiency of the SPSI algorithm is due to a small number of messages exchanged for constructing the CDS. Unlike in ECDS, nodes do not compete and exchange messages to join the CDS. Instead they only need to perform local check for the nomination of a connector and send exactly one message. In the case of PACDS, the high message overhead is caused by the pruning process and it is more significant when the network becomes denser since more nodes need to be checked for pruning. In addition, SPSI constructs the CDS in one phase as opposed to ECDS and PACDS, both require two phases to form the CDS. Therefore, the number of exchanged messages drops tremendously.



(a) CDS size versus network size for sparse network.

(b) CDS size versus network density for 100 nodes network size.

Fig. 3. CDS size comparison with respect to network size and node degree.

C. Energy Consumption

This section measured the amount of energy involved for constructing the CDS by considering the energy dissipation of sending and receiving packets activities as in [16]. Every time a node sends a packet, its transmit energy $E_{\mathrm{TX}}(k,d)$ drops by

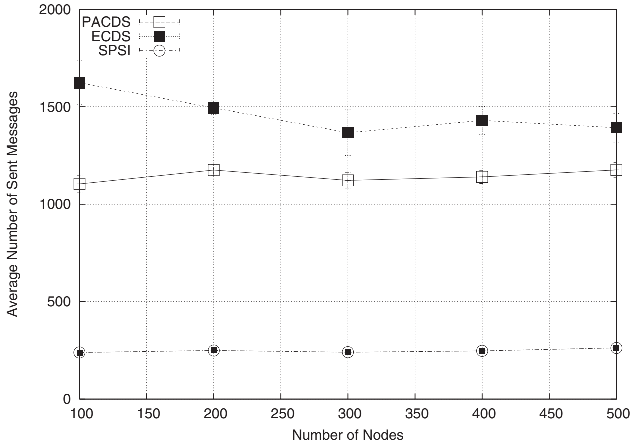
$$E_{\mathrm{TX}}(k,d) = \delta k + \gamma k d^2, \quad (2)$$

where δ is the receiver circuitry constant assumed as 50 nJ/bit, γ is the transmit amplifier constant assumed as 100 pJ/bit/m^2, d is the transmit distance and k is the packet length assumed as 2000 bits long.

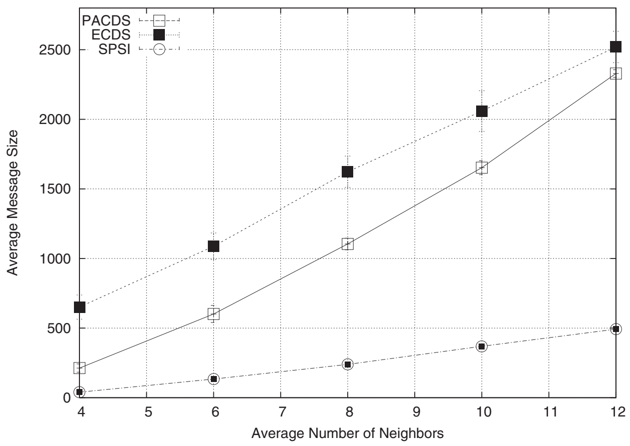
The energy consumption for receiving packets $E_{\mathrm{RX}}(k)$ is computed by

$$E_{\mathrm{RX}}(k) = \delta k. \quad (3)$$

The results of Figures 5(a) and 5(b) show that the SPSI constructs the CDS using a low and bounded energy consumption. This can be explained by the low number of messages exchanged between nodes. This shows that the SPSI algorithm is scalable and can be used for a large network deployment. The ECDS and PACDS algorithms however, consume a sig-

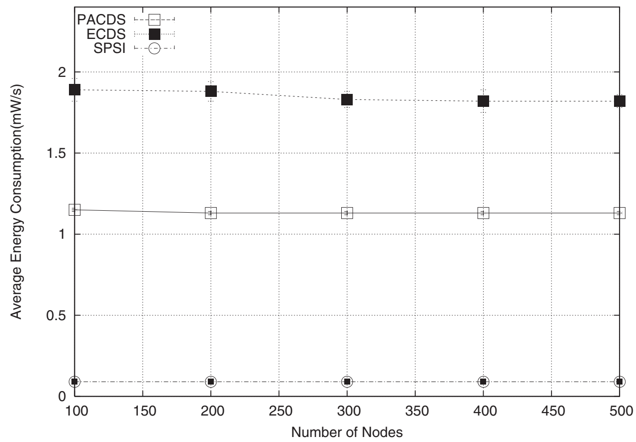


(a) Message overhead versus network size for medium network.

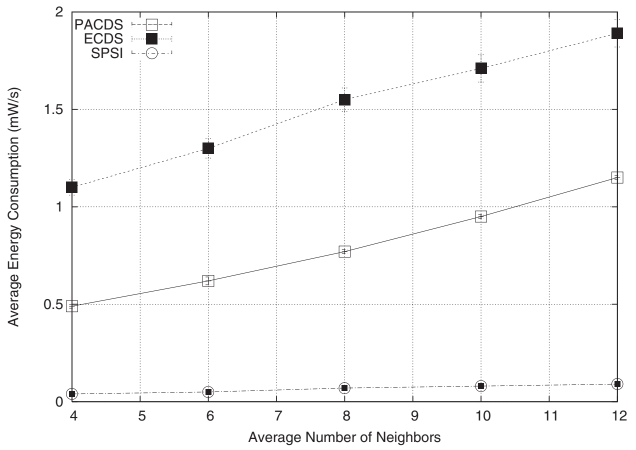


(b) Message overhead versus network density for 100 nodes network size.

Fig. 4. Message overhead comparison with respect to network size and node degree.



(a) Energy consumption versus network size for dense network.



(b) Energy consumption versus network density for 100 nodes network size.

Fig. 5. Energy consumption comparison with respect to network size and node degree.

nificant amount of energy and their energy increases linearly with the number of neighbors.

VII. Conclusion

In this paper, we have proposed an energy-efficient CDS algorithm called SPSI that can construct a CDS in a single phase as opposed to two phases. The SPSI limits the number of exchanged messages among nodes and keeps the CDS size small. We compared our algorithm with ECDS and PACDS in terms of CDS size, energy consumption and message overhead. The simulation results indicate that our algorithm outperforms the ECDS and PACDS. Our future work will focus on the effects of mobility and CDS reconstruction on the network lifetime.

References

[1] N. Biggs, E. Lloyd, and R. Wilson, *Graph Theory*. Oxford University Press, 1936.

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

[3] 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.

[4] Y. Ma, Y. Guo, and M. Ghanem, "Reca: Referenced energy-based cds algorithm in wireless sensor networks," *International Journal of Communication Systems*, vol. 23, no. 1, pp. 125–138, 2010.

[5] 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.

[6] Z. Yuanyuan, X. Jia, and H. Yanxiang, "Energy efficient distributed connected dominating sets construction in wireless sensor networks," in *Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing*. ACM, 2006, p. 802.

[7] J. Blum, M. Ding, A. Thaeler, and X. Cheng, "Connected dominating set in sensor networks and manets," in *Handbook of Combinatorial Optimization*, D.-Z. Du and P. M. Pardalos, Eds. Springer US, 2005.

[8] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*. ACM, 1999, p. 14.

[9] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, no. 4, pp. 374–387, 1998.

[10] J. Wu, M. Gao, and I. Stojmenovic, "On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless

networks," in *International Conference on Parallel Processing, 2001.*, Sep. 2001, pp. 346 – 354.
[11] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast messages in mobile wireless networks," in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences, 2002. HICSS.* IEEE, 2002, pp. 3866–3875.
[12] O. Liang, Y. A. Sekercioglu, and N. Mani, "A survey of multipoint relay based broadcast schemes in wireless ad hoc networks," *IEEE Communications Surveys and Tutorials*, vol. 8, no. 4, pp. 30–46, 2006.
[13] L. Viennot, "Complexity results on election of multipoint relays in wireless networks," *Report RR-3584, INRIA*, 1998.
[14] "Omnet++ discrete event simulator," http://www.omnetpp.org/.
[15] "MiXiM simulator for wireless and mobile simulations," http://mixim.sourceforge.net/.
[16] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Hawaii International Conference on System Sciences*, vol. 8. Citeseer, 2000, p. 8020.