A Distributed Topology Control Method for Improving Energy Efficiency of 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*—We present a distributed topology control method that builds a virtual backbone in a wireless sensor network (WSN). It aims to reduce the communication overhead in the network. The method is based on the Connected Dominating Set (CDS) concept, which takes energy reserves of the individual nodes into consideration when constructing the virtual backbone. To achieve this, we have developed a novel algorithm called three-phase single initiator (TPSI). By using the TPSI algorithm, our method can build the virtual backbone in a fully distributed manner using localized information. It creates a small-size backbone, has low message overhead, and ensures that only the nodes with sufficient energy capacity are selected. It also allows the energy-exhausted nodes in the backbone to be replaced with new ones. Simulation results show that our algorithm performs better than the two leading distributed CDS algorithms with respect to the size of the backbone, message overhead and energy-efficiency.**

I. Introduction

Wireless sensor networks (WSNs) are self-configured and self-healing networks that do not require any fixed installation of infrastructure to operate. This flexibility is attractive for military, environmental, industrial and health care monitoring applications. However, these networks are known to suffer from limited energy resources. The key reason is mainly due to the fact that sensor nodes usually operate using a light-weight battery which determines the lifetime of the networks. Replacing or recharging the battery once the network has been deployed in a field may not be practical. Therefore, energy conservation remains as an important issue in the design of WSNs.

Topology control is a well-known strategy for addressing the energy constraints of WSNs [1][2]. It is beneficial for achieving network performances such as connectivity and coverage. There are several ways to exercise the topology control. One is using the *power control* technique [3] which adjusts the transmission power of nodes to minimize the energy spent on communication. Another technique is by using the *power mode* [4], which aims to put redundant nodes into a sleep mode periodically to save energy wasted for idle listening. Finally, it can select a set of nodes to form a backbone in the network using a Connected Dominating Set (CDS), which is derived from the concept of graph theory [5]. It has an advantage in organizing the network into a hierarchical structure, which cannot be achieved by *power mode* or *power control* techniques. The CDS offers scalability, simplifies the network management and reduces communication overheads.

A CDS has been used for broadcasting over the blind flooding [6] to minimize the number of redundant transmissions by restricting the transmission to the nodes in the CDS. In a network, a set of nodes is defined as a dominating set (DS) if all nodes are either in the set or adjacent to a node in the set. The dominating set that is connected forms a backbone and it is called a connected dominating set (CDS). Figure 1 gives an example of the backbone formed by nodes u, v, w, x, y and z.

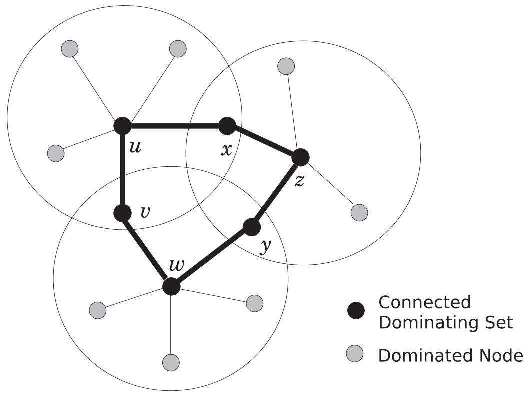


Fig. 1. A backbone in the network built using a connected dominating set.

An efficient algorithm should generate a small size CDS so that the overhead involved in maintaining the backbone is minimized. Therefore, many existing CDS algorithms [7][8][9][10] were proposed to minimize the CDS size. But, these algorithms fail to consider the remaining energy capacity of nodes during the selection of the backbone. As a result, the lifetime of the CDS can be shortened if the chosen CDS nodes do not have sufficient energy to perform various tasks. The recent works such as [11], [12], [13] and [14] take into account this shortcoming by using the residual energy as a selection metric for deciding the CDS nodes. The common practice is to assign the role of the CDS to higher remaining energy nodes in the network. This allows the role of the CDS to be alternated among nodes in the network once the energy drops to a certain level or fully depletes. However, these techniques

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

either require high message overheads to build the CDS or produce a large CDS size.

Our work, on the other hand, aims to construct an efficient technique that not only generates a small CDS size but also achieves a good performance in terms of producing low message complexity and minimizing the energy consumption when constructing the CDS. The great features of the TPSI algorithm are: (1) operates in a distributed fashion and requires only localized information gathered using beacons to support dynamic nature of networks, (2) guarantees the network connectivity as long as nodes are in the CDS or have a neighbor in the CDS, (3) achieves an energy-efficient CDS due to a low number of exchanged messages for CDS construction and the consideration of the nodes' remaining energy for CDS election and (4) minimizes the size of the CDS by eliminating redundant nodes in the CDS.

The rest of this paper is organized as follows. Section II presents definitions used for the TPSI algorithm. Section III discusses the detailed implementation of our algorithm. Section IV provides the theoretical performance of the algorithm including the time complexity and message complexity. Section V describes the simulation results. Finally, Section VI concludes our work and discusses our future work.

II. Preliminaries

An undirected graph $G = (V, E)$ is used to represent a WSN, where V is a set of sensor nodes in the network, called vertices and E is the edge denoted as $(u, v) \in E$ that describes a set of a communication link between a pair of sensor nodes. 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.

We assume that the network consists of at least three vertices and remains connected. Each node is able to estimate the distances to each neighbor. The distance information can be provided via a global positioning system (GPS) or a localization algorithm [15].

In the graph G, 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. In our work, we first find a DS in the network and then use connectors to connect the DS to form the CDS.

III. TPSI Algorithm Description

The three-phase single initiator (TPSI) algorithm first constructs a maximal independent set (MIS) of a graph and then joins the MIS with nonMIS nodes to generate the CDS. The first phase is the generation of a MIS, in which the MIS is a dominating set (DS). The second phase is the generation of a CDS, in which connectors are selected to connect the MIS nodes. The third phase is the pruning process of redundant CDS nodes to further reduce the size of the CDS.

A. Selection Criterion

TPSI algorithm uses a *key* as a selection metric for CDS nodes.

We define the *key* as follow:

Definition 1 (Key). *Node u with key(u) has a higher priority than node v with key(v) if key(u) is larger than key(v). In case of a tie, the node with the larger identifier is used to break the tie.*

The *key* of node u is computed using

$$key(u) = fairness(u) + nodeDegree(u), \tag{1}$$

where *nodeDegree*(u) is the number of neighbors of node u. The node degree is used to reduce the number of CDS nodes, thus reducing the number of nodes forming the backbone. This metric ensures that nodes with the largest number of neighbors, which can cover more nodes are chosen as a CDS, hence creating a small CDS.

The *fairness* [16] is determined by

$$fairness(u) = \left(\frac{dn}{d^n + n - 1}\right)\left(\frac{E_r}{E_i}\right), \tag{2}$$

where d is the distance between a pair of nodes, E_r is the residual energy, E_i is the initial energy and n is the path loss exponent, in our case n is set to 3. The *fairness* metric takes into account the energy consumption involved during multihop communication as well as the energy reserves of nodes when choosing eligible nodes for the CDS. This allows nodes to deplete their energy reserves in a fair and uniform manner and rotate the role of the CDS. In order to prolong the network lifetime, nodes with high energy reserves should be selected as the backbone.

B. MIS Generation Phase

The MIS generation begins with the initiator election. We use a simple method for the initiator election by selecting the initiator with the smallest identifier 0. Alternatively, a leader election algorithm in [17] can be used to find the initiator.

All nodes initially stay in uncovered state and at the end of the CDS construction, they will subsequently become either a dominatee or CDS. During the CDS construction, the nodes can transit into any of the four states, which is illustrated in Figure 2.

At the beginning, all nodes stay in an uncovered state. Each node broadcasts a beacon to discover its symmetrical neighbors. The MIS construction starts with node 0 volunteering to be an initiator. It changes its state to dominator and broadcasts a *Dominator Message*. Upon receiving the *Dominator Message*, an uncovered node changes its state to a dominatee. This dominatee then selects the uncovered neighbor with the largest *key* to be a dominator. The dominator election is broadcast to its neighbor by sending a *Dominatee Message*. An uncovered node receiving the *Dominatee Message* will first check whether it has been nominated as a dominator. If it is chosen, it changes its state to a dominator and broadcasts a *Dominator Message*. Otherwise, it will remain in the uncovered state. The process is repeated until all uncovered nodes change their state

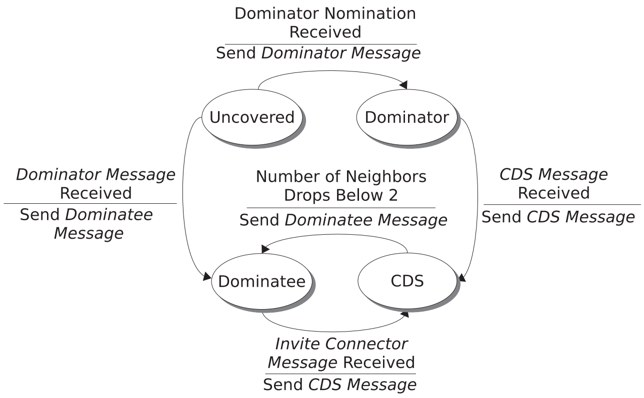


Fig. 2. The state transition diagram of a node in the TPSI algorithm.

to either a dominatee or a dominator. The decision to terminate the MIS generation is made locally when a node changes its state to dominatee or dominator and no longer has uncovered nodes.

C. CDS Generation Phase

The CDS generation is interested in finding connectors to join the DS constructed by the first phase. The DS and connectors both form the CDS in the network. At the completion of the CDS process, all nodes will remain either in a CDS state or a dominatee state. The CDS generation begins as soon as the MIS generation process is completed. Due to the order of the message propagation, the leader will be the first to complete the MIS generation and to begin the CDS generation.

A node changes its state to CDS and broadcasts a *CDS Message*. A dominatee receiving the *CDS Message* sends a *Volunteer Connector Message* to its neighbors and waits for an invitation to become a CDS. A dominator receiving the *Volunteer Connector Message* will set its timer T_{dom} [18] to wait for the arrival of *Volunteer Connector Message* from surrounding neighbors. The timer is computed by

$$T_{\text{dom}} = T_{\max}\left(\frac{1}{\textit{nodeDegree(u)}} \times \frac{1}{\textit{E(u)}_r}\right), \qquad (3)$$

where $T_{\max}$ is a pre-defined time value, *nodeDegree(u)* is the number of direct neighbors of node u and $\textit{E(u)}_r$ is the residual energy of node u.

Equation 3 allows the qualified dominator to be the first to find a connector, thus supresses the chances of the less qualified dominator to nominate a connector. The selection criteria for the qualified dominator are based on the residual energy and node degree. Based on Equation 3, it is apparent that the dominator with the largest residual energy and node degree has a shorter delay than its counterpart. Therefore, it is the first to send an invitation to a potential connector.

During the time-out, if the dominator receives a *CDS Message*, it will change its state to a CDS and broadcasts a *CDS Message*. This indicates that a connector has been chosen by other dominators. Therefore, it has to be a CDS to connect to this connector. Otherwise, when the timer is expired, it computes the *key* of each dominatee neighbor using the Equation 1 and nominates the node with the largest *key* to be a connector. The connector nomination is then broadcast using an *Invite Connector Message*.

Upon the receipt of an *Invite Connector Message*, the dominatee first checks whether it has been nominated as a CDS. If it is nominated, it changes its state to a CDS and broadcasts a *CDS Message*. Otherwise, it stays in the dominatee state. The process is repeated until all nodes change their state to either a CDS or a dominatee. The decision to terminate the CDS generation is also made using localized information. The decision to terminate the MIS generation is made when all nodes are either in CDS or dominatee state.

D. CDS Pruning Phase

The CDS formed at the completion of the second phase is not a minimum dominating set. Since, computing the minimum dominating set is a well known NP-hard process [19], the pruning aims to reduce the CDS size as small as possible. After the completion of the second phase, it is observed that some nodes in the CDS are already covered by at least a CDS. Therefore, they are redundant and can be removed from the CDS to minimize the size of the CDS.

To describe the CDS pruning process, the term "pendant node" is introduced in Definition 2.

Definition 2 (Pendant Node). *A pendant node is a node that has exactly one neighbor in the network.*

The pruning of the CDS is made according to Rule 1.

Rule 1. *A pendant node u is eliminated from a CDS if its state is CDS and has a CDS among its 1-hop neighbors.*

Once pruned from the CDS, the pendant node changes its state to dominatee and broadcasts a beacon to inform the change of its state.

IV. Performance Evaluation

Theorem 1. *TPSI has $O(n)$ total time complexity and $O(n)$ total message complexity, where n is the total number of nodes in the network.*

Proof. The overall time complexity of the TPSI algorithm is the time consumed for constructing the MIS and CDS and the time spent on pruning the generated CDS. During the MIS generation, each dominatee needs $O(\triangle)$ time to find the potential dominator for the MIS, where $\triangle$ is the node degree. Therefore, this process has a constant time. During the CDS generation, the time complexity is measured by the time taken for finding connectors to join the MIS. Each CDS node waits a constant time, $O(\triangle)$ to build its connector set. The time involved in the CDS pruning is also $O(\triangle)$ because each pendant node has a constant number of one-hop neighbors to prune. Due to the constant time involved in the three phases, the algorithm has a total time complexity of $O(n)$.

In this algorithm, each node broadcasts a constant number of messages. During the MIS generation, each node sends at most one message of either a *Dominator Message* or a *Dominatee Message*. During the CDS generation, the worst case occurs when a dominatee, which is nominated as a connector sends two messages; *Volunteer Connector Message* and *CDS Message*. Since each node sends a bounded number of messages, its total message complexity is $O(n)$.

Lemma 1. *The size of a MIS generated by the first phase is at most* $3.8 \cdot opt + 1.2$*, where opt is the size of a minimum CDS (MCDS) as deduced in [20].*

Proof. Wu et. al [20] proved that each node is adjacent to at most $3.8 \cdot opt + 1.2$ nodes in the MIS. Therefore, the size of the MIS generated by the algorithm during the first phase is also bounded by $3.8 \cdot opt + 1.2$.

Lemma 2. *The size of the connector set C found during the second phase of the algorithm is bounded by* $3.8 \cdot opt + 0.2$*.*

Proof. The TPSI algorithm chooses the connectors among the dominatees formed by the first phase. Let *M* denote the MIS size and *C* is the size of a connector set. Recall that to become a connector, each dominatee must receive two messages: a *CDS Message* from a node in *M* and also an *Invite Connector Message* from another node in *M*. Therefore, $C \leq M-1$. From Lemma 1, $C \leq 3.8 \cdot opt + 1.2 - 1 \leq 3.8 \cdot opt + 0.2$.

Theorem 2. *TPSI algorithm has an approximation factor of at most* $7.6 \cdot opt + 1.2$*.*

Proof. The approximation factor of the TPSI algorithm is bounded by the size of the MIS *M* and the size of connectors *C* found during the first phase and second phase respectively. Based on Lemmas 1 and 2, the CDS size generated by the algorithm is given by $\leq M + C \leq 7.6 \cdot opt + 1.2$.

V. Simulations and Discussions

In this section, 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 TPSI algorithm with the ECDS [14] and PACDS [13] algorithms. We selected the ECDS for a comparison due to its ability to generate a small CDS. Although PACDS does not generate smallest CDS size, it builds a CDS quickly as shown in [21].

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). This is important to understand the behavior of the algorithms as the number of neighbors in the network rises. It indicates how well the proposed algorithm performs when it is subjected to network scalability.

The simulations were conducted using the discrete event simulator OMNeT++ (version 4.1) [22]. We used the MiXiM framework [23] to incorporate the detailed modeling of physical layer effects such as path loss and signal interference to ensure our algorithm works in practice. For the simulations, we generated realistic topologies that were extended from a small network calibrated using the testbed in our department. Three types of topologies were created to investigate the performance of the algorithms under various network density (average number of neighbors per node). The generated topologies were sparse networks (network density below 6), medium networks (network density 6-10) and dense networks (network density over 10). The network density values chosen for the networks were based on the values described in [24]. 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 was not uniform.

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 sparse network. In comparison with the ECDS and PACDS, the TPSI consistently generates the lowest number of CDS with respect to the network size. The same results were observed in medium and dense networks.

Figure 3(b) illustrates the size of the CDS generated by all algorithms as nodes have more neighbors. We can see that our algorithm had the best performance over the ECDS and PACDS algorithms under sparse and medium networks, whereas under dense network, the ECDS created the smallest CDS size. Although the ECDS claims that it is efficient in generating the smallest CDS size, the finding shown in Figure 3(b) proves that this claim is true only for dense networks.

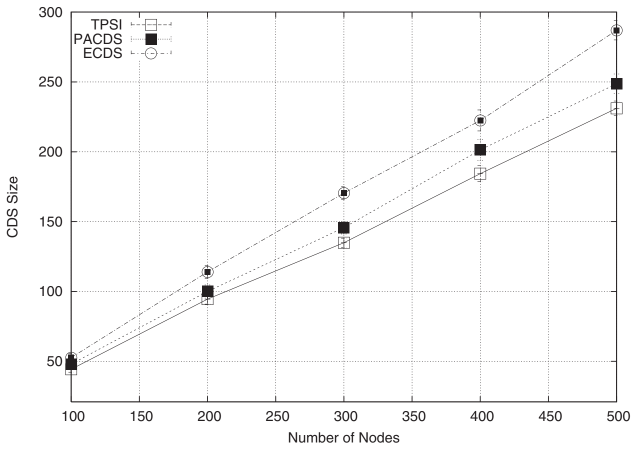
The PACDS algorithm on the other hand is not efficient in reducing the CDS size. Its CDS size is 36% higher than the size of the TPSI. This result is expected because PACDS uses a simple but less efficient rule for finding a CDS, therefore a self-pruning is required for reducing the CDS size.

The efficiency of the TPSI in reducing the CDS size is contributed to two reasons. One is the use of a node degree for the CDS selection. Since nodes with more neighbors are chosen as a CDS, more nodes can be covered by these nodes, thus less nodes are required for building the CDS. The other is due to the CDS elimination process that successfully removes the redundant CDS. This effect is more apparent in dense networks.

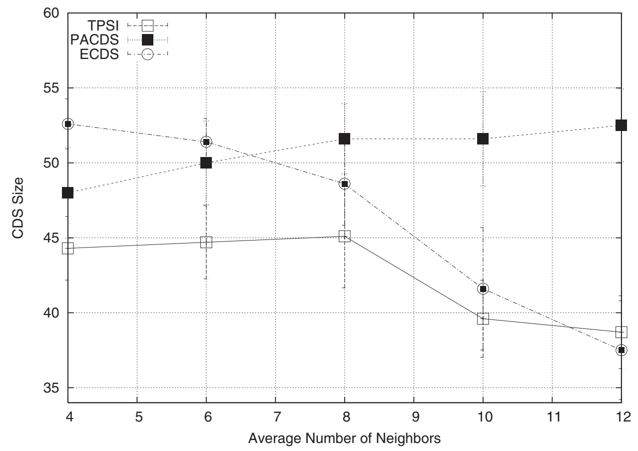
B. Message Overhead

We analyze the message overhead based on the number of messages sent by nodes during the CDS construction. The TPSI requires a lower number of messages for constructing a CDS compared to the ECDS and PACDS as shown in Figure 4(a) and Figure 4(b). It is interesting to see that the message overhead of the TPSI in Figure 4(b) remains almost constant although the node degree increases. However, the message overhead of the PACDS and ECDS has a linear relationship with the node degree. Hence, as the network density increases, more messages are required. This is obvious as in the case of node degree 12, in which the message overhead of the ECDS is 91% and PACDS is 92% higher than the one in the TPSI.

The efficiency of the TPSI algorithm is due to a bounded number of exchanged messages used for generating the MIS

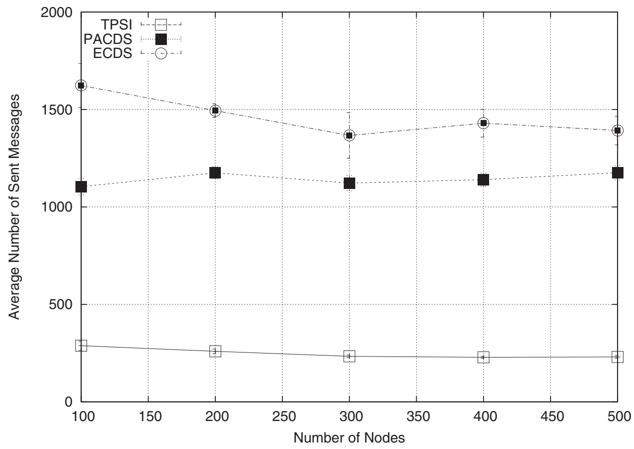


(a) The CDS size versus network size for sparse network.

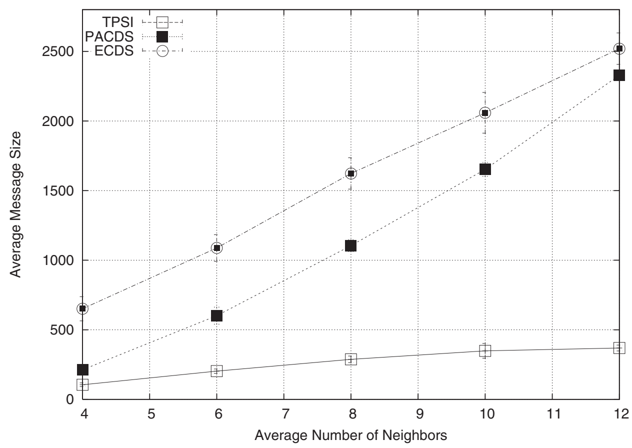


(b) The CDS size versus network density for 100 nodes network size.

Fig. 3. CDS size comparison with respect to network size and network density.



(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 network density.

and CDS. Unlike in ECDS, nodes do not compete and broadcast messages to find out whether they have been nominated as MIS or connector nodes. Instead, they only need to perform local check for the nomination of a connector or MIS node and send messages only if their state changes upon the nomination. In the case of the PACDS, the high message overhead is caused by the self-pruning process. A larger number of messages is needed when the network becomes denser due to the increase in the number of neighbors to be pruned.

C. Energy Consumption

In this section, we study the amount of energy involved for constructing the CDS by measuring the energy dissipation of sending and receiving packets activities as in [25]. Every time a node sends a packet, its transmit energy $E_{\mathrm{TX}}(k, d)$ drops by

$$E_{\mathrm{TX}}(k, d) = \delta k + \gamma k d^2, \tag{4}$$

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

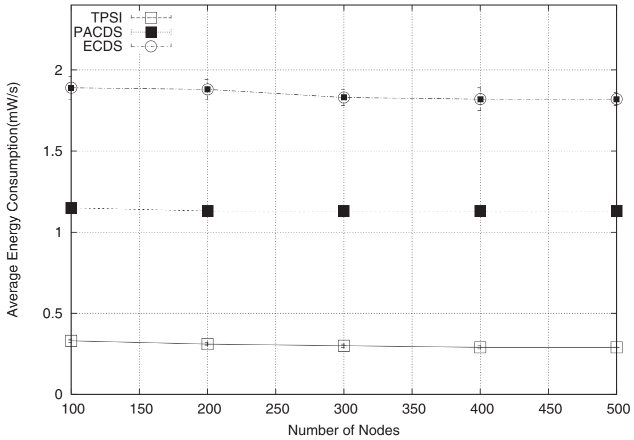
$$E_{\mathrm{RX}}(k) = \delta k. \tag{5}$$

The results of Figures 5(a) and 5(b) show that TPSI 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 TPSI algorithm is scalable and can be used for large network deployment. The ECDS and PACDS algorithms however, consume a significant amount of energy and their energy increases linearly with the number of neighbors.

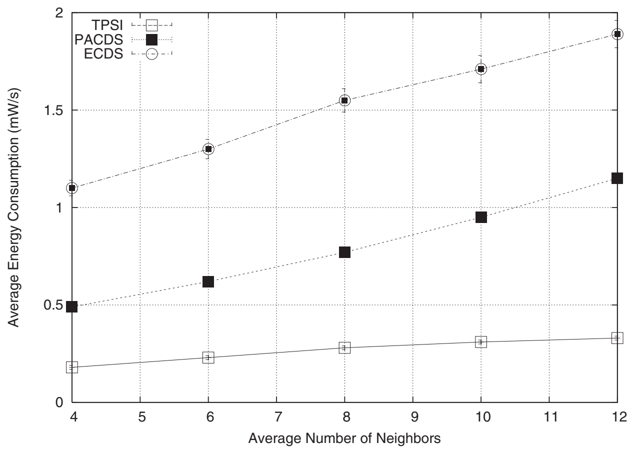
VI. Conclusion

This paper presents the three-phase single initiator (TPSI) algorithm, which can generate a small-size CDS using a minimum number of message overhead and consume low energy consumption . The performance results showed that our algorithm is scalable for WSNs and it outperforms the ECDS and PACDS algorithms.

In our future work, the issue of the CDS reconstruction will be studied. We will investigate various parameters that



(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 network density.

may trigger the reconstruction of the CDS and evaluate the efficiency of the TPSI during the CDS maintenance.

REFERENCES

[1] P. Santi, "Topology control in wireless ad hoc and sensor networks," *ACM Computing Surveys (CSUR)*, vol. 37, no. 2, pp. 164–194, 2005.

[2] R. Rajaraman, "Topology control and routing in ad hoc networks: a survey," *ACM SIGACT News*, vol. 33, no. 2, pp. 60–73, 2002.

[3] Y. Liu, Q. Zhang, and L. Ni, "Opportunity-based topology control in wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 21, no. 3, pp. 405–416, 2010.

[4] A. Cerpa and D. Estrin, "Ascent: Adaptive self-configuring sensor networks topologies," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 272–285, 2004.

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

[6] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath, "Flooding for reliable multicast in multi-hop ad hoc networks," in *Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*. ACM, 1999, pp. 64–71.

[7] S. Butenko, X. Cheng, D. Z. Du, and P. M. Pardalos, "On the construction of virtual backbone for ad hoc wireless networks," *Cooperative Control: Models, Applications and Algorithms*, vol. 1, pp. 3–43, 2003.

[8] B. Das and V. Bharghavan, "Routing in ad-hoc networks using minimum connected dominating sets," in *IEEE International Conference on Communications (ICC' 97)*, June 1997.

[9] J. Wu, W. Lou, and F. Dai, "Extended multipoint relays to determine connected dominating sets in MANETs," *IEEE Transactions on Computers*, pp. 334–347, 2006.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. Moses, and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, 2005.

[16] W. Dargie, R. Mochaourab, A. Schill, and L. Guan, "A topology control protocol based on eligibility and efficiency metrics," *Journal of Systems and Software*, vol. 84, no. 1, pp. 2–11, 2011.

[17] B. Awerbuch, "Optimal distributed algorithms for minimum weight spanning tree, counting, leader election, and related problems," in *Proceedings of the nineteenth annual ACM symposium on Theory of computing*. ACM, 1987, pp. 230–240.

[18] B. Kim, J. Yang, D. Zhou, and M. Sun, "Energy-aware connected dominating set construction in mobile ad hoc networks," in *14th International Conference on Computer Communications and Networks, 2005. ICCCN 2005. Proceedings.* IEEE, 2005, pp. 229–234.

[19] M. R. Garey and D. S. Johnson, "Computers and intractability," *A Guide to the Theory of NP-Completeness*, 1979.

[20] W. Wu, H. Du, X. Jia, Y. Li, and S. C. H. Huang, "Minimum connected dominating sets and maximal independent sets in unit disk graphs," *Theoretical Computer Science*, vol. 352, no. 1-3, pp. 1–7, 2006.

[21] B. Yin, H. Shi, and Y. Shang, "An efficient algorithm for constructing a connected dominating set in mobile¡ i¿ ad hoc¡/i¿ networks," *Journal of Parallel and Distributed Computing*, vol. 71, no. 1, pp. 27–39, 2011.

[22] "Omnet++ discrete event simulator," http://www.omnetpp.org/.

[23] "MiXiM simulator for wireless and mobile simulations," http://mixim.sourceforge.net/.

[24] I. Stojmenovic, "Simulations in wireless sensor and ad hoc networks: matching and advancing models, metrics, and solutions," *Communications Magazine, IEEE*, vol. 46, no. 12, pp. 102–107, 2008.

[25] 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.