

## Research Article

# Energy Efficient Low-Cost Virtual Backbone Construction for Optimal Routing in Wireless Sensor Networks

**K. Mohaideen Pitchai and B. Paramasivan**

*Department of Computer Science and Engineering, National Engineering College, Kovilpatti, Tamil Nadu 628503, India*

Correspondence should be addressed to K. Mohaideen Pitchai; kmopi786@yahoo.com

Received 27 December 2013; Revised 11 June 2014; Accepted 17 June 2014; Published 7 July 2014

Academic Editor: Fuzhong Nian

Copyright © 2014 K. M. Pitchai and B. Paramasivan. 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.

Many prominent applications in wireless sensor networks which require collected information have to be routed to end nodes in an efficient manner. In general, weighted connected dominating Sets (WCDS) based routing is a promising approach for enhancing the routing efficiency in sensor networks. Backbone has been used extensively in routing. Here an efficient WCDS algorithm for constructing a virtual backbone with low total cost, hop spanning ratio, and minimum number of dominators is proposed. We report a systematic approach, which has three phases. Initial phase considers the issues of revoking a partial CDS tree from a complete CDS tree. Secondary and final phases make the design of the complete algorithm by considering the determination of dominators using an iteration process. Our findings reveal better performance than the existing algorithms in terms of total cost, hop spanning ratio, and number of dominators.

## 1. Introduction

Researchers and practicing scientists are finding new avenues for using wireless sensor network (WSN) in varieties of application areas that include emergency response, disaster relief, monitoring, surveillance, data collection, and community wireless networking. Sensor networks consist of one or more base stations and a large number of inexpensive nodes, which combine sensors and low power wireless radios. Due to limited radio range and battery power, most nodes cannot communicate directly with a base station but rather rely on their peers to forward messages to and from base stations. The task of constructing stable and efficient routing algorithms for WSN represents greater challenge compared to routing in networks based on a fixed and wired infrastructure. Since there is no fixed infrastructure or centralized management in WSN, a low-cost virtual backbone has been proposed as the routing infrastructure [1, 2]. The virtual backbone should be designed in such a way that the nodes not present in the skeleton are within the transmission range of some nodes in the skeleton.

Routing based on a connected dominating set (CDS) is a frequently used approach, where the searching space for a

route is reduced to nodes in the set. Using only a subset of all nodes for the routing tasks could lead to a more efficient use of resources. The nodes in the subset act as gateways used for routing by the rest of the nodes not in this subset. Therefore it is essential to distinguish between gateway and nongateway nodes. For routing, it is beneficial to construct a small size backbone network that tends to reduce overhead for access time and update time. Construction of the CDS is a well-known problem in graph theory. Unfortunately, the dominating set problem has been proved to be NP-hard [3]. Nodes within a wireless network communicate over a shared, scarce wireless channel. Compared with cables in wired networks, wireless links have much less available bandwidth. Each node has an omnidirectional antenna and messages are transmitted in all directions. The transmission can be heard by all the other nodes within the transmission range. Since the nodes in a CDS may have heavy load working as the central management agents, minimizing the size of the CDS can greatly help with reducing transmission interference and the number of control messages. If the receiver is not within the transmission range of the sender, they need to communicate through multihop links by using some intermediate nodes

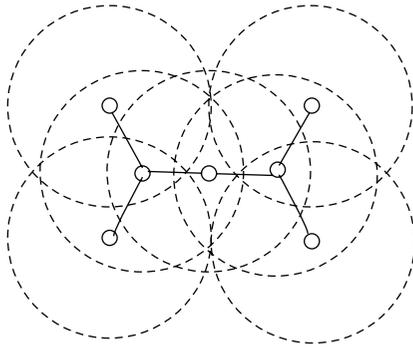


FIGURE 1: Illustrating UDG.

to relay the messages. We call this multihop routing. This characteristic of wireless networks induces us to take into account another factor of a CDS, the hop spanning ratio, which is the shortest path between any pair of nodes in the CDS. With the help of a CDS with small size and spanning ratio, routing is easier and can adapt quickly to topology changes of a network.

In this paper a weighted connected dominating set (WCDS) has been proposed to serve as a virtual backbone. WCDS based routing is a promising approach for enhancing the routing efficiency in WSN [4]. The WCDS is very suitable for cluster formation. To keep the virtual backbone as small as possible, our idea is to generate a WCDS algorithm, which can resolve the issue of minimizing the total cost of the backbone, hop spanning ratio, and number of dominators. Our findings reveal better performance than the existing algorithms in terms of WCDS total cost, average hop spanning ratio, and number of dominators. This paper is organized as follows. Section 2 overviews the literature review and preliminaries of related works. Section 3 presents the proposed algorithm on generating low-cost virtual backbone. Performance evaluation is done in Section 4 where the proposed algorithm is compared with existing algorithms. We summarize our major results in Section 5. Finally in Section 6 we conclude this paper and discuss future work.

## 2. Literature Review

We can use an unweighted graph  $G = (V, E)$  to represent a WSN, where  $V$  represents a set of wireless sensor nodes and  $E$  represents a set of edges. An edge between host pairs  $\{u, v\}$  indicates that both hosts  $v$  and  $u$  are within their wireless transmitter ranges. Thus the corresponding graph will be an undirected graph. If two hosts that want to communicate are outside their wireless transmission ranges they could communicate only if other hosts between them are willing to forward packets for them. A subset of nodes  $R \subseteq V$  is called dominating set if any vertex in  $V$  is either in  $R$  or adjacent to a node in  $R$ . A connected dominating set (CDS) is a dominating set whose induced graph is connected. All known algorithms for constructing CDS consider the number of nodes as the sole criteria. A CDS with minimum cardinality is the minimum connected dominating set (MCDS). In wireless

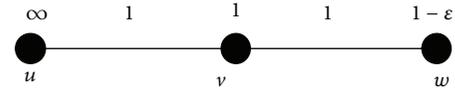


FIGURE 2: Example for Greedy algorithm 1.

sensor networks each node  $u$  has a cost  $c(u)$ . Then, a CDS  $C$  is called a weighted connected dominating set (WCDS). A subset  $C$  of  $V$  is a minimum weighted connected dominating set (MWCDS) if  $C$  is a WCDS with minimum total cost. The smaller the size of a dominating set is, the better the quality is. Minimizing the size of the computed WCDS can help to decrease the control overhead and topology update is restricted to a small subset of nodes.

Many works seek a minimum weighted connected dominating set in unit disk graph (UDG) as their major design goal. The communication range of a node in a wireless sensor network is typically modeled as a disk centered at the node with radius equal to the transmission range of the radio. When transmission range is fixed for all nodes, the network has the property of a UDG, where an edge exists if and only if two nodes have internodal distance less than or equal to 1 unit [5]. Figure 1 shows a UDG.

Most of the methods proposed in the literature aim to find a small dominating set for homogeneous networks. The idea of using a CDS as a virtual backbone for routing was proposed in [6]. Since then many efforts have been made to design approximations or heuristics for WCDS. Many of them are based on classical greedy algorithms. Guha and Khuller (GK) first proposed a two-stage greedy  $(\ln \Delta + 3)$  approximation [7] for MCDS in general graphs where  $\Delta$  is the maximum node degree in the graph, as well as a lower bound  $(\ln \Delta + 1)$  for any polynomial-time approximation. The first algorithm (Greedy 1) to generate a dominating set is generating a maximal independent set as follows. First, all nodes are originally marked as WHITE, which represents that the node is not assigned any role yet. Node  $u$  sends a message `IamDominator` to all its one-hop neighbors if it has the smallest cost (ID is often used if every node has a unit cost) among all its WHITE neighbors. Node also marks itself Dominator. When a node  $v$  receives message `IamDominator` from its one-hop neighbors, node then marks itself Dominatee. Node  $v$  then sends a message `IamDominatee` to all its one-hop neighbors. Clearly, the nodes marked with Dominator indeed form a dominating set. Figure 2 shows an illustration of the Greedy 1 algorithm. Assume that three wireless nodes  $u$ ,  $v$ , and  $w$  are distributed along a line with one-unit intervals. The nodes costs of  $u$ ,  $v$ , and  $w$  are  $\infty$ , 1, and  $1 - \epsilon$ , respectively. The dominators selected by the first method are nodes  $w$  and  $u$  and the total cost of the solution is  $\infty$ . However, the optimal solution is formed by  $v$  with a total cost 1.

The second algorithm (Greedy 2) of constructing a dominating set is based on the minimum weighted set cover [8]. The method can be described in a centralized way as follows. In each round, select an unselected node  $i$  with minimum ratio  $c(i)/d_i$ , where  $d_i$  is the number of nodes not covered by previously selected dominators. It is well known that this

centralized method produces a dominating set whose total cost is no more than  $\log(\Delta + 1)$  times optimum, where  $\Delta$  is the maximum original degree of all nodes.

Das et al. [9] propose the distributed implementations of the two greedy algorithms given by Guha and Khuller. The first algorithm grows a CDS from one node with maximum degree. Thus the first step involves selecting the node with the highest degree. Therefore a node must know the degree of all nodes in the graph. On the other hand, each iterative step selects either one or two edged paths emanating from the current CDS; thus the nodes in the CDS must know their one and two hop unmarked neighbors. These two requirements force the flooding of degree information in the network. This algorithm generates a CDS with approximation ratio of  $2H(\Delta)$  in  $O(|C|(\Delta + |C|))$  time, using the  $O(n|C|)$  messages, where  $C$  represents the generated CDS.

A few distributed algorithms for constructing CDS have been reported. Most distributed algorithms for CDS are theoretically interesting but are difficult for real world implementation. Some distributed algorithms for CDS are, in fact, distributed implementations of the variations of centralized algorithms and have high message complexity overhead [10]. In recent years, a few interesting localized algorithms for CDS have been proposed.

Wu and Li [11] proposed a completely localized algorithm to construct CDS in general graphs. Initially all vertices are unmarked. They exchange their open neighborhood information with their one-hop neighbors. Therefore each node knows all of its two-hop neighbors. The marking process uses the following simple rule: any vertex having two unconnected neighbors is marked as a dominator. The set of marked vertices form a connected dominating set, with a lot of redundant nodes. Two pruning principles are provided to postprocess the dominating set, based on the neighborhood subset coverage. A node  $u$  can be taken out from  $S$ , the CDS, if there exists a node  $v$  with higher ID such that the closed neighbor set of  $u$  is a subset of the closed neighbor set of  $v$ . For the same reason, a node  $u$  will be deleted from  $S$  when two of its connected neighbors in  $S$  with higher IDs can cover all of  $u$ 's neighbors. This pruning idea is generated to the following general rule: a node  $u$  can be removed from  $S$  if there exist  $k$  connected neighbors with higher IDs in  $S$  that can cover all  $u$ 's neighbors.

The ultimate goal of most protocols is still to minimize the number of cluster heads (or the size of the backbone) not the total weight of cluster heads (or the backbone). For example, the stability or mobility of each node is considered as the weight. They preferred the node with high stability and low mobility to be the cluster head. In [12], the authors also combined the stability with the degree of each node as the weight. The higher priority is given to relatively stable and high degree nodes. Weight metric takes into account several system parameters like the node degree, transmission power, mobility, and battery power of the nodes. Most of these proposed weighted clustering algorithms applied simple greedy algorithms where the nodes with highest priority (lowest cost) become cluster heads.

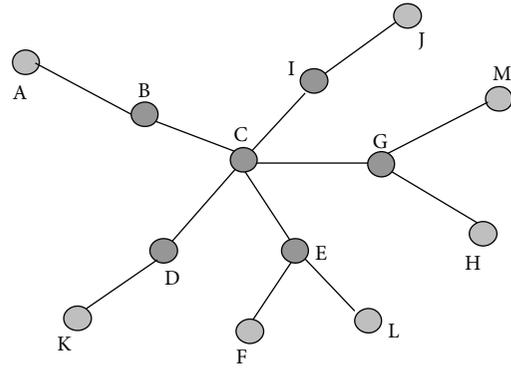


FIGURE 3: Typical WSN.

### 3. Backbone Construction Algorithm

The proposed algorithm WCDS has a systematic approach, consisting of three phases. The first phase constructs a CDS tree from a spanning tree. The second and final phases perform the elimination of dominant-act nodes using an iteration process. The dominating set constructed with this algorithm is more reliable and provides load-balanced routing. Even though this algorithm is slightly more complex than the existing algorithms in the context of the size of the dominating set generated, it is faster. The input given to the algorithm is a unit disc graph representing the wireless sensor network. The output from this algorithm is a set of nodes to be connected. To reduce the size of the CDS, it is observed that what is more related to the size of the CDS is the node degree. In this regard, the final phase of the algorithm still removes some nodes in the CDS, because they act as dominant nodes though originally they were regular nodes.

*Step 1. CDS tree Construction:* (i) let  $v \in V$  be the largest degree node. Include  $v$  to empty tree  $T$ ; (ii) for all nodes  $w$  adjacent to  $v$ , choose largest degree node. Add edge  $(v, w)$  to  $T$ ; (iii) if two or more nodes like  $w_1, w_2, \dots, w_n$  have the largest degree, add edge  $(v, w_1), (v, w_2), \dots, (v, w_n)$  to  $T$ ; (iv) repeat the above steps until all the nodes in the graph are present in the tree  $T$ ; (v) after constructing the partial CDS tree remove all leaf nodes present in the tree. Now the tree is called *CDS tree*.

Consider the following typical WSN UDG model as shown in Figure 3. It has 13 nodes and 12 edges. After applying the CDS tree construction steps the original tree will be made into a CDS tree, which is shown in Figure 4, where the node  $c$  has the largest degree. So it becomes the root node for the tree.

All other nodes adjacent to node  $c$  are the children of the root node  $c$  in the tree. The process is repeated until all the nodes in the network are present in the CDS tree. After construction of the spanning tree, the leaf nodes get removed. The leaf nodes are represented as gray nodes. The resultant tree is called the CDS tree.

After constructing the CDS tree, the CDS stack has to be computed according to the following steps.

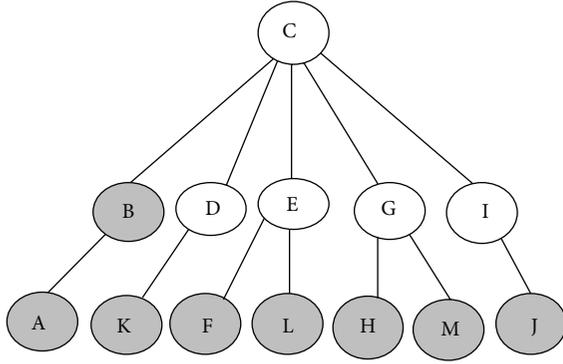


FIGURE 4: CDS tree.

*Step 2.* CDS stack Construction: (i) stack  $S = \text{null}$ ; (ii) based on the CDS tree construction, classify the nodes in the tree according to the level of the tree; (iii) Push the elements from lower level to upper level of the tree  $l_n, l_{n-1}, \dots, l_0$  into the stack  $S$ .

In the CDS tree node  $c$  is in level-0. Nodes B, D, E, G, and I are present in level-1. From level-0 to level-1 add the elements to the stack. Then the stack is called the CDS stack. It is shown in Table 1. In the CDS tree, some of the nodes are not truly dominant; they are regular nodes. To remove those nodes, use the criteria stated in the next step.

*Step 3.* Finding Dominators in CDS tree: after constructing the CDS tree [13], some nodes are not truly dominant; that is, they are regular nodes. To remove those nodes, we give some criteria,

- (i) while there is at least one element in  $S$ 
  - (1) Let  $w$  be the element with one unconnected dominant node and one regular node that has maximum degree.
  - (2) Remove the element  $w$  from the stack and delete that node from the tree, since the node is considered as regular node.

*Step 4.*  $R$  is set to nonleaf nodes of  $T$ . To further reduce the size of CDS, we observe that what is more related to the size of CDS is the node degree. According to our proposed algorithm Step 3 still removes some nodes in CDS, because they act as dominant nodes but originally they are the regular nodes. From Table 1, it is seen that the nodes B, D, E, G, and I have the maximum level. But nodes B, D, and I have the minimum degree. So nodes B, D, and I are not true dominant nodes. They act as dominating nodes. They can be considered only as regular nodes in the transmission. So the nodes B, D, and I got removed from the CDS tree as shown in Figure 5.

Hence the size of the CDS is reduced to three and as a result nodes C, D, and E were present in the CDS. These three nodes form the backbone of the entire network. This algorithm produces a set of nodes connected as a tree that form a CDS. The nodes in the CDS are called as dominators

TABLE 1: CDS stack.

Element	Level	Degree
C	0	5
B	1	2
D	1	2
E	1	3
I	1	2
G	1	3

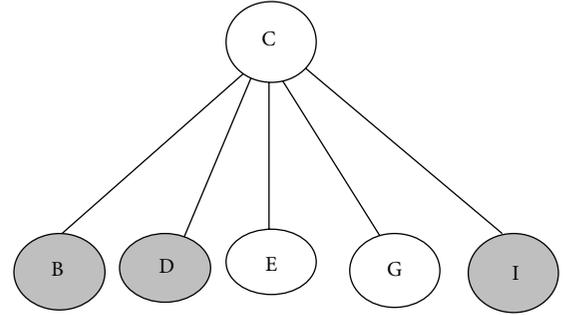


FIGURE 5: CDS tree after removal of dominant-act nodes.

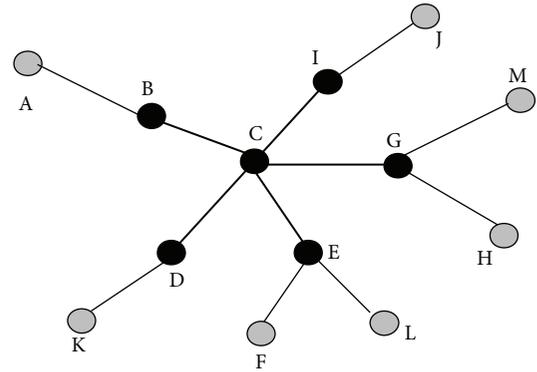


FIGURE 6: CDS for the network shown in Figure 3.

(dominant nodes). The CDS for the original network is shown in Figure 6.

**Lemma 1.** *The spanning tree  $T$  generated from our algorithm represents a connected graph  $R$ .*

*Proof.* Let  $G$  be a connected graph. Let  $T$  be a minimal connected spanning subgraph of  $G$ . Then for any line  $X$  of  $T$ ,  $T - X$  is disconnected and hence  $X$  is a bridge of  $T$ . Hence  $T$  is acyclic. Further  $T$  is connected and hence is a tree.  $\square$

## 4. Performance Evaluation

In this section, we compare our approaches with Greedy and WL algorithm. Our comparison was conducted through analytical study and simulation. Simulation for recalculation of connected dominating set in a dynamic network will be part of our future work.

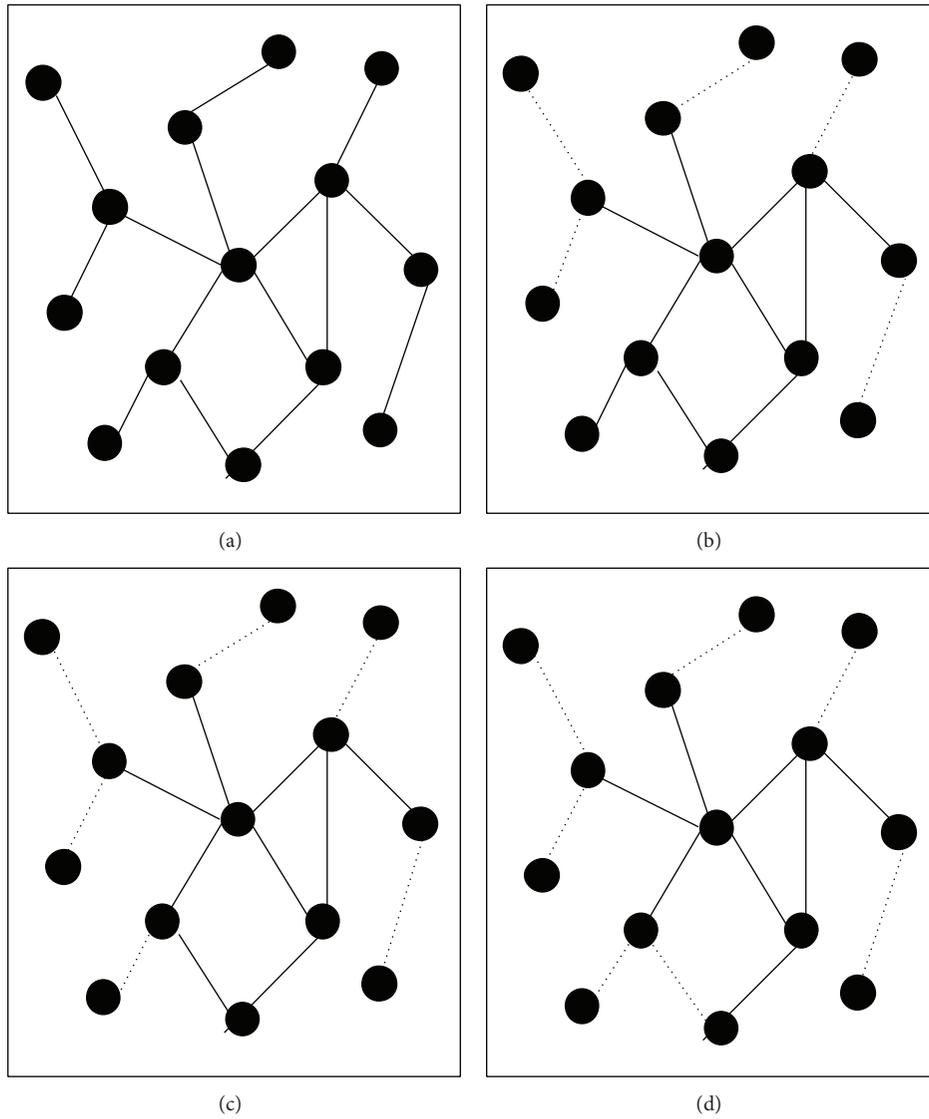


FIGURE 7

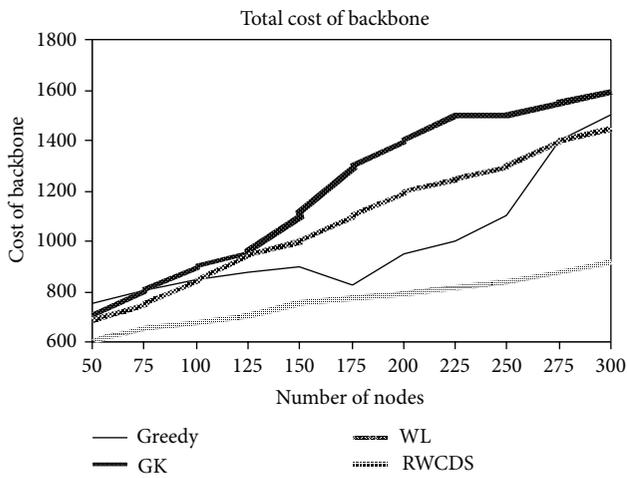


FIGURE 8: Performance of backbone (total cost).

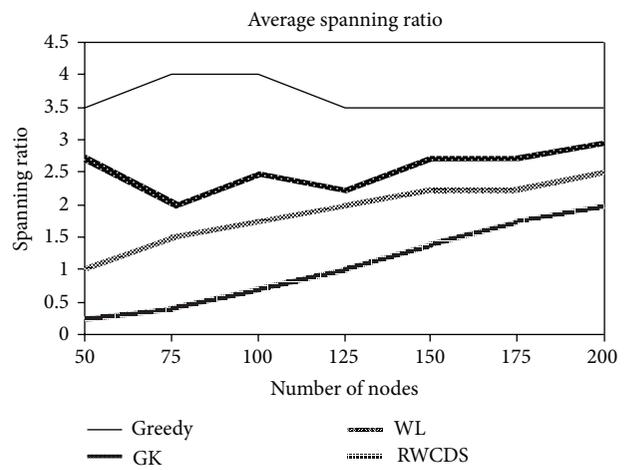


FIGURE 9: Performance of backbone (spanning ratio).

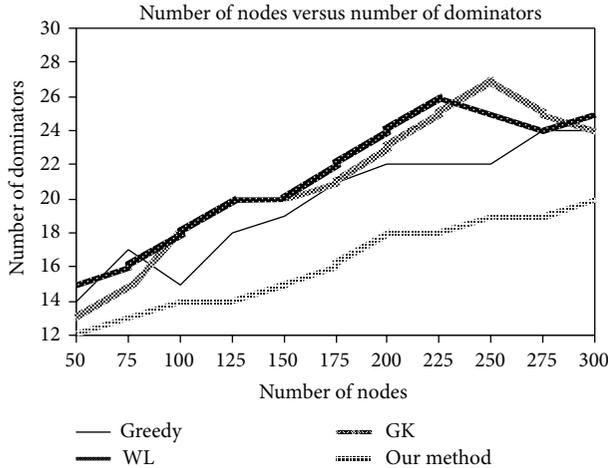


FIGURE 10: Number of nodes from different methods.

The generated WCDS in Greedy methods has high approximation ratio and high implementation complexities (in message and time). In addition it is not clear in their algorithm description how each individual node is informed when to start the marking process. The CDS maintenance is expensive too, as its approaches need to maintain a minimum spanning ratio. There is no performance analysis in Wu and Li's original paper, which incorrectly analyzes the algorithm's time complexity.

This algorithm needs at least two-hop neighborhood information. Clearly our approach is less complex than existing algorithm in all measurements, in particular the number of rounds needed. Note that the number of rounds is important metrics measuring the performance of the algorithm. Our proposed algorithm takes  $O(n^2)$  time, where  $n$  is the number of nodes in the network.

Figure 7(a) shows a graph representing a WSN. Figure 7(b) shows a dominating set produced by the Greedy algorithm Figure 7(c) shows a dominating set produced by WL algorithm for the same network. But this network admits a size and weight reduced CDS as shown in Figure 7(d).

The cost could represent the power consumption rate of this node if a backbone with small power consumption is needed, the robustness of this node if fault-tolerant backbone is needed, or a function of its security level if a secure backbone is needed. Therefore, by defining different costs, our proposed low-cost backbone formation algorithms can be used in various practical applications.

Let  $dG(u)$  be the degree of node  $u$  in a graph  $G$  and let  $\Delta$  be the maximum node degree of all wireless nodes (i.e.,  $\Delta = \max_{u \in V} dG(u)$ ). Each wireless node  $u$  has a cost  $c(u)$  of being in the backbone. Here, the cost  $c(u)$  could be the value computed based on a combination of its mobility, its node degree in the unit disk graph, and so on. In general, a smaller  $c(u)$  means that the node is more suitable for being in the backbone. Let  $\delta = \max_{ij \in E} \frac{c(i)}{c(j)}$ , where  $ij$  is the edge between nodes  $i$  and  $j$ ,  $E$  is the set of communication links in the wireless network  $G$ , and the maximum operation is taken on all pairs of adjacent nodes  $i$  and  $j$  in  $G$ . In

other words,  $\delta$  is the maximum ratio of costs of two adjacent nodes. The total cost of the constructed backbone is within  $\min(2\delta + 3\Delta)$  times the optimum for homogeneous networks when all nodes have the same transmission range. Here,  $\delta$  is the maximum ratio of costs of two adjacent wireless nodes and  $\Delta$  is the maximum node degree in the unit disk graph.

## 5. Experimental Results

The simulation platform was developed using JAVA programming language. In order to test the performance of the proposed algorithm we implemented it for experimental investigation. The algorithms considered for the experimental investigation are (i) Greedy algorithm, (ii) WL algorithm, (iii) GK algorithm, and (iv) RWCDs algorithm.

We randomly generated a set  $V$  of  $n$  wireless nodes with random costs drawn from the range 1 to 100 and the induced UDG( $V$ ) and, then, tested the connectivity of UDG( $V$ ). If it is connected, we construct different algorithms on UDG( $V$ ) to form dominating sets and measure the total costs of these dominating sets. Then, we apply our new method to construct the weighted backbone. We test the total cost of the final backbone and measure the average hop spanning ratio. In the experimental results presented here,  $n$  wireless nodes are randomly distributed in a 500 m  $\times$  500 m square, and the transmission range is set to 100 m. The relationship between number of nodes and total cost, average spanning ratio, and number of dominators is plotted and shown in Figures 8, 9, and 10, respectively.

We tested all algorithms by varying  $n$  from 50 to 300, where 50 vertex sets are generated for each case. Then we apply Greedy algorithm, GK algorithm, and WL algorithm to find the dominating set. Finally we determine the dominating set by our proposed algorithm. From the results one interesting observation is that the size of the backbone becomes stable when the network becomes denser. This may be due to the fact that dense network provides more candidates for backbone with potential lower costs.

## 6. Conclusion

We presented an overview of existing algorithms for constructing connected dominating sets. We presented an algorithm named reduced weight connected dominating set (RWCDs) to generate better virtual backbone for sensor networks. We also presented several experimental results by implementing the proposed algorithm in the JAVA programming language. Finally the results are consolidated effectively to compare various algorithms. These results show that the proposed techniques are effective in generating weighted connected dominating sets of reduced total cost, low spanning ratio, and minimum number of dominators.

Node mobility is not considered. In real situations, nodes may change their position with time. Nodes may also become inactive after a certain time and vice versa. The proposed algorithm does not generate a virtual backbone when the nodes transmission radius is large. At the current stage, our investigation does not include the dynamic maintenance of

the backbone, since the paper focuses only on determining a WCDS from a WSN. The future work will extend the proposed algorithm to the WSN in which each node has different transmission radius. We need to verify the effectiveness of our approach when the topology of the underlying networks changes due to node moving, node joining or leaving, or node failure while preserving the approximation quality.

## Conflict of Interests

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

## References

- [1] B. Das and V. Bharghavan, "Routing in ad-hoc networks using minimum connected dominating sets," in *Proceedings of the 1997 IEEE International Conference on Communications (ICC '97)*, vol. 1, pp. 376–380, June 1997.
- [2] B. Das, R. Sivakumar, and V. Bharghavan, "Routing in ad hoc networks using a spine," in *Proceedings of the 6th International Conference on Computer Communications and Networks (ICCCN '97)*, pp. 34–39, Las Vegas, Nev, USA, September 1997.
- [3] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman Publishers, 1979.
- [4] A. Schumacher, "Dominating set based Routing, Algorithms for ad-hoc networking," Seminar talk, 2003.
- [5] J. O'Rourke, *Computational Geometry in C*, Cambridge University Press, 2nd edition, 1998.
- [6] A. Ephremides, J. E. Wieselthier, and D. J. Baker, "Design concept for reliable mobile radio networks with frequency hopping signaling," *Proceedings of the IEEE*, vol. 75, no. 1, pp. 56–73, 1987.
- [7] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Journal of Algorithmica*, vol. 20, no. 4, pp. 374–387, 1998.
- [8] V. Chvatal, "A greedy heuristic for the set-covering problem," *Mathematics of Operations Research*, vol. 4, no. 3, pp. 233–235, 1979.
- [9] B. Das, R. Sivakumar, and V. Bharghavan, "Routing in ad hoc networks using a spine," in *Proceedings of the 6th International Conference on Computer Communications and Networks (ICCCN '97)*, pp. 34–39, Athens, Greece, September 1997.
- [10] I. Stojmenovic, *Handbook of Wireless Networks and Mobile Computing*, John Wiley & Sons, New York, NY, USA, 2002.
- [11] J. Wu and H. Li, "On calculating connected dominant 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 (DIALM '99)*, pp. 7–14, 1999.
- [12] U. C. Kozat, G. Kondylis, B. Ryu, and M. Marina, "Virtual dynamic backbone for mobile AdHoc networks," in *Proceedings of the IEEE International Conference on Communications*, vol. 1, pp. 250–255, 2001.
- [13] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, McGrawHill, 2nd edition, 2001.



# Hindawi

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

