

CONSTRUCTING MINIMUM CONNECTED DOMINATING SET IN MOBILE AD HOC NETWORKS

Mallikarjun Avula¹, Seong-Moo Yoo¹ and Seungjin Park²

¹Department of Electrical and Computer Engineering, University of Alabama in
Huntsville, Huntsville, AL, U.S.A

{ma0004@uah.edu, yoos@eng.uah.edu}

²Department of Management and Information Sciences, University of Southern Indiana,
Evansville, IN, U.S.A
park@usi.edu

ABSTRACT

One of the most important challenges of a Mobile Ad Hoc Network (MANET) is to ensure efficient routing among its nodes. A Connected Dominating Set (CDS) is a widely used concept by many protocols for broadcasting and routing in MANETs. Those existing protocols require significant message overhead in construction of CDS. In this paper, we propose a simple, inexpensive and novel algorithm of computing a minimum CDS. The proposed algorithm saves time and message overhead in forming a CDS while supporting node mobility efficiently. Simulation results show that the proposed algorithm is efficient in terms of both message complexity and the size of the CDS.

KEYWORDS

Connected dominating set, mobile ad hoc network, message overhead, node mobility

1. INTRODUCTION

A mobile ad hoc network (MANET) is defined as a collection of mobile hosts that dynamically form a wireless network without any backbone infrastructure and centralized administration. The mobile hosts in a MANET individually act as wireless network interfaces. The primary application of MANETs will be in situations where there is no fixed backbone infrastructure like battlefield scenarios, natural calamities such as earthquakes and hurricanes. MANET is considered to be adaptable and convenient because it consists of hosts which are of heterogeneous nature.

MANET operates on the concept of flooding where each host, after receiving a message, broadcasts it to the entire network. This could result in waste of valuable resources such as network bandwidth and battery power of the devices. One of the greatest challenges in forming this type of network is to involve the minimum number of nodes (hosts) in the routing process because not every node in the network may be required to forward the messages. A solution to this challenge is to identify Minimum Connected Dominating Set (MCDS) [9] among the hosts in a given area. A Dominating Set (DS) is a subset of nodes of a network such that every node that is not in the DS is directly connected to at least one member of DS. A Connected Dominating Set (CDS) is defined as a set of nodes in a network such that each node is either in the set or adjacent to a node in the set. In addition, every node in a CDS should be able to reach every other node in

the CDS by a path that stays entirely within the CDS. In a MANET, the smaller is the size of CDS, the less is the message overhead. This results in improved network routing efficiency. Several existing protocols [14] [15] [16] [17] [18] [19] for MANETs propose different algorithms for CDS construction. In this paper, we propose an efficient CDS construction mechanism which yields CDS of competitive size and low message overhead for both static and dynamic topologies.

We propose an algorithm to compute the CDS of a given network. We assume that all nodes in a MANET are distributed in a two dimensional plane and have an equal maximum transmission range which can be set during simulation as per the requirement. This algorithm begins with each host having no information of the neighboring nodes and uses one-hop neighbor information in the later part of algorithm after the broadcasting phase. With the information obtained from its neighbor nodes, each node arranges the neighbor nodes in the descending order by the number of their number of neighbors. The algorithm then starts from the node with maximum number of neighbors, performs the marking process and continues until all the nodes in the network are covered. The proposed algorithm supports node mobility efficiently compared to existing algorithms. The simulation results also validate that this algorithm indeed constructs a CDS with competitive size and low message overhead.

The rest of this paper is organized as follows. Several existing MANET CDS protocols are reviewed in Section II. Section III describes the proposed algorithm along with elaborate examples with detailed explanation. The issue of nodal mobility in the network is addressed in section IV. The simulation results are presented in Section V. The conclusion is outlined in Section VI.

2. RELATED WORK

One of the most important challenges of MANETs is to reduce the message and routing overhead which can be realized by construction of the smallest possible CDS. Computing minimum CDS is known to be NP-hard [2]. Assuming a completely known network topology, which is practically not possible for MANETs due to node mobility, several researchers have proposed the ways to approximate the computation of minimum CDS [13].

Wu and Li [3] proposed a CDS protocol that consists of two phases. During the first phase, each node collects two-hop neighboring information by exchanging messages with its one-hop neighbors. If a node finds that there is a direct link between any pair of its one-hop neighbors, it removes itself from the CDS. In the second stage, two additional rules are applied to further reduce the size of the CDS. Similarly, Dai and Wu [6] extended the localized algorithm with a generalized pruning rule. This algorithm yields a CDS which is less in size most of the times than the one yielded by the algorithm proposed by Wu and Li [3].

Chen et al. [7] proposed an algorithm (SPAN) to select specific nodes as the coordinators. These coordinators thereby form a CDS and let the other nodes go into low power mode thus saving energy. According to this algorithm, a node becomes a coordinator if 1) it has two neighbors that are not directly connected, or 2) it has two neighbors that are indirectly connected through one or two intermediate nodes. The status change of a node from a non-coordinator to a coordinator is governed by back-off delay such that nodes with less delay are entitled to higher probability of becoming coordinators. The problem with this algorithm is that it cannot ensure a CDS when there is simultaneous change of two coordinators to non-coordinators.

Guha and Khuller [8] proposed an approximation algorithm that involves the construction of a spanning tree with a node which has maximum number of neighbors. All nodes are marked white

initially. The node with the maximum number of neighbors is marked black, added to the tree and its neighbors are colored gray. All these gray colored nodes are added then to the spanning tree. This process continues until all nodes in the network are marked either gray or black and thus the black nodes in the network form the CDS. This algorithm results in high maintenance cost and lack of addressing the mobility issue.

Wang et al. [4] proposed a distributed CDS algorithm. The algorithm assumes a unit-disk graph model. It is also a two-phase protocol. During the first stage, the maximal independent set of the given network topology is constructed in distributed manner by repeated selection of the nodes with the highest number of local neighbors. The nodes in the maximal independent set become the backbone of the CDS. The distance between any pair of its complementary subsets is known to be exactly two hops away, although nodes in the maximal independent set are not connected. During the second phase, a localized search is performed to include additional nodes in CDS to connect the nodes in the maximal independent set. Though the above discussed CDS protocols are distributed and decentralized, they generate a huge number of messages. Also, nodal mobility issue has not been addressed in these protocols which may indicate that if the network topology changes, an entire CDS has to be reconstructed from scratch.

Next, we will review how these protocols support node mobility. MANETs are prone to nodal mobility, and therefore it is expected for the CDS computing algorithms to adapt accordingly to the changing topology of the network. Although Dai and Wu's [6] algorithm extends the mechanism with regards to mobility handling, it demands two-hop neighbor information which clearly implies that it takes two broadcast rounds for a node to accommodate the change in the network topology which may take significantly high message overhead.

An extended mobility handling mechanism has been proposed by Sakai et al. [11]. It incorporates Height-Reduction and Initiator-Reduction procedures to handle the nodal mobility better. The limitations of this algorithm are the extra overhead in the broadcast messages and slow convergence in recovering the CDS. Span [7] algorithm is proactive in nature and uses periodical broadcast of HELLO messages to discover and react to changes in the topology of the network. This algorithm, however, does not yield promising results in terms of CDS size in dense networks.

Our proposed algorithm addresses the aforementioned limitations and provides a reliable and efficient mobility handling scheme with significantly reduced message overhead. The mobility scenarios examined in our proposed algorithm but are handled in an efficient manner so as not to have any single point failures. At the same time, our proposed algorithm also ensures a competitive CDS size.

3. PROPOSED ALGORITHM

A network is modeled as unit disk graph [1] [10] [12] denoted by $G(V, E)$, where V is the set of all hosts (nodes) and E is the set of all edges (links) in the network. A pair of nodes, v_1 and v_2 , that are directly connected with a link are called neighbors. The link that connects v_1 and v_2 is denoted as (v_1, v_2) , and v_1 and v_2 are called the end nodes of (v_1, v_2) . (v_1, v_2) is also incident to v_1 (and v_2). A node is called pendant, if it has only one link incident to it (that is, only one neighbor).

Nodes in the DS are colored black. A node is *covered* if it is the neighbor of a black node, since it can directly receive packets from the black node. A covered node has a gray color. A node with a white color indicates that it is neither a part of a DS nor covered by a node in the DS. The

proposed algorithm consists of three stages and requires no neighborhood information to start with.

- **Broadcasting:** Initially, each node in the network has no information about its neighbor. The algorithm begins with broadcast of a packet that contains the number of neighbors (initially zero for each node) and a node ID (MAC address). After a certain period of time, which is decided based on transmission range of each node, each node will have information about its one-hop neighbors and their corresponding MAC IDs.
- **Sorting of nodes:** $N(v)$ is the number of open neighbors of node v . Nodes in V are now sorted and arranged in the decreasing order of the number of neighbors they have. The node with the highest number of neighbors will be placed first followed by the node with next highest number of neighbors.
- **CDS formation:** There are two phases in the formation of CDS. The first phase performs the coloring process and the second phase finalizes the CDS.

Phase one:

1. All nodes in V are initialized to white color.
2. The first node in V changes its color to black and sends a notification to all its neighbors. On receiving this notification, the white neighbours of this node change their colour to gray.
3. Now consider the second node in V .
 - 3.1 If it is white in color, repeat the above process.
 - 3.2 If it is gray in color, check to see if it has any white neighbors. If yes, the color of the second node is changed to black and its white neighbors turn gray after receiving the notification from the second node.
4. The above process continues until there are no more white nodes in the network. Note that this phase is carried only until the network gets exhausted of white nodes therefore not checking for all nodes in the network. This saves computation time.
5. After the above steps are completed, if any gray node in the network is found to have at least two black neighbors, it is a possible candidate for a CDS and hence should be colored yellow.

Phase two:

1. Check for yellow nodes in the network, and if present,
 - 1.1 If the neighbors of the yellow node are not all present in the neighbor set of either/both of the two black neighbors, the color of the node is changed from yellow to black. Otherwise, the color is changed from yellow to gray.
2. If a black node has at least two black neighbors,
 - 2.1 If the neighbors of the black node are all present in the neighbor set of either/both the two black neighbors, the color of the node is changed from black to gray.
3. Now, all the black nodes in the network form a CDS.

3.1. CDS Formation Example

Consider the following network of ten nodes which represent set of vertices V of a graph G . The graph is assumed to be a unit disk graph. There exists an edge between two nodes if and only if they are within the transmission ranges of each other.

- Initially none of the nodes has any information about their neighbors so each node broadcasts a message with information about its MAC address and the number of

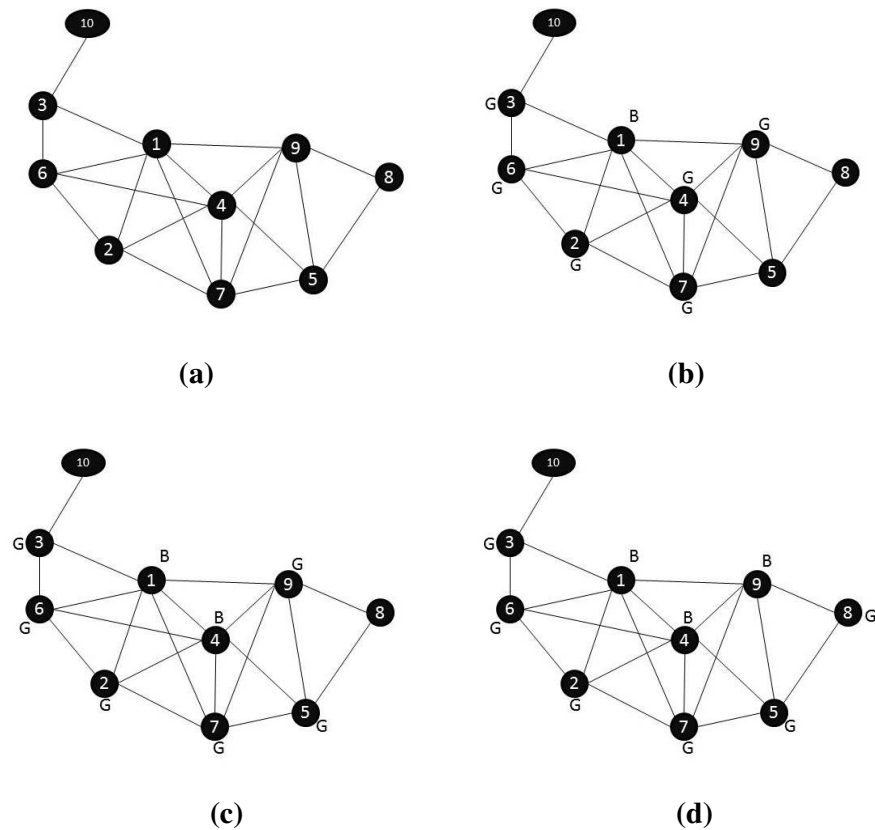
of neighbors it has (initially zero).

- After a certain time, each node updates its database with the information it receives from its one-hop neighbors, i.e., MAC address and number of neighbors. Table 1 lists the information that each node has received. Now, by applying the node sorting procedure i.e., arranging the nodes in the decreasing order of their number of neighbors, we can deduce $V = \{1, 4, 9, 7, 6, 2, 5, 3, 8, 10\}$.
- CDS formation:

Phase 1: All ten nodes are initialized to white as shown in Figure 1(a). The color of the first node in the set V (node 1 in this case) is changed to black and its white neighbors are colored gray as shown in Figure 1(b). Next node in V (node 4) has white neighbor node 5 and therefore is changed to black and node 5 is changed to gray as shown in Figure 1(c). The color of node 9 is changed to black since it has a white neighbor node 8 which turns gray as shown in Figure 1(d). Similarly, node 3 is changed to black and node 10 to gray as shown in Figure 1(e). Now there are no white nodes in the network. Node 6, 2, 7, 5 have two black neighbors and therefore colored yellow as shown in Figure 1(f). This marks the end of phase 1.

Phase 2: Node 6 has three black neighbors 1, 4 and 3. $N(6) = \{3, 1, 4, 2\}$; $N(1) = \{3, 6, 2, 7, 4, 9\}$, $N(4) = \{6, 1, 2, 7, 5, 9\}$ and $N(3) = \{10, 6\}$. It can be observed that the neighbors of node 6 are neighbors to either node 1, 4 or 3. Therefore, node 6 is marked gray. Similarly, 2, 7 and 5 are marked gray.

Node 1, 4 and 9 form a triangle of black nodes; therefore, each node has two black neighbors, respectively. Let us consider those three nodes. $N(4) = \{6, 1, 2, 7, 5, 9\}$; $N(1) = \{3, 6, 2, 7, 4, 9\}$ and $N(9) = \{1, 4, 7, 5, 8\}$. It can be observed that neighboring nodes of node 4 are neighboring nodes to either node 1 or node 9. Therefore, node 4 can be marked gray. The same applies to



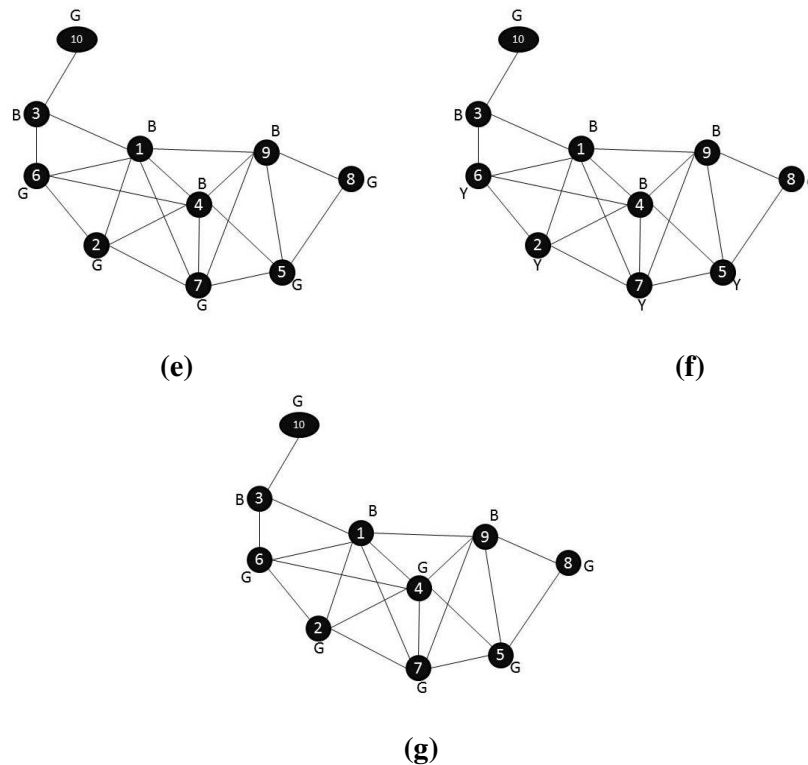


Figure 1. An example of a network and its CDS formation.

node 1 and 9. This marks the end of phase 2. From the network in Figure 1(g), it can be observed that 3-1-9 forms a MCDS.

3.2. Correctness

Lemma 1. If a node in the network G leaves the network, an alternative CDS within the new network will be computed if necessary.

Proof. If the node that left is not present in the CDS, phase two of the proposed algorithm is applied to the nodes in the CDS to identify redundant black nodes, if any, that can be marked gray. According to phase two, if the neighbor set of a black node is shared by any other black node in the CDS, the former black node is considered redundant and therefore marked gray. This reduces the size of the CDS. On the contrary, if the node that left is in the CDS, a gray node from its neighbor set is selected such that the selected node has no black neighbors. From the neighbor set of the selected gray node, a gray node is selected and marked black so that it forms a part of the newly formed CDS.

Lemma 2. If a new node enters the network G , an alternative CDS within the new network will be computed if necessary.

Proof. The new node entering the network is initially marked white for other nodes to identify it as a new node. It will then broadcast its id to gather information about neighboring nodes within one-hop distance. If the new node has a black node as its neighbor, it marks itself gray and the

size of the CDS remains same because there is no need for reformation of the CDS in this case. If the new node has no black neighbor, it will scan through its neighbor set to find a node which has a black neighbor and marks that node black thereby forming a new CDS.

Table 1. Number of neighboring nodes in Figure 1.

Node # (i)	Neighboring nodes $N(i)$	Number of neighboring nodes
1	3, 6, 2, 7, 4, 9	6
2	1, 4, 6, 7	4
3	1, 6, 10	3
4	1, 2, 5, 6, 7, 9	6
5	4, 7, 8, 9	4
6	1, 2, 3, 4	4
7	1, 2, 4, 5, 9	5
8	5, 9	2
9	1, 4, 5, 7, 8	5
10	3	1

Theorem 1 The set of all the nodes marked black by the proposed algorithm forms a CDS of the network if the topology of the network remains unchanged for a duration of time.

Proof. In this theorem, it is claimed that if there is no change in the topology of the network under consideration, the nodes which have been marked black will form a CDS. This theorem is proved by induction on the number of nodes of the network N . Let us consider the base case for induction when $N = 1$. This means that there is only one node in the entire network and it has no neighbors. After the phase one of the proposed algorithm, the node marks itself black and hence is a CDS. Therefore, the aforementioned claim is rendered true. As an inductive step, it is assumed that $m - 1$ nodes are being covered by the CDS of a network which is denoted by $G(m - 1)$. Now, let us consider the case when a new node m enters the network. If the new node m has a black node (a member of the CDS) as its neighbor, the CDS will remain intact since the new node is being covered by the existing CDS. Therefore, the CDS which is covering $G(m - 1)$ is same as the one covering $G(m)$. If the new node m has no black neighbors, the neighbor set of m has the nodes which have nodes that are members of the CDS of $G(m - 1)$. According to the proposed algorithm, when a new node m is being identified by one of the new node's neighbor which has a black node in its neighbor set, the new node is marked gray by the its neighbor which marks itself black. Therefore, the new node is covered by the newly formed CDS. This proves that the CDS covering $G(m - 1)$ and the new node included is same as the CDS covering $G(m)$.

4. MOBILITY SUPPORT

A MANET has a dynamic network topology because of the continuous movement of nodes. This algorithm handles the nodal mobility involved in MANETs effectively by the following three different mobility cases. Figure 2 shows another example of network. Table 2 lists the number of neighboring nodes in Figure 2.

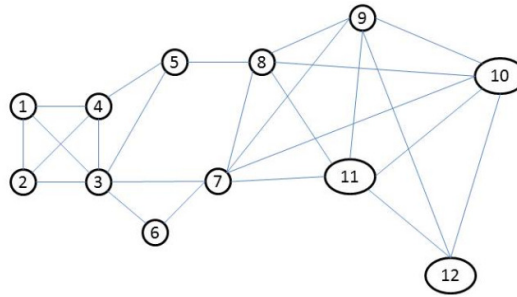


Figure 2. Another example of a network

Here, the order of the nodes with the decreasing number of neighbors is $V = \{3, 7, 8, 9, 10, 11, 4, 1, 2, 5, 12, 6\}$. In the above example, the possible CDSs are $\{3-7-9\}$, $\{3-7-11\}$, and $\{3-7-10\}$. For better understanding, let us consider the CDS to be $\{3-7-9\}$. The nodes in the CDS are black in color whereas the remaining nodes are gray according to the coloring scheme of the proposed algorithm. To support mobility, there are three cases, Case 1 (when a non-CDS node leaves the network), Case 2 (when a CDS node leaves the network), and Case 3 (when a new node joins the network). Cases 1 and 2 are proved in lemma 1 while case 3 is proved in lemma 2. Following are examples to illustrate the proposed algorithm's handling of nodal mobility.

Case 1: If a non-CDS node (a gray node) leaves the network, algorithm will check to see if the size of the CDS can be reduced due to this recent change in the network topology. In Figure 2, if node 12 decides to leave the network, the CDS reduces to $\{3-7\}$. This is because all the neighbors of node 9 are also neighbors to another black node 7; therefore, according to phase 2, node 9 can be changed to gray. If there is no possibility of reducing the CDS size, the CDS will remain the same.

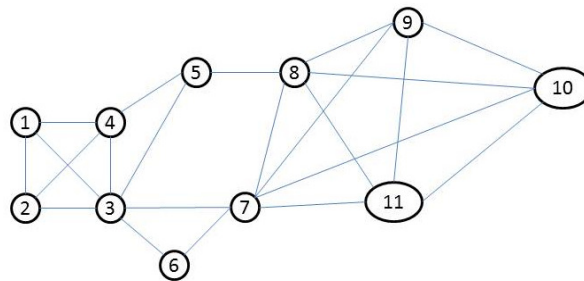


Figure 3. CDS changed to 3-7 after node 12 left the network

Case 2: If a node in the CDS (black node) leaves the network, the following cases are to be considered

- a) In some cases, when a node in the CDS leaves the network, two distinct disconnected networks are formed and therefore each network forms its own CDS. In Figure 4, the CDS consists of node 3, 4 and 5. If node 4 leaves the network, it can be observed that two disconnected networks will be formed. One of them will have the node 1, 2, 3, 9 and 10 whereas the other network will have the node 5, 6, 7 and 8. Each of these networks will form its own CDS.
- b) In the Figure 2, suppose that node 9, which is a part of the CDS, leaves the network. Then, there is a change in the topology of the network, as shown in Figure 5, and requires some computation to see if an alternative CDS can be formed with the given nodes of the

network. The neighbor set of node 9 is $N(9) = \{7B, 8G, 10G, 11G, 12G\}$. Node 9 has only one black neighbor (node 7) and the remaining neighbors are all gray. Among the gray nodes, node 8, 10 and 11 have at least a black node as their neighbor but node 12 has no black neighbor. The neighbor set of node 12 is $N(12) = \{10G, 11G\}$. From the neighboring set of node 12, either node 10 or 11 should change to black in order to form a MCDS. Therefore, the resulting CDS is 3-7-10 or 3-7-11.

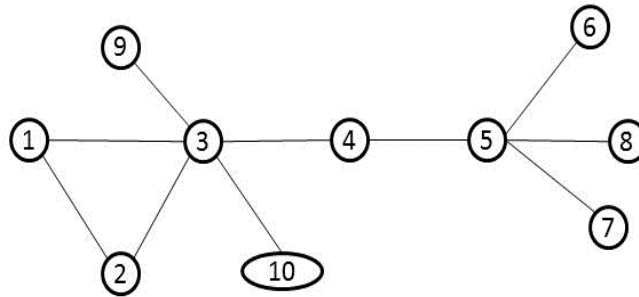


Figure 4. Formation of disconnected networks.

Case 3: If a new node joins the network, the algorithm will mark its color white and check the following:

- a) If the new node is a neighbor to a black node as shown in Figure 6, it will change its color from white to gray. There does not need to be any change in the CDS as the new node is already a neighbor to a black node.
- b) If the new node is not a neighbor to a black node, it needs to find a neighboring node which has black node as its neighbor and changes the color of that node to black. In this case as shown in Figure 7, node 12 is a neighbor to both the black node 9 and the new (white) node 13. Therefore, the color of node 12 changes to black thereby changing the CDS to $\{3-7-9-12\}$.

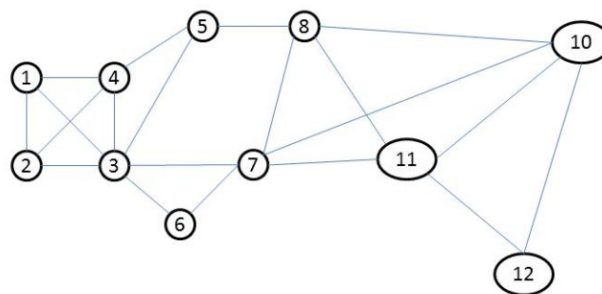


Figure 5. Node 9, a part of CDS in Figure 2, leaves the network.

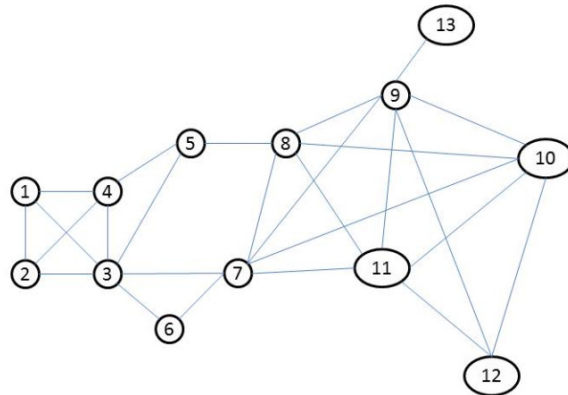


Figure 6. New node 13 enters the network.

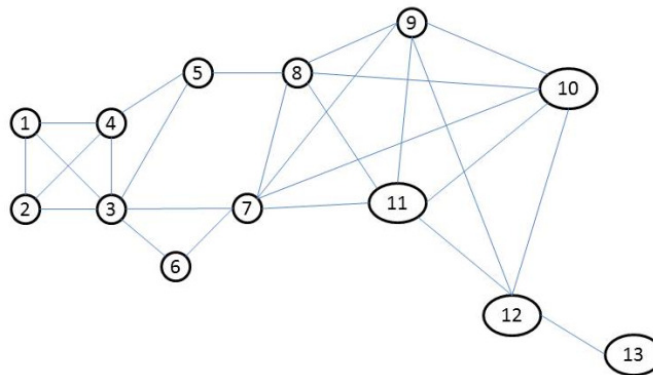


Figure 7. A CDS size change due to a new node 13.

Table 3 summarizes many algorithms reviewed in Section II whether the algorithm supports the nodal mobility or not.

Table 3. A comparison of nodal mobility support.

Algorithm	Mobility Support
[3]	Compute the CDS from scratch
[4]	None
[6]	Compute the CDS from scratch
[7]	Yes
[8]	None
Proposed	Yes

5. SIMULATION

A simulation study has been conducted in a custom built, standalone C++ simulator to study the performance of the proposed algorithm against existing CDS computing algorithms. A number of hosts in a MANET are randomly scattered in a $100m \times 100m$ area. There exists a link between a pair of hosts if they are within the transmission range of each other which is currently 50m. All links in the network are assumed to be bidirectional. To evaluate the performance of these algorithms, two metrics are studied, i.e., the number of messages required to form CDS and the size of the CDS, with increasing number of nodes in the network.

Figure 8 shows the CDS size of several algorithms (MCDS [8], CDS Span [7], Wu and Li [3], Dai and Wu [6]), including the proposed algorithm with respect to the increasing number of nodes in the network. It is quite evident that our proposed algorithm yields the smallest CDS among other algorithms. The smaller is the CDS size, the better is the performance of network when nodal mobility is considered because the lesser is the CDS size, the lesser is the probability of the nodes moving out of the CDS.

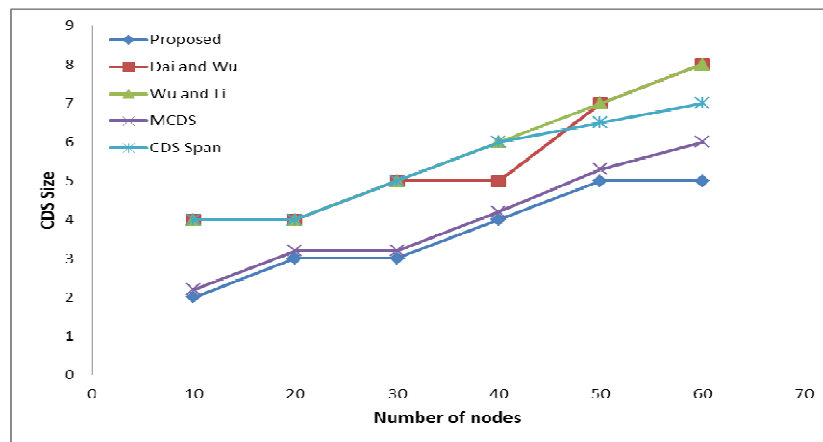


Figure 8. Comparison of the CDS size of the proposed with existing algorithms.

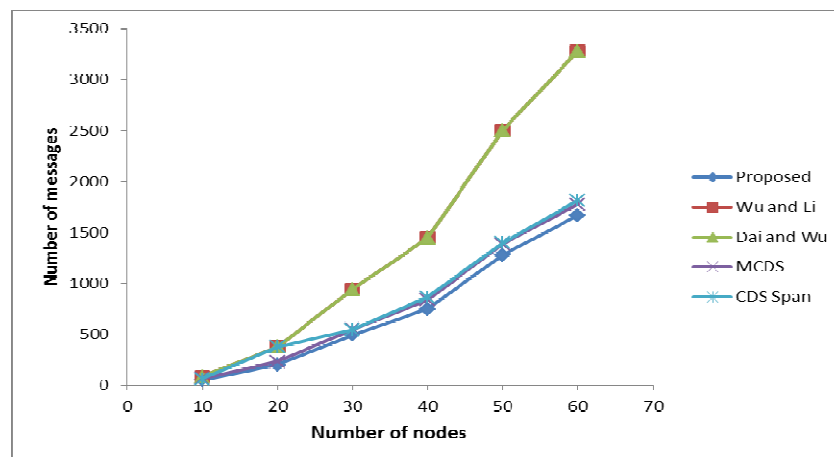


Figure 9. Comparison of the number of message exchanges in the proposed with existing algorithms.

Figure 9 illustrates the number of messages exchanged to form a CDS with respect to number of nodes in the network. Our proposed algorithm generates the least number of messages in CDS formation when compared to the other algorithms. The algorithms proposed by Wu and Li [3] and Dai and Wu [6] exchange the same amount of messages but differ in the procedure of reducing the CDS size. Hence, the proposed algorithm has very low message overhead since the number of messages exchanged between the nodes is significantly low compared to other algorithms.

6. CONCLUSION

This paper presents an efficient algorithm for CDS construction in mobile adhoc networks. Our approach is based on sorting the nodes in the decreasing order of their degree and forming a CDS beginning with the node with the highest degree in the network. Pruning techniques are applied to remove any redundant nodes from the CDS and thereby reducing message overhead. The performance of the proposed algorithm has been verified through extensive simulations for both static and dynamic topologies. The simulation results have validated that this algorithm outperforms several existing algorithms in terms of size of CDS and message overhead. The size of the CDS generated by our proposed algorithm is significantly smaller than the one generated by several existing algorithms. The message overhead in the proposed algorithm is found to be significantly less as well when compared to several existing algorithms. Also, the algorithm adapts to dynamic topologies by following a set of rules and guarantees a CDS at almost all times without rebuilding from scratch.

REFERENCES

- [1] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit Disk Graphs", *Discrete Mathematics*, 86:165-177, 1990.
- [2] D. S. Hochbaum, *Approximation Algorithms for NP-Hard Problems*, PWA Publishing Company, Boston, MA, 1995.
- [3] J. Wu, H. L. Li, "On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks," *Proc. of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pp. 7-14, Aug. 1999.
- [4] P. J. Wan, K. M. Alzoubi, O. Frieder, "Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks," *ACM Mobile Networks and Applications*, vol.9, issue 2, pp. 141-149, Apr. 2004.
- [5] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," *Wireless Networks*, vol. 8, nos. 2/3, pp. 153-167, Mar.-May 2002.
- [6] F. Dai and J. Wu, "An Extended Localized Algorithm for Connected Dominating Set Formation in Ad Hoc Wireless Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, issue 10, pp. 908-920, Oct. 2004.
- [7] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," *ACM Wireless Networks J.*, vol. 8, no. 5, pp. 481-494, Sept. 2002.
- [8] S. Guha and S. Khuller, "Approximation Algorithms for Connected Dominating Sets," *Algorithmica*, vol. 20, no. 4, pp. 374-387, Apr. 1998.
- [9] S. Butenko, X. Cheng, C. Oliveira and P. M. Paradlos, "A New Heuristic for the Minimum Connected Dominating Set Problem on Ad Hoc Wireless Networks," *Recent Developments in Co-operative Control and Optimization*, pp. 61-73, Kluwer Academic Publishers, 2004.
- [10] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, "Introduction to Algorithms," 2nd Edition, MIT Press, 2001.
- [11] K. Sakai, M.-T. Sun and W.-S. Ku, "Maintaining CDS in Mobile Ad hoc Networks," *Wireless Algorithms Systems and Applications, Lecture Notes in Computer Science*, vol. 5258, pp. 141-153, October 2008.

- [12] F. Kuhn, T. Moscibroda and R. Wattenhofer, "Unit Disk Graph Approximation," Proceedings of the ACM DIALM-POMC Joint Workshop on the Foundations of Mobile Computing, pp. 17–23, Philadelphia, October 2004.
- [13] P.-R. Sheu, H.-Y. Tsai, Y.-P. Lee and J. Y. Cheng, "On Calculating Stable Connected Dominating Sets Based on Link Stability for Mobile Ad hoc Networks," Tamkang Journal of Science and Engineering, vol. 12, no. 4, pp. 417 – 428, 2009.
- [14] A. Raj, D. Saha, P. Dasgupta , "A cost-efficient algorithm for finding connected dominating sets in static wireless ad hoc networks with obstacles," *Advanced Networks and Telecommunication Systems (ANTS), 2010 IEEE 4th International Symposium*, pp.73-75, Dec 2010.
- [15]. Bolian Yin, Hongchi Shi, Yi Shang, An efficient algorithm for constructing a connected dominating set in mobile ad hoc networks, *Journal of Parallel and Distributed Computing*, Volume 71, Issue 1, January 2011, Pages 27-39.
- [16]. Chun Meng, Meina Song, Junde Song, Junmin Jia, "A Dominating-Set-Based Broadcast Gossip Protocol in Mobile Ad Hoc Networks", *Wireless Communications, Networking and Mobile Computing , 2008. WiCOM '08. 4th International Conference on*, pp.1-5, Oct. 2008.
- [17]. C.D. Young, A.D. Amis, "UCDS: Unifying connected dominating set with low message complexity, fault tolerance, and flexible dominating factor," *Military Communications Conference (MILCOM 2011)*, pp.1357-1362, Nov. 2011.
- [18]. M. Rai, N. Garg, S. Verma, S. Tapaswi, "A New Heuristic Approach for Minimum Connected Dominating Set In Adhoc Wireless Networks," *Advance Computing Conference, IACC 2009. IEEE International*, pp.284-289, March 2009.
- [19]. V. Raghavan, A. Ranganath, R. Bharath, M. Khan, Nagappan, "Simple and Efficient Backbone Algorithm for Calculating Connected Dominating Set in Wireless Adhoc Networks", *World Academy of Science, Engineering and Technology* 33, 2007.

Authors

Mallikarjun Avula received his B.Tech degree in Electronics and Communication Engineering from Jawaharlal Nehru Technological University, India in May 2007. He received his M.S degree in Electrical Engineering from University of Alabama in Huntsville (UAHuntsville) in December 2009. He is currently a Ph.D. candidate in Computer Engineering from Electrical and Computer Engineering Department, UAHuntsville. His major research interests are mobile ad hoc networks, wireless sensor networks, and network security.

Seong-Moo Yoo is an Associate Professor of Electrical and Computer Engineering at the University of Alabama in Huntsville (UAH). Before joining UAH, he was an Assistant Professor at Columbus State University, Columbus, Georgia – USA. He earned MS and PhD degrees in Computer Science at the University of Texas at Arlington. His research interests include computer network security, wireless network routing, and parallel computer architecture. He has co-authored over 70 scientific articles in refereed journals and international conferences. He is a senior member of IEEE and a member of ACM.

Seungjin Park received the BE degree in civil engineering from Hanyang University, Korea, in 1973, MS degree in computer science from University of Texas at Arlington, in 1986, and PhD degree in computer science from Oregon State University, in 1993. He is currently an assistant professor in the Department of MNGT, MIS, and CS, University of Southern Indiana. His research interests include parallel algorithms, wireless ad hoc networks, and sensor networks.