

Connected Dominating Set Construction Using an Efficient Pruning Method in Ad hoc Networks

Vahid Ghasemi, Seyed Naser Hashemi and Mojtaba Mozaffari

Department of Computer Science
Amirkabir University of Technology
Tehran, Iran

{vahid.ghasemi, nhashemi, mozaffar}@aut.ac.ir

Abstract— In a broadcasting process in wireless ad hoc networks, the number of retransmissions of a packet i.e. the number of forwarder nodes, is generally considered as the cost of broadcasting. A connected dominating set (CDS) is an infrastructure used to manage broadcast of packets such that only nodes in the CDS forward the packets. In this paper an enhanced algorithm to construct a connected dominating set is presented. In the first step, we use a known local algorithm to determine the primary CDS. Then an efficient probabilistic pruning process is suggested to implement the secondary CDS. As a result, the number of nodes in the pruned CDS decreases efficiently. The performance of the suggested method, in terms of the size of the secondary CDS and the message length of broadcast, is discussed and compared to an efficient local algorithm based on complete 2-hop information having a message length $O(m^2)$, where m is the maximum degree of nodes in the network. Simulations show that our method is good for sparse networks, while for dense networks, it works almost similarly like the algorithm using complete 2-hop information. Our algorithm has the message length $O(m)$ and decreases the size of CDS significantly. Therefore, it has a better performance in contrast to the algorithm with complete 2-hop information.

Keywords-broadcasting; connected dominating set (CDS); wireless ad hoc networks; flooding; pruning;

I. INTRODUCTION

A mobile ad hoc network is a set of mobile devices which communicate with each other using wireless media, and can move freely from one place to another having no fixed infrastructure. Broadcasting in ad hoc networks is a problem that has attracted special attentions. The easiest way to broadcast packets is blind flooding in which every recipient of a packet rebroadcasts it once. But this method has some problems, including an increase in the rate of packet collisions, repeating the broadcast of packets unduly, and hence wasting resources such as energy of nodes and network bandwidth (also known as broadcast storm problem [16]).

Broadcasting problem in mobile ad hoc networks (MANETs) can be managed effectively by determining a small subset of nodes as a connected dominating set (CDS). A dominating set (DS) in a network is a subset of nodes such that every node in the network is either adjacent to at least one node in DS, or belongs to DS. If the induced graph of nodes in DS is connected then DS is said to be a connected dominating set. In fact nodes of CDS cover all nodes in the network, and in a

broadcasting process it is enough that only the nodes of CDS, like a virtual backbone, rebroadcast packets. The less number of nodes in the CDS, the less broadcasting is done unduly and therefore the less wasting of resources happens. So it has always been struggled to minimize the number of nodes in the CDS. A CDS is minimum, if it has the smallest number of nodes in the network as possible. Determination of minimum CDS (MCDS) is an NP-complete problem [1]. So far, a large number of methods have been proposed to estimate a MCDS. But many of these methods use the global information of the network (such as topology information of the whole network) or global structures (such as spanning trees). So they are not suited for MANETs because in a MANET, the topology changes continuously and global information or structures will not be stable. In a stable system, topology changes must be transferred in real time and considered in computations.

Local algorithms are stable solutions for calculating CDS in MANETs. In local algorithms to determine CDS, k -hop neighborhood information of every node is used, where k is a small number. Generally, a k -hop neighborhood of a node includes nodes at most k links away from it. So if neighborhood information is sent periodically via "HELLO" messages, it takes k rounds for a node to gather its k -hop neighborhood information. It should be considered that k must not be too large because collecting information on time will not be possible.

Some local algorithms to determine CDS are based on multipoint relays (MPR) [2] which in turn are based on 2-hop neighborhood information of a node. In these methods, a small subset of 1-hop neighbor nodes, called MPR nodes which cover all 2-hop neighbors, is selected. In source independent algorithms, which will be discussed in the next section, MPR nodes are not essentially packet forwarders, and MPR calculation algorithms are just used to determine the status of a node (whether a node becomes MPR or not).

In this paper a probabilistic method to locally prune a CDS resulted from a source independent and MPR based algorithm is proposed. Our algorithm has message length $O(m)$ where m is the maximum degree of a node in the network, and effectively decreases the number of nodes in the CDS. We will show the efficiency of our algorithm in terms of number of nodes in the CDS and its message size in contrast to an

efficient source independent algorithm that has been proposed so far.

The rest of the paper is organized as follows: Section 2 discusses some related works on CDS construction algorithms and their features. Section 3 describes the graph model used throughout the paper. In Section 4 we present our proposed algorithm. Message length of the proposed algorithm is discussed in Section 5. Simulation results are shown in section 6 and at last Section 7 concludes the paper.

II. RELATED WORK

The question of determining minimum CDS in a general network is proved to be NP-Complete [1]. It is also shown to be NP-Complete for unit disk graphs too [6]. There are algorithms that estimate a minimum CDS. These algorithms are divided into four general categories: local [2], [3], [4], [5], [11] quasi-local [10], global [7], [8] and quasi-global [9].

Primary works to estimate CDS resulted in global algorithms [8] in which approximation ratio is $O(\ln(m))$ where m is the maximum degree of nodes in the network. Quasi-global algorithms [9] use methods based on shortest path trees to construct a CDS having an approximation ratio of $O(1)$. In quasi-local algorithms [10] at first some cluster heads for a group of nodes are selected then these nodes are connected using gateway nodes. These methods have an approximation ratio of $O(1)$ in unit disk graphs.

Local algorithms use k -hop information of nodes to estimate the CDS. These algorithms can be source dependent or source independent. In source independent algorithms, CDS is determined based on network topology and priority of nodes. In these methods the structure of CDS is static, meaning while network topology is not changed, the structure of CDS is fixed. The source independent MPR based algorithm in [3] and its extensions in [4] and [5] and also algorithms in [12] and [13] are in this family.

In [3] a source independent algorithm is presented in which every node calculates its MPR set using a greedy algorithm [2]. Then a node is included in the CDS via 2 rules: 1- It has the smallest ID in its 1-hop neighborhood. 2- It is selected as MPR by its neighbor with the smallest ID. In this algorithm every node broadcasts its 1-hop neighborhood information via "HELLO" messages. Then the size of broadcast messages is $O(m)$ where m is the maximum degree of nodes in the network. We call this algorithm AD (stands for Adji's algorithm) throughout this paper.

In [4] AD algorithm is optimized and size of the CDS is reduced more, using partial 2-hop neighborhood information and defining 1-hop free neighbors. Partial k -hop neighborhood information of a node, namely v , is a subgraph $G_k(v) = (N_k(v), E_k(v))$ such that $N_k(v)$ is a set of nodes within k -hop neighborhood of v and $E_k(v)$ is a set of all links between nodes in $N_k(v)$ except for links between nodes at exactly k hops from v . In this method two changes are made in AD algorithm. Firstly, in MPR determination algorithm prior to any thing, free neighbors are added to MPR set of a node and their neighbors are marked as covered then other steps are done. Secondly, rule 1 in AD algorithm is changed as: node with the smallest ID

among its neighbors that has at least two unconnected neighbors is added to CDS. In [4] simulations show that this algorithm reduces the size of CDS more than AD algorithm. In this algorithm every node sends its 1-hop neighborhood information to its neighbors via "HELLO" messages periodically. So the size of "HELLO" messages will be $O(m)$ where m is the maximum degree of nodes in the network. In this method the size of CDS in the worst case is $O(n)$ that n is the number of nodes in the network. This algorithm is called WP (stands for Wu's Partial 2-hop information algorithm) throughout this paper.

In [5], using complete 2-hop neighborhood information, and introducing 2-hop free neighbors, WP algorithm is optimized again. Complete k -hop neighborhood information of a node, called v , is a subgraph $G'_k(v) = (N'_k(v), E'_k(v))$ such that $N'_k(v)$ is a set of nodes within k -hop neighborhood of v and $E'_k(v)$ is a set of all links between nodes in $N'_k(v)$ including links between nodes at k hop distance from v . In this algorithm first of all every node, called v , calculates $C(v)$ that is a set of pairs (u, w) such that u is in 1-hop neighborhood of v and w is in 1-hop neighborhood of u . $C(v)$ covers all nodes in 2-hop neighborhood of v . Then using an extended algorithm, CDS is calculated in the network. Since in this method complete 2-hop neighborhood information is used, every node broadcasts its 2-hop neighborhood information via "HELLO" messages. So the size of "HELLO" messages is $O(m^2)$ in which m is the maximum degree of nodes in the network. This method is an efficient source independent algorithm in term of average number of nodes in the CDS that has been proposed so far. In [5] it is shown that every node selects at most 48 forwarders locally. However, this constant approximation ratio is local and does not determine the overall size of the CDS. The size of CDS in the worst case remains $O(n)$ where n is the number of nodes in the network. Simulations show that this algorithm is more efficient than WP in term of average number of nodes in CDS but its message length is worse than that of WP. This algorithm is called WC (stands for Wu's Complete 2-hop information algorithm) throughout this paper.

In source dependent algorithms, calculation of CDS depends on some factors like the source of a broadcast, distance between nodes and packet transmission delay. In these algorithms when broadcasting starts, every node that receives the broadcast packet, calculates whether it is (and/or some of its neighbors are) forwarder or not. With this scheme, as the broadcast packet is forwarded along the nodes, CDS is formed dynamically, meaning that CDS has no fixed structure even without topology changes. The source dependent MPR based algorithm in [2], dominant pruning algorithm in [14] and the approach of [15] are in this family.

Local algorithms for calculating a CDS can also be pruning based [11], [12], neighbor-designing [3], [4], [5] or hybrid [17]. In pruning based algorithms a node is included in the CDS if it has two neighbors that can't communicate with each other through another node with a higher priority. In neighbor-designing algorithms, the status of a node is determined firstly and then, whether the node becomes a member of CDS or not, is calculated according to local neighborhood information. The hybrid methods are combinations of previous ones.

In this paper a hybrid method to calculate a CDS is proposed in which the CDS resulted from a source independent algorithm i.e. WP algorithm, is pruned probabilistically using local information. In our algorithm the size of CDS is reduced significantly and the message length is $O(m)$, having a better performance compared to WC algorithm.

III. GRAPH MODEL

A MANET can be shown as a graph $G=(V,E)$ in which V and E denote network nodes and links between nodes respectively. It is supposed that every node has a unique attribute like MAC address or IP address by which it can be identified uniquely; this attribute is referred to as ID shortly. G is usually a unit disk graph meaning that two nodes are neighbors if and only if their distance is less than a transmission range r . We use the concept of unit disk graphs for ad hoc networks in our approach. Through this paper $N_k(v)$ is a set of nodes within k -hop neighborhood of v including v , where k is a natural number. A two way path between two nodes, namely i and j , is shown in parenthesis like $(i, v_1, v_2, v_3, \dots, j)$, in which " v_k "s are intermediate nodes and a bidirectional link between two neighbors is shown using symbol " \leftrightarrow ".

IV. PROPOSED PRUNING METHOD FOR CALCULATING CDS

The CDS constructed by WP algorithm in Section I has a big size, because this method is using the partial local information. In this paper, we present a pruning algorithm for reducing efficiently a number of nodes in the CDS resulted from WP algorithm. Therefore, for the formation of CDS, the following two steps are performed:

- **Step 1.** Computation of primary CDS using the WP algorithm.
- **Step 2.** Constructing secondary CDS by pruning the primary CDS using local information.

Throughout this paper the notation CDS_1 refers to the primary CDS constructed by WP algorithm, and CDS_2 stands for the pruned CDS_1 as secondary CDS.

By the definition of CDS, each node in the network has at least one adjacent node in CDS. In the proposed pruning algorithm, after the determination of CDS_1 in Step 1, every node selects only one node of its neighbors that belongs to CDS_1 for forwarding messages. This node is called secondary forwarder. In this way, if all neighbors of a node, namely $u \in CDS_1$, select together the same node $w \in CDS_1$ as a secondary forwarder, then the algorithm omits node u from CDS structure because they can communicate with each other through w , and u is useless. Thus these secondary forwarders must stay in secondary CDS i.e. CDS_2 . From this basic idea, the following problem arises, which scheme of its solution will be discussed.

Problem 1: How can the neighbors of a node $u \in CDS_1$ select the same node $w \in CDS_1$ as a secondary message forwarder?

To solve Problem 1 and Problem 2 (which will be seen later), these definitions are presented:

Definition 1: A CDS-selector(w) is said to be a set of all nodes that select w as their secondary forwarder.

Definition 2: In comparing two nodes, it is said that a node, i , is smaller than a node, j , when the following relation is satisfied:

$$i < j \leftrightarrow (\text{degree}(i) > \text{degree}(j)) \text{ or } (\text{degree}(i) = \text{degree}(j) \text{ and } \text{id}(i) < \text{id}(j)),$$

where $\text{id}(\cdot)$ denotes the identification of a node.

A scheme for the solution of problem 1: For this problem, a probabilistic approach is introduced to construct the elements of CDS_2 using the primary CDS (i.e. CDS_1). The question arises: how would be a secondary forwarder node selected in CDS_1 so that all neighbors of a node $u \in CDS_1$ can most likely select only one common node $w \in CDS_1$ as a secondary node for forwarding messages. Clearly, the higher degree of a node, the more likely it will be selected as a secondary node. Therefore, for the node $u \in CDS_1$, if a node $v \in N_1(u)$ selects the highest degree node $w \in CDS_1 \cap N_1(v)$ as a secondary forwarder, then this node, i.e. w , can be most likely selected as a secondary forwarder by the other neighbors of u . If two nodes have the same degree, then the node with smaller ID can be selected as a secondary forwarder. In fact node v selects its smallest neighbor (see Definition 2) that belongs to CDS_1 as secondary forwarder.

Now after selection of the most likely node, w , the second question arises:

Problem 2: How can one ensure that the selected node w will remain in CDS_2 ?

A Scheme for the Solution of problem 2: To answer this question, we postulate the condition that a node $w \in CDS_1$ cannot be omitted from CDS structure, if this node is selected as a secondary forwarder in the network i.e. CDS-selector(w) is not empty. Considering this condition, we ensure that this node will remain in CDS_2 .

On the other hand, for cancellation of a node $u \in CDS_1$ the following conditions must be satisfied:

- **Condition 1.** CDS-selector(u) = \emptyset .
- **Condition 2.** The neighbors of u are connected through a common node $w \in CDS_2 \subseteq CDS_1$.

Now, in the next section, we explain the development of our algorithm illustrating the cancellation process of nodes in the primary CDS, i.e. CDS_1 , to obtain the smaller and more efficient CDS, i.e. CDS_2 .

A. Pruning process

For omitting a node from the CDS, so far only the 1-hop neighborhood information of nodes has been used i.e. every node selects secondary forwarder from its 1-hop neighborhood as discussed. Since in our method choice of the most likely node plays an important role in pruning the CDS, we attempt to increase the probability that neighbors of a node $u \in CDS_1$ select the same node $w \in CDS_1$ as a secondary forwarder.

To implement this basic idea, first a pair of nodes, (v_i, w_i) , for a node i in the network is selected, where $w_i \in CDS_1$

represents the smallest node (see Definition 2) within 2-hop neighborhood of node i , i.e. $N_2(i)$, and $v_i \in CDS_1$ is the smallest intermediate node within 1-hop neighborhood of i , i.e. $N_1(i)$, laying between i and $w_i \in CDS_1$. After selecting this pair, node i is added to $CDS_selector(v_i)$.

This idea is conveyed in Fig. 1. Suppose $u \in CDS_1$ has two neighbors v_1 and v_2 that are located on the border of the unit disc centered at u . If v_1 and v_2 select their forwarders, called w_1 and w_2 respectively, from CDS_1 and their 1-hop neighborhoods, for sure $w_1 \neq w_2$ because 1-hop neighborhoods of v_1 and v_2 have no intersection. In contrast, if v_1 and v_2 select w_1 and w_2 respectively, from their 2-hop neighborhoods, i.e. $N_2(v_1)$ and $N_2(v_2)$, since the intersection of 2-hop neighborhoods of v_1 and v_2 , namely S in Fig. 1, is not empty, the possibility that $w_1 = w_2$ increases. In this case, the possibility that a path exists between v_1 and v_2 through intermediate nodes and w_1 (or w_2) such that doesn't pass through u is increased and u is more likely to be omitted from CDS_1 .

Now the selection rule for choosing a pair (v,w) , associated with a node, is defined.

Selection Rule: Every node of the network, called u , selects a pair (v,w) and then is added to $CDS_selector(v)$ in such a way that the nodes w and v satisfy following conditions :

- **Condition 1.** w is the smallest node (according to Definition 2) in $CDS_1 \cap N_2(u)$.
- **Condition 2.** v is the smallest node (according to Definition 2) in $CDS_1 \cap N_1(w) \cap N_1(u)$.

The components of the selected pair may be identical due to Selection Rule. This means that node $u \in CDS_1$ selects the pair (u,u) if it is the smallest node within $N_2(u) \cap CDS_1$ and also it selects the pair (v,v) where $v \in N_1(u) \cap CDS_1$ is the smallest node within $N_2(u) \cap CDS_1$.

Each node in CDS_1 , with the use of the Selection Rule, chooses a pair of nodes which can be informed to its neighbors through a "HELLO" message broadcast. Indeed, the nodes update their selector set through this "HELLO" message. From this action, each node saves the information about the chosen pair of its neighbor nodes. It is important to note that each node in the network will be in the $CDS_selector$ set of only one of its neighbors.

Following the Selection Rule, we are concerned to establish another rule to complete the pruning process in CDS_1 , so in the next a cancellation rule is defined.

Cancellation Rule: For a node $u \in CDS_1$, assume that nodes $i, j \in N_1(u)$, select respectively, pairs (v_i, w_i) and (v_j, w_j) using Selection Rule. Then, node $u \in CDS_1$ with $CDS_selector(u) = \emptyset$ is omitted from the list of CDS , if at least one of the following five conditions holds for every $i, j \in N_1(u)$:

$$v_i = v_j. \quad (1)$$

$$v_i = w_j. \quad (2)$$

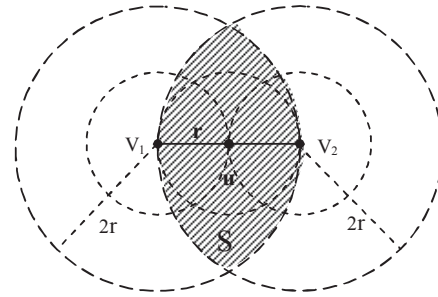


Figure 1. Shaded area, namely S , is the intersection of 2-hop neighborhoods of nodes v_1 and v_2 .

$$v_i \in N_1(j). \quad (3)$$

$$w_i = w_j. \quad (4)$$

$$w_i \in N_1(j). \quad (5)$$

The equal sign "=" in Cancellation Rule means that the node identifications i.e. their IDs, are identical.

A result of Cancellation Rule is that if a node selects a pair (v, w) via Selection Rule, then node v will not be omitted from CDS structure and will stay in CDS_2 because its selector set is not empty. So every node of the network either belongs to CDS_2 or has at least one neighbor that belongs to CDS_2 .

Formally, in the following statement, we will prove that the resulting dominating set i.e. CDS_2 , after applying Cancellation Rule on the primary CDS , remains connected.

Theorem: By using Cancellation Rule and pruning the primary connected dominating set, CDS_1 , we get a connected dominating set, CDS_2 .

Proof of the Theorem: Suppose the nodes u, v, w (regardless of their indices) are members of CDS_1 . Moreover, in the following graphs, nodes of CDS_2 , as a subset of CDS_1 , are shown by a solid point surrounded by a circle. Selected pair of a node, called x , is shown as (v_x, w_x) in the proof and following graphs e.g. selected pair of v_i is (v_{v_i}, w_{v_i}) .

Suppose $CDS_2 \subseteq CDS_1$ has been calculated by applying Cancellation Rule on CDS_1 . To prove this theorem, we use Cancellation Rule and show that if a node $u \in CDS_1$ is excluded from CDS_2 via Cancellation Rule, then there exists a route between every two nodes of its neighbors, which passes along a subset of nodes in $CDS_2 \subseteq CDS_1$, and doesn't pass the node u .

Consider two nodes $i, j \in N_1(u)$ that have selected pairs (v_i, w_i) and (v_j, w_j) , respectively, according to Selection Rule. Note that nodes i and j may be contained in CDS_1 . If any one of Conditions 1-3, stated in Cancellation rule, hold since the nodes v_i and v_j are selected as members of CDS_2 (because $CDS_selector(v_i) \neq \emptyset$ and $CDS_selector(v_j) \neq \emptyset$ e.g. $i \in CDS_selector(v_i)$ and $j \in CDS_selector(v_j)$, according to Selection Rule), there is a route between nodes i and j as shown in Fig. 2 such that all intermediate nodes are in CDS_2 and doesn't include node u .

If the nodes i and j satisfy Conditions 4 or 5, meaning that $w_i = w_j = w_1$ or $w_1 = w_i \in N_1(j)$, and the node w_1 is in CDS_2 , as shown in Fig. 3a and Fig. 4a, then there are routes (i, v_i, w_1, v_j, j) ,

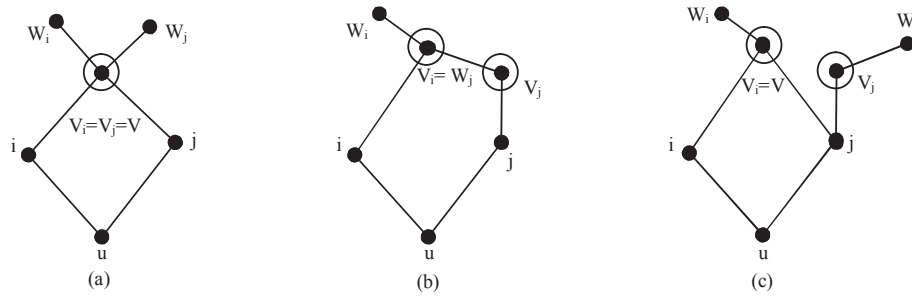


Figure 2. Omission of node u from CDS with conditions 1-3 (a) if condition (1) holds then path (i, v_j) exists between i and j . (b) if condition (2) holds then path (i, v_i, v_j, j) exists between i and j . (c) if condition (3) holds then path (i, v_i, j) exists between i and j .

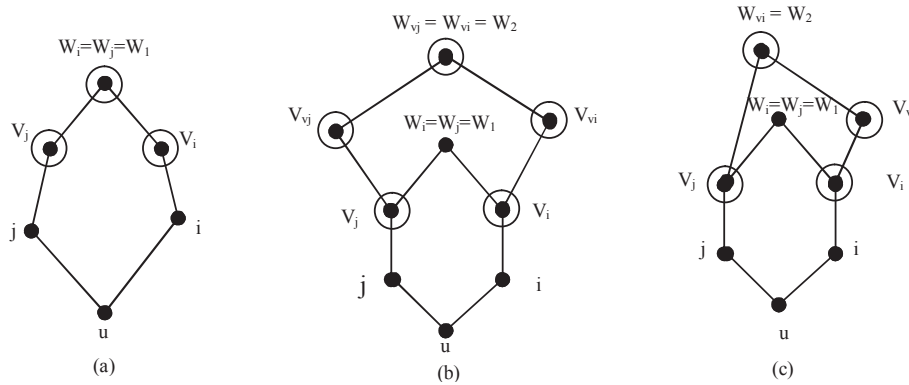


Figure 3. Condition 4 holds for some neighbors of node u : (a) condition 4 is true for i and j , W_1 remains in CDS_2 and path (i, V_i, W_1, V_j, j) exists. (b) W_1 is omitted from CDS_2 , condition 4 is true for V_i and V_j i.e. $W_{v_j} = W_{v_i} = W_2$ remains in CDS_2 and path $(i, V_i, V_{v_i}, W_2, V_{v_j}, V_{j,j})$ exists. (c) W_1 is omitted from CDS_2 , condition 5 is true for V_i and V_j i.e. $W_{v_i} \in N_1(V_j)$, W_2 remains in CDS_2 and path $(i, V_i, W_2, V_{v_i}, V_{j,j})$ exists.

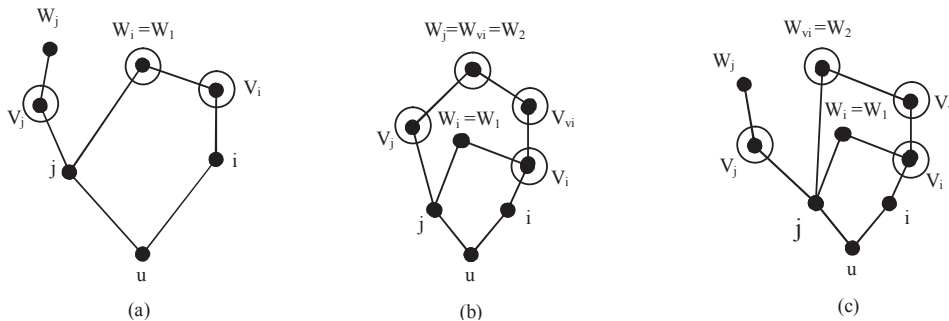


Figure 4. Condition 5 holds for some neighbors of node u (a) condition 5 is true for i and j , W_1 remains in CDS_2 and path (i, V_i, W_1, j) exists. (b) w_1 is omitted from CDS_2 , condition 4 is true again for V_i and j i.e. $W_j = W_{v_i}$, W_2 remains in CDS_2 and path $(i, V_i, V_{v_i}, W_2, V_{j,j})$ exists. (c) W_1 is omitted from CDS_2 , condition 5 is true again for V_i and j i.e. $W_{v_i} \in N_1(j)$, W_2 remains in CDS_2 and path $(i, V_i, V_{v_i}, W_2, j)$ exists.

and (i, v_i, w_1, j) respectively, between i and j , which exclude u and all intermediate nodes are in CDS_2 . But, if w_1 is excluded from CDS_2 via Cancellation Rule, every two neighbors of w_1 definitely satisfy at least one of the conditions of Cancellation Rule and $CDS\text{-}selector(w_1)$ must be empty; suppose that conditions 1-3 hold for the neighbors of w_1 as well as nodes v_j and v_i in Fig. 3a, or j and v_i in Fig. 4a. This implies that since $v_i, v_{v_i}, v_{v_j}, v_j \in CDS_2$ (because their selector sets are not empty), similar to the previous argument, there must be a route

between v_i and v_j in Fig. 3a, and between v_i and j in Fig. 4a, such that these routes don't pass the node w_1 and intermediate nodes are in CDS_2 . In short, these routes are denoted, respectively, by $v_i \leftrightarrow v_j$ and $v_i \leftrightarrow j$. These two routes are not shown separately because they are similar to the cases of Fig. 2. Using this notation, the route between i and j is represented by $i \leftrightarrow v_i \leftrightarrow v_j \leftrightarrow j$, in case of Fig. 3a, and $i \leftrightarrow v_i \leftrightarrow j$, in case of Fig. 4a.

On the other hand suppose w_1 is excluded from CDS_2 subject to conditions 4 or 5, for two of its neighbors as well as v_i and v_j in Fig. 3a, or j and v_i in Fig. 4a. Then there exists a node w_2 , as shown in Fig. 3b, Fig. 3c, Fig. 4b or Fig. 4c, in such a way that it is selected based on Selection Rule where the relation $u > w_1 > w_2$ (see Definition 2) emerges; because w_2 must be smaller than all neighbors within 2-hop neighborhood of v_i and v_j in Fig. 3 or j and v_i in Fig. 4b as it is selected by them via Selection Rule and also w_1 must be smaller than all nodes within 2-hop neighborhood of node i because it was selected by it according to Selection Rule. Similarly, if w_2 is in CDS_2 (as in Fig. 3b, Fig. 3c, Fig. 4b and Fig. 4c), or w_2 is excluded from CDS_2 subject to Conditions 1-3, then there exists a route between i and j such that all nodes along this route are in CDS_2 . But, if w_2 is not included in CDS_2 subject to conditions 4 or 5 then similarly there is a node, w_3 , such that with the use of Selection Rule the following relation holds, i.e., $u > w_1 > w_2 > w_3$ (see Definition 2).

In general, for the k -th step, if w_k is in CDS_2 (as in Fig. 3b, Fig. 3c, Fig. 4b and Fig. 4c for $k=2$), or w_k is excluded from CDS_2 subject to Conditions 1-3, similar to the previous discussion, there exists a route between i and j that passes along nodes of CDS_2 . Moreover, if w_k is not included in CDS_2 subject to Conditions 4 or 5, there will be a sequence with the ordering relation, $u > w_1 > w_2 > w_3 > \dots > w_k$ (see Definition 2). K is finite since the number of nodes of network is limited. Therefore, there is the smallest node, w_n , in the ordering which is selected via Selection Rule and $CDS-selector(w_n) \neq \emptyset$. Consequently, w_n will remain in CDS_2 because its selector set is not empty and ultimately there is a route between i and j that passes along nodes of CDS_2 . So it is shown that if a node of CDS_1 is excluded from CDS_2 via Cancellation Rule, then there must be a route between every two nodes of its neighbors such that passes through nodes of CDS_2 . According to this result and since CDS_1 is connected and every node of the network has at least one neighbor that belongs to CDS_2 (according to Selection and Cancellation Rules), newly constructed structure i.e. CDS_2 , is a connected dominating set as it is required. ■

It must be pointed out that the CDS forwarding rule is as follows:

- **CDS forwarding rule:** a node in the CDS rebroadcasts a packet once, if it is received from a neighbor node that belongs to the CDS.

V. THE MESSAGE LENGTH OF THE PRUNING ALGORITHM

The WP algorithm, which is performed in the first step to form CDS_1 , requires that each node carries some information such as the address of 2-hop and 1-hop neighbors, and also the MPR set of its 1-hop neighbors. Therefore, each node puts the addresses of its 1-hop neighbors with an additional bit for each neighbor, indicating the one which is selected as MPR in the "HELLO" message. In addition, for our pruning algorithm it is necessary that each node knows about addresses and degrees of its 1-hop and 2-hop neighbors which are in CDS_1 . So every node uses a single bit for itself and one bit for every 1-hop neighbor in the "HELLO" message to show whether they are in the primary CDS, i.e. CDS_1 , or not. Moreover, each node transmits addresses of nodes in its selected pair to its neighbors via "HELLO" message.

Let m denote the maximum degree of a node. As a consequence a node broadcasts a total of $m + 2$ addresses for its 1-hop neighbors and its selected pair, m values of degrees of its 1-hop neighbors, $2m$ bits to show which neighbors are in CDS_1 and its MPR set and a single bit to show whether the node itself belongs to CDS_1 or not. Summation of length of all these data results in a scalar coefficient of m . So the size of "HELLO" messages will be $O(m)$.

VI. SIMULATION RESULTS AND ALGORITHM PERFORMANCE EVALUATION

By simulation, we compare our algorithm with those of WP and WC explained in Section I. The simulation is performed by the simulator NS, with the assumption that there is an ideal network in a sense that no collision happens and mobility of nodes is not considered. However, this simplification has no effect on the ultimate result of our method.

For the first scenario as shown in Fig. 5 and Fig. 6, a certain number of nodes ranging from 20 to 100 (with 10-step spacing) and 100 to 1000 (with 100-step spacing) are uniformly distributed in a plane. The area of plane is 100×100 square meters. The broadcast range for each node is r meters, and the simulation is done for $r=25$ and $r=50$. For the simulation, we use only connected networks and for each fixed number of nodes sufficient number of experiments is averaged to make 90% confidence interval within $\pm 5\%$.

Fig. 5a shows the result when the number of nodes varies between 20 and 100, i.e. a sparse network, and Fig. 5b depicts the results when the number of nodes changes between 100 and 1000, i.e. a dense network.

In Fig. 5 ascension in values of three curves is seen. This is because with the increase of number of nodes in the network, each node attempts to select more forwarders to cover new nodes. As seen, the pruning algorithm has a better performance either for sparse networks (Fig. 5a) or dense networks (Fig. 5b) comparing to WP algorithm. However, our algorithm behaves better than WC algorithm for sparse networks (Fig. 5a) but for

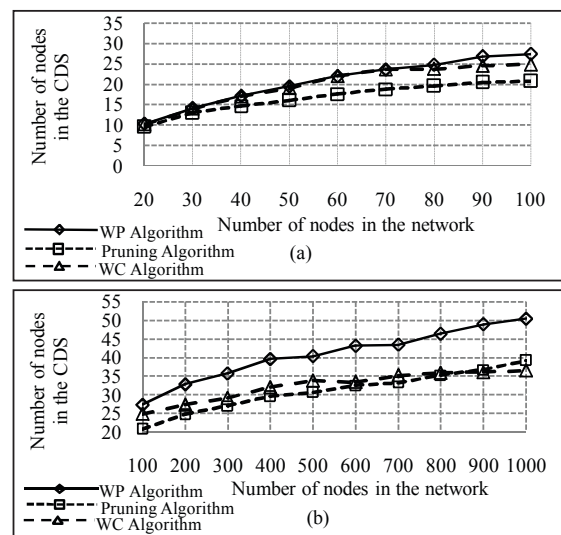


Figure 5. Simulation results with $r=25$ meters: (a) results for nodes from 20 to 100 (b) results for nodes from 100 to 1000.

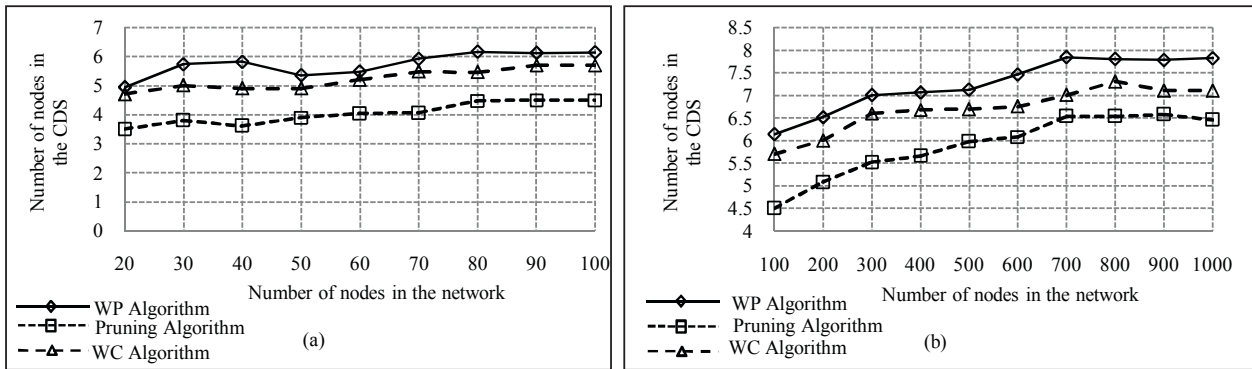


Figure 6. Simulation results with $r=50$ meters: (a) results for nodes from 20 to 100 (b) results for nodes from 100 to 1000.

dense networks, shown in Fig. 5b, it behaves approximately like WC algorithm. The simulation shows that, for the networks with more than 600 nodes, the difference between cardinality of CDSs generated by WC algorithm and our algorithm is less than 2, while our pruning algorithm, for $r=25$, reduces in average 19.7 percentage of the cardinality of CDS generated by WP algorithm.

Fig. 6 shows that, for $r=50$, our pruning algorithm has much better performance than WP algorithm. Fig. 6a and Fig. 6b show, respectively, the result of the simulation when the number of nodes ranging from 20 to 100, and when the number of nodes is varying between 100 and 1000 (dense networks). For the last case in Fig. 6, however proposed method has better performance, but it is observed that the difference between proposed algorithm and WC algorithm is less than 2. The reason is that, by increasing broadcast range of nodes, the area covered by each node is increased and as a result, the diameter of the network decreases, hence the size of CDS is decreased and difference becomes less.

In the second scenario, a certain number of nodes (first 200 nodes and then 1000 nodes) are uniformly distributed in a plane. A number of simulations for the broadcast ranges from 20 to 75 meters are implemented. For a fixed broadcast range, we have done sufficient number of experiments and averaged the results.

Fig. 7 shows the ratio of the forwarder nodes to the network nodes, versus the broadcast range, when the number of 200 and 1000 nodes are selected.

As shown in Fig. 7, by increasing the broadcast range, the ratio decreases and all three algorithms behave similarly. Because, by increasing the broadcast range, a node can cover a larger area, hence less number of nodes are needed to cover whole the area. Again, it is observed that, with different broadcast ranges, our algorithm is giving a better performance than WP algorithm. Also we see that when a dense network is simulated, i.e. the broadcast range is more than 45 meters in Fig. 7a or the number of nodes in the network is high in Fig. 7b, then both WC algorithm and our algorithm follow almost the same performance. Furthermore, for the sparse networks, i.e. the broadcast range is less than 45 meters in Fig. 7a or the number of nodes of the network is small (as

shown in Fig. 5a or Fig. 6a), our algorithm works better than WC algorithm.

Despite the fact that the pruning algorithm and WC algorithm have similar results in some situations, the length of broadcast messages in pruning method i.e. $O(m)$ as discussed in section V, is less than that of WC algorithm i.e. $O(m^2)$. So pruning method is more efficient than it.

VII. CONCLUSION

In this work, a new method, based on a local source independent algorithm and a pruning algorithm, to form CDS in a network, has been presented. The pruning algorithm can be used to prune any source independent CDS structure based on partial 2-hop information. We have shown that the size of broadcasted messages in the proposed method is $O(m)$. We also showed in simulation that our algorithm, based on the pruning method, in comparison to WP algorithm, reduces size of the CDS significantly, and has a better performance.

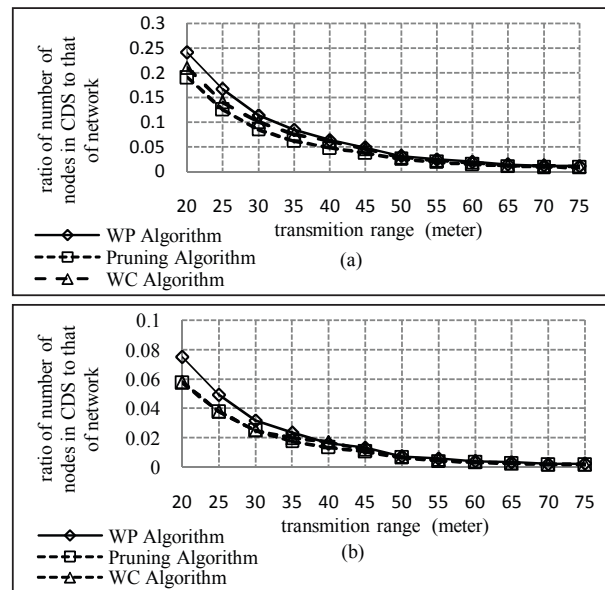


Figure 7. Simulation results for different transmission ranges: (a) 200 nodes in the network (b) 1000 nodes in the network.

Moreover, we have compared our algorithm to WC algorithm which uses complete 2-hop information of neighborhoods. We saw that our algorithm, for dense networks, has almost the same performance as the WC algorithm, but in a sparse network our algorithm works better. On the other hand, in WC algorithm, the size of the broadcasted messages is $O(m^2)$, as discussed in section I, while the message size of our algorithm is $O(m)$, therefore our algorithm has a better performance than WC algorithm.

References

- [1] V. Chvatal, "A Greedy Heuristic for the Set-Covering Problem," *Math. Operation research*, vol. 4, no. 3, pp. 233-235, 1979.
- [2] A. Qayyum, L. Viennot, and A. Laouti, "Multipoint Relaying for Flooding Broadcast message in Mobile Wireless Networks," *Proc. 35th Hawaii Int'l Conf. System Sciences (HICSS-35)*, pp. 3898-3907, Jan. 2002.
- [3] C. Adjih, P. Jacquet, and L. Viennot, "Computing Connected Dominated Sets with Multipoint Relays," *Ad Hoc & Sensor Networks* vol. 1, pp. 27-39, May 2005.
- [4] J. Wu, "An Enhanced Approach to Determine a Small Forward Node Set Based on Multipoint Relay," *Proc. IEEE Semi-Ann. Vehicular Technology Conf. Fall (VTC 2003)*, Sept. 2003.
- [5] J. Wu, W. Lou and F. Dai, "Extended Multipoint Relays to Determine Connected Dominating Sets in MANETs," *IEEE Transactions on Computers*, vol. 55, no. 3, pp. 334-347, Mar. 2006.
- [6] M.V. Marathe, H. Brey, H.B. HuntIII, S.S. Ravi, and D.J. Rosenkrantz, "Simple Heuristics for Unit Disk Graphs," *Networks*, vol. 25, pp. 59-68, 1995.
- [7] B. Das, R. Sivakumar, and V. Bharghavan, "Routing in Ad-Hoc Networks Using a Spine," *Proc. Sixth Int'l Conf. Computer Comm. and Networks (ICCCN '97)*, pp. 1-20, Sept. 1997.
- [8] S. Guha and S. Khuller, "Approximation Algorithms for Connected Dominating Sets," *Algorithmica*, vol. 20, no. 4, pp. 374-387, 1998.
- [9] P.J. Wan, K. Alzoubi, and O. Frieder, "Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks," *Proc. IEEE INFOCOM 2002*, vol. 3, pp. 1597-1604, June 2002.
- [10] J. Wu and W. Lou, "Forward-Node-Set-Based Broadcast in Clustered Mobile Ad Hoc Networks," *Wireless Networks and Mobile Computing*, special issue on algorithmic, geometric, graph, combinatorial, and vector aspects, vol. 3, no. 2, pp. 155-173, 2003.
- [11] I. Stojmenovic, S. Seddigh, and J. Zunic, "Dominating Sets and Neighbor Elimination Based Broadcasting Algorithms in Wireless Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 13, no. 1, pp. 14-25, Jan. 2002.
- [12] 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.
- [13] P. Sinha, R. Sivakumar, and V. Bharghavan, "Enhancing Ad Hoc Routing with Dynamic Virtual Infrastructures," *Proc. IEEE INFOCOM 2001*, vol. 3, pp. 1763-1772, Apr. 2001.
- [14] H. Lim and C. Kim, "Flooding in Wireless Ad Hoc Networks," *Computer Comm. J.*, vol. 24, nos. 3-4, pp. 353-363, 2001.
- [15] W. Peng and X.C. Lu, "On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Networks," *Proc. First Ann. Workshop Mobile and Ad Hoc Networking and Computing (MOBIHOC 2000)*, pp. 129-130, Aug. 2000.
- [16] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Proc. of ACM/IEEE MOBICOM'99*, pp. 151-162, Aug. 1999.
- [17] J. Wu and F. Dai, "A generic distributed broadcast scheme in ad hoc wireless networks," *Proc. of ICDCS'2003*, pp. 460-468, May 2003.