Construction of Connected Dominating Set to Reduce Contention in Wireless Ad-hoc Network

Afia Anjum
Military Instituite of Science and Technology
Dhaka, Bangladesh
afia.anjum70@gmail.com

Tahrima Oannahary
Military Instituite of Science and Technology
Dhaka, Bangladesh
totusi96@gmail.com

Diana Shahrin
Military Instituite of Science and Technology
Dhaka, Bangladesh
diana.shahrin@gmail.com

Chowdhury Nawrin Ferdous
Military Instituite of Science and Technology
Dhaka, Bangladesh
bipashachowdhury03@gmail.com

ABSTRACT

A wireless ad-hoc network is a combination of nodes, connected in a wireless manner and has freedom of random movement, which spontaneously arranges its components with purpose under proper conditions. The two ways for communication in wireless multi-hop network are Broadcast and Multi-cast. If flooding is used for broadcasting message, *broadcast storm problem* arises which results in contention, collision and redundancy. One of the ways to avoid the problems is creating a virtual backbone consisting some nodes of the network which will cover all the nodes eventually. This is called the Connected Dominating Set (CDS). There have been some works in order to reduce the number of rebroadcasting but very few of them are designed to minimize contention in a network. In this paper, two algorithms called Centralized Contention Reduced CDS (Centralized CRCDS) and Distributed Contention Reduced CDS (Distributed CRCDS) are proposed which will select nodes in such a way that will reduce contention in centralized and distributed network respectively. These algorithms have improved performance than the previous ones regarding number of forwarding nodes and contentions in a large, dense network. Proposed algorithms reduce almost from 7.4% to 89.95% contention in different scenarios.

CCS CONCEPTS

• **Networks → Network architectures**; **Network algorithms**; **Network performance analysis**; *Network design principles*; *Packet scheduling*; Wireless access points, base stations and infrastructure;

KEYWORDS

broadcast, communication, mobile ad-hoc network (MANET), connected dominating set (CDS), minimum connected dominating set (MCDS), contention

6th NSysS 2019, December 17–19, 2019, Dhaka, Bangladesh

ACM ISBN 978-1-4503-7699-0/19/12…$15.00
https://doi.org/10.1145/3362966.3362975

ACM Reference Format:
Afia Anjum, Tahrima Oannahary, Diana Shahrin, and Chowdhury Nawrin Ferdous. 2019. Construction of Connected Dominating Set to Reduce Contention in Wireless Ad-hoc Network. In *6th International Conference on Networking, Systems and Security (6th NSysS 2019), December 17–19, 2019, Dhaka, Bangladesh.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3362966.3362975

1 INTRODUCTION

A well known technique for communication in a wireless ad-hoc network is broadcasting which is an information dissemination process of sending a message from a source node to all other nodes of the network. Wireless ad-hoc networks need to broadcast messages for various services such as route discovery, erasing an invalid route, locating a node or even for sending alarm signals in the entire network.

Flooding is an important communication primitive in mobile ad-hoc networks. In flooding, a node which receives a message for the first time, rebroadcasts the message only one time which eventually covers all nodes [13]. If a message is broadcasted using flooding, it requires each node to forward message which leads to *broadcast storm problem* [9].

When a node broadcasts a message to all of its neighbors, but all the neighbors have previously received the message from another nodes, makes the message a redundant and unnecessary one.

When a node broadcasts a message and some of its neighbors rebroadcast the message to spread it in the whole network and multiple nodes within the same transmission region try broadcasting the message at the same period of time, then their transmission will conflict with each other creating contention.

The reason behind collisions are insufficiency of back-off mechanism, the absence of collision detection, the inadequacy of RTS/CTS dialogue etc. In a CSMA/CA network, these are some disadvantages of flooding [13].

Blind flooding, an effortless approach to perform broadcast, ensures full coverage at high mobility but causes high energy consumption and *broadcast storm problem.* Some algorithms can be used to solve the broadcast storm problem e.g. Connected Dominating Set (CDS) [3], Minimum Connected Dominating Set (MCDS) [14].

The main objective of this work is to reduce contention that occurs in flooding of message among nodes. Contention is one of the broadcast storm problems which refers to *Competition for Resources*. Figure 1 is showing possible contention area in a network. When more than one nodes within the same transmission area try to broadcast a message at the same time, this problem arises. In a wireless ad-hoc network after broadcasting a message by a mobile host, if many of its neighbors want to rebroadcast it, these transmissions (which are all from nearby hosts) may face serious contention with each other. This will lead to an unstable network. Our aim is to minimize contention in global topology network and also in distributed network. One solution to solve this problem is to compute a virtual backbone based on the physical topology and run any existing routing protocol over the virtual backbone [3]. We will be using the concept of Connected Dominating Set and Minimum Connected Dominating Set [14] for this context.

A lot of research works have been done on reducing the number of forwarding packets using the concept of CDS and MCDS in centralized and distributed networks and many algorithms have been proposed. Most of the works aim at minimizing redundancy in network while one of them aim at minimizing contention [4]. Our main contribution is to construct a connected dominating set that will select the forwarding nodes in such a way that the contention among the forwarding nodes is minimized, while reducing the redundancy. We have provided two algorithms which focus on connected dominating set (CDS) to minimize contention, using global topology information of the network as well as using 2-hop neighborhood information of each node. We have presented a comprehensive simulation to analyze the behavior of the proposed algorithms and compare their performances with other state-of-the-art algorithms in terms of number of forwarding nodes and number of contention. The rest of the paper is organized as follows. Section II discusses the related researches previously done in this area. Section III presents the algorithms followed by the performance analysis of the proposed algorithms with state-of-the-art algorithms in section IV. Finally, section V concludes the article.

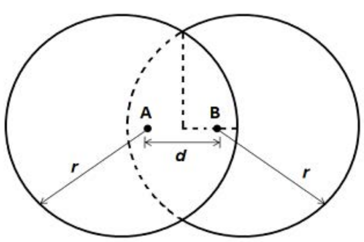

Figure 1: A and B sharing common broadcasting area

2 LITERATURE REVIEW

A significant number of research has been carried on related to reducing broadcast storm problems in both global topology and distributed networks. This section briefly discusses those related papers.

2.1 Global Topology Network Algorithms

Guha and Khuller [5] proposed two approximate CDS construction strategies, both for solving the redundancy problem in the network. The first strategy constructs a greedy algorithm. It begins by coloring all vertices white. Choosing the node with maximum white neighbors, it colors the node black and colors all its white neighbors grey. This process continues until all the nodes are either grey or black. Eventually the black nodes construct the final MCDS. The second algorithm is an improvement of the first algorithm. The algorithm finds a dominating set in the first phase, and in the second phase connects the dominating set.

Ruan et al. [12] gave a one-step greedy approximation algorithm with performance ratio at most $3+\ln(\Delta)$ to construct CDS. It requires each node to be colored white at the very beginning. If there exists a white or grey node such that coloring it black and its white neighbors grey would reduce the potential function, which is stated in [12], then choose the one that causes maximum reduction in the potential function.

Sergiy et al. [2] proposed an algorithm for constructing MCDS. This algorithm is based on two types of vetices, fixed vertex and non-fixed vertex, definition of which is stated in [2]. At the end of the algorithm, only fixed vertices will be part of the final dominating set.

A limited number of papers was conducted focusing on reducing the number of contentions occurred. Chowdhury Nawrin Ferdous [4] discussed about a centralized algorithm for constructing CDS with contention as minimum as possible. In this algorithm, a concept of *Candidate_Set* has been introduced which is well stated in [4]. All the nodes are white initially. The concept of white, grey and black nodes are also here like [5]. But to minimize contention, the major modification is done using *Candidate_Set* while choosing the forwarding nodes among the grey nodes. As the black nodes are already in CDS, choosing a grey node with minimum black neighbors reduces the chance of contention.

2.2 Distributed Network Algorithms

Lim and Kim [6] proposed to show that the minimum cost flooding tree problem is similar to MCDS (Minimum Connected Dominating Set) problem and proved the NP-completeness of the minimum cost flooding tree problem. They proposed two flooding methods called *self pruning* and *dominant pruning*. In self pruning, each node needs to be aware of its one-hop neighbors, whereas dominant pruning is based on 2-hop neighbor information.

Rab et al. [10] presented a modification of the self pruning algorithm named Improved Self Pruning. This algorithm completes the broadcast with smaller packet header but uses extended neighbor knowledge that includes 3-hop neighbor information which gives a better knowledge to decide whether to forward or not.

W. Lou and J. Wu in [7] proposed an approach called PDP (Partial Dominant Pruning) which appears to make the most efficient use of neighborhood information.

Ashikur Rahman et al. [11] developed an algorithm that enhances the efficiency of PDP. This algorithm applies intentional delay on every node before its transmission. If a node v receives a packet m from node u, it constructs P, U, B lists of v and updates them if necessary after the timer goes off as stated in [11]. This approach shows better performance than both dominant pruning and PDP.

Chowdhury Nawrin Ferdous [4] proposed another algorithm for distributed network that minimizes contention. Here, each node uses its 2-hop neighborhood information for constructing the forward list. The core difference between the distributed-CACDS and the dominant pruning algorithm is that, distributed-CACDS selects the adjacent forwarding nodes to keep the contention minimum while rebroadcasting the packet.

In sum, the previous studies suggest that different approaches were proposed to reduce the redundancy problem by constructing MCDS. It is certain that not much but only one approach was proposed in the paper [4] of Chowdhury Nawrin Ferdous to reduce the number of contention. But the limitation of the paper [4] is that it increases number of forwarding nodes. Thus, in this paper, we aim to propose two approaches which will reduce the contention as well as try not to increase the number of forwarding nodes.

3 THE ALGORITHMS

In this section, we describe our algorithms for the contention problem, keeping redundancy problem in mind. To solve the redundancy problem, we could not simply construct Minimum Spanning Tree (MST) or Broadcast Tree, due to some valid reasons, such as: in MST, it contains all the nodes of the network in it. Considering all the nodes as a forwarding node will never be able to solve the redundancy problem. On the other hand, if the power consumed by each node is not adjustable in a network, minimizing the total power consumed by a Broadcast Tree is equivalent to MCDS [1]. So we need to minimize the power consumption if we construct a Broadcast Tree. Keeping these drawbacks in mind, it can be said that building a Minimum Connected Dominating Set (MCDS) would be the best solution here. MCDS is the Connected Dominating Set (CDS) of graph G which has the smallest possible cardinality among all CDSs of graph G. To become a CDS of a graph G, the set D needs to have two properties. Firstly, any node in D should be able to reach any other node in D by a path that stays entirely within D. Secondly, every vertex in G should either belong to D or be adjacent to a vertex in D. As we have already mentioned, most of the existing heuristics are for the MCDS problem. So, we proceed our two algorithms in such a way that they reduce contention in global topology network and distributed network respectively. To have a contention free CDS, no more than one node within the same transmission area should broadcast a message at the same time.

3.1 Centralized Contention Reduced CDS

Our first algorithm is an algorithm for the minimum connected dominating set problem in a network where each node is assumed to have information about every other node in the network. At the beginning of the algorithm, all the nodes present in the network are considered to be white. Let us consider node u, where u is the node with the minimum degree. First, we delete u temporarily and see if the graph becomes disconnected or not. If the graph becomes disconnected, then we add u again and make it black (fixed). otherwise we see if u has any black neighbor or not.

(1) If no black neighbor, then choose the neighbor v with maximum white neighbors and delete u. Make v black (fixed).
(2) If there are black neighbors, count how many.
    - If only one black neighbor, then check if any other neighbor of u is connected to that black neighbor or not.
        - If connected, then delete u;
        - If it is not connected and also if there is no white(non-fixed) vertices left in graph G or if it is not connected and also if u is degree 1 vertex, then delete u permanently;
        - Otherwise add u in graph G and make it black.
    - If more than one neighbors, then check if the neighbors are connected to each other. If connected, then there may be contention. So delete u. Else for each black neighbors of u check if any other neighbor of u is connected to that black neighbor or not.
        - If it is true for all the black neighbors, then delete u;
        - If it is not true for all the black neighbors and also if there is no non fixed vertices left in graph G or if it is not true for all the black neighbors and also if u is degree 1 vertex, then delete u permanently;
        - Otherwise add u and make it black.

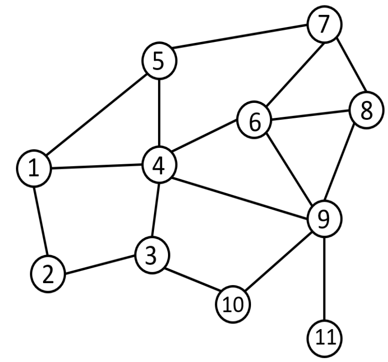

Figure 2: A random example scenario consisting 11 nodes

The details of this algorithm is stated in Algorithm 1.

Let us explain Algorithm 1 with a sample network of 11 nodes which is shown in Figure 2. The one hop and two hop neighbor list is given in the Table 1.

Table 1: 1-hop and 2-hop neighbors of each node of the scenario in Figure 2

v	*N(v)*	*N(N(v))*
1	1, 2, 4, 5	1, 2, 3, 4, 5, 6, 7
2	1, 2, 3	1, 2, 3, 4, 5, 10
3	2,3, 4, 10	1, 2, 3, 4, 5, 6, 9, 10
4	1, 3, 4, 5, 6, 9	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
5	1, 4, 5, 6, 7	1, 2, 3, 4, 5, 6, 7, 8, 9
6	4, 5, 6, 7, 8, 9	1, 3, 4, 5, 6, 7, 8, 9, 10, 11
7	5, 6, 7, 8	1, 4, 5, 6, 7, 8, 9
8	6, 7, 8, 9	4, 5, 6, 7, 8, 9, 10, 11
9	4, 6, 8, 9, 10, 11	1, 3, 4, 5, 6, 7, 8, 9, 10, 11
10	3, 9, 10	2, 3, 4, 6, 8, 9, 10
11	9, 11	4, 6, 8, 9, 10, 11

The steps of the proposed algorithm, how it reduces contention is expressed in detailed manner in Table 2. According to the algorithm, node 11 will be chosen as it has the minimum number of white neighbor, which is 1. It will be removed as it does not disconnect the

Algorithm 1 Centralized CRCDS Pseudocode

```
/*D is the current CDS; F is the set of fixed vertices; D1 is the set
of vertices that have degree 1 in G*/
D ← V
F ← ∅
while D still has a non-fixed neighbor do
    if D1 is not empty then
        u ← {w|w ∈ D1\F}
    else
        u ← {v|v has minimum white neighbor and v ∈ D\F}
    end if
    if graph G is not connected after removing vertex u then
        F ← F ∪ {u}
    else
        if u has no fixed vertices as its neighbor then
            x ← {v|v has maximum white neighbor and v ∈ N(u)}
            F ← F ∪ {x}
            if D1 is not empty then
                remove u from D1 and D
            else
                remove u from D
            end if
        else
            t ← total number of fixed neighbors of u
            if t = 1 then
                b ← black neighbor of u
                if any other neighbor of u is connected to b or u
                is the last non fixed vertex in G or u has degree 1
                then
                    if D1 ≠ ∅ then
                        remove u from D1 and D
                    else
                        remove u from D
                    end if
                else
                    F ← F ∪ {u}
                end if
            else
                B ← set of black neighbors of u
                if s ∈ B is connected to any other vertex of B
                then
                    if D1 ≠ ∅ then
                        remove u from D1 and D
                    else
                        remove u from D
                    end if
                else
                    count ← 0
                    for all s ∈ B do
                        if u has a neighbor connected to s then
                            count + +
                        end if
                    end for
                    if count=s or u is the last non fixed vertex in
                    G or u has degree 1 then
                        if D1 ≠ ∅ then
                            remove u from D1 and
                            D
                        else
                            remove u from D
                            d(p) ← d(p) − 1
                        end if
                    else
                        F ← F ∩ {u}
                    end if
                end if
            end if
        end if
    end if
end while
Return D
```

network and node 9 will be made black and the white neighbor list of each node will be updated. Next node 10 is chosen as it has only 1 white neighbor. Node 10 is made black as none of its neighbors are connected to any black node and the white neighbor list is updated again. Now node 2 is chosen. It is removed as it does not make the graph disconnected and node 1 is made black as node 2 has no black neighbor and the white neighbor list is updated each time a node is made black. Now node 3 is chosen and it is made black because its neighbor is not connected to any black node. Then node 4 is chosen and it is made black for the previous reason. Then node 5 is chosen and deleted. Then node 6 is chosen and deleted. Node 7 is picked next and deleted and then node 8 is made fixed as Node 7 has no black neighbors. So we get the final CDS which is node 1, 3, 4, 8, 9, 10.

These forwarding nodes will not create any contention when they will broadcast the message.

In the following theorems we show that the algorithm outputs a CDS correctly.

Theorem 3.1.1 *Contention Reduced CDS is a connected dominating set.*

Proof: We show by induction that the returned set D is a connected dominating set. Since the graph is connected, so it is true at the beginning. Thus, D = V is a CDS, where V is the set of all vertices in the network. At each step, we look for a vertex which has the minimum degree and delete it only if its removal does not make D disconnected. It is also made sure by the algorithm that for each removed vertex u, there is at least one neighbor $v \epsilon N(u)$ which is fixed. Thus, for each vertex u which is not in D, there will be at least one adjacent vertex of u which is included in the final set of D. This proves that D is a connected dominating set. ∎

Theorem 3.1.2 *The connected dominating set constructed by Contention reduced CDS covers all the nodes in the network.*

Proof: Let us assume that, U is the set consisting all the vertices in the network and $U = \{x_1, x_2, ...x_n\}$, where n is the number of vertices in the network. Let, V is the set of vertices which are in the final CDS and $V = \{v_1, v_1, v_3...v_m\}$, where m is the total number of vertices in the final CDS. The set N represents the set that consists all the adjacent nodes of v_i ,where v_i belongs to CDS set and $N = N(v_1) \cup N(v_2) \cup N(v_3) \cup ...N(v_m)$.

Table 2: Detailed analysis of the Centralized Contention Reduced CDS (Centralized CRDS) in scenario in Figure 2

Chosen Node	Number of White Nodes	Removal Disconnects Network?	State	Remarks
11	1	No	Removed	**Make node 9 black** as node 11 has no black neighbor
10	1	No	Black	As its neighbor is not connected to any black neighbor
2	2	No	Removed	**Make node 1 black** as node 2 has no black neighbor
3	1	No	Black	As its neighbor is not connected to any black neighbor
4	2	No	Black	As its neighbor is not connected to any black neighbor
5	2	No	Removed	
6	1	No	Removed	
7	1	No	Removed	**Make node 8 black** as node 7 has no black neighbor
The Forwarding nodes are 1, 3, 4, 8, 9, 10				

In order to prove, Contention Reduced CDS covers all the nodes in the network, we have to prove that U = N.
Contention Reduced CDS is a connected dominating set. So every node $x_i \epsilon U$ is either same as v_j or is adjacent to v_j for some j , where v_j belongs to CDS set. In other words,

$$\forall_i [\exists_j [x_i \epsilon N(v_j)]] \tag{1}$$

$$\{x_1, x_2, ...x_n\} \subseteq N \tag{2}$$

By definition,

$$x_1, x_2, ...x_n \subseteq U \tag{3}$$

From Equation 3.2 and 3.3, by the axiom of extensionality, we can say that,

$$\forall x_i (x_i \epsilon U \leftrightarrow x_i \epsilon N) \rightarrow U = N \tag{4}$$

■

3.2 Distributed Contention Reduced CDS

Centralized wireless network is not free from problems. It needs the existence of a central authority, which may limit the scalability and flexibility in mobile environments. Whereas, distributed wireless networks provide several advantages over centralized ones for mobile environments. Communication here does not depend on a central authority, rather it takes place in a multi-hop manner. Self-organization and self-construction is also possible here, which makes its maintenance and expansion cost low [8]. For these advantages, in wireless network applications, distributed infrastructures are mostly used. Thus, we also propose a distributed algorithm here.

Our algorithm for contention reduction in distributed network follows the EPDP algorithm, which is stated and described in [11], upto the step where node v will receive packet from node u, will set timer, will construct F_v, U_v and B_v. After constructing F_v, it will check some cases:

(1) It will check whether the nodes in list F_v are neighbors or not.
  - If they are neighbors, for example $F_v = \{a, b\}$ and a, b are neighbors. Now it will perform a check:
    $R_a = N(a) - N(all\ nodes\ except\ a)$
    If $R_a = \varnothing$, it means that a′s neighbors can be covered by any other node. So a will be deleted from B_u.
  - We will do the same checking for b.
    $R_b = N(b) - N(all\ nodes\ except\ b)$
    If $R_b = \varnothing$, it means that b′s neighbors can be covered by any other node. So b will be deleted from B_u.
  - Then we will again produce F_v.

(2) If they are not neighbors, we will proceed as EPDP.

This algorithm is stated in the Algorithm 2 and 3.

Algorithm 2 CreateForwardList(U_v, B_v)

```
/*
Input :      Set U_v and set B_v of node v
Output:      The forward list F_v of node v
*/

F_v ← ∅
repeat
    Select w ∈ B_v such that w ∉ F_v and |N(w)∩U_v| is maximized
    F_v ← F_v ∪ w
    U_v ← U_v − N(w)
until (U_v = ∅) ∨ (F_v grows no further)
return F_v
```

Algorithm 3 Distributed CRCDS Pseudocode

/*
Precondition : Node v receives a packet p from node u where v $\in F_u$
$Input:$ $Set\ N(u), N(v), N^2(v)$
$Output:$ $The\ forward\ list\ F_v\ of\ node\ v$
*/

Step 1: $P = N(N(u) \cap N(v))$
$U_v = N^2(v) - N(u) - N(v) - P$
$B_v = N(v) - N(u)$
$wait\ time\ of\ node\ v \leftarrow current\ time + defer\ time$
In Step 2, if the same packet p is received again, interrupt the waiting and perform Step 5, assuming that w is the parent node from which the same packet p has been heard.
Step 2: Wait for $wait\ time$
Step 3: $F_v \leftarrow CreateForwardList(U_v, B_v)$
Let C_v is the set of vertices which will create contention if the vertices of F_v set forwards.
Step 4: **for** each $c \in C_v$ **do**
    $Q = N(c) - N(all\ the\ nodes\ in\ the\ network\ except\ c)$
    if $Q = \emptyset$ **then**
        $B_v = B_v - c$
    end if
end for
$F_v \leftarrow CreateForwardList(U_v, B_v)$
Now v will rebroadcast packet p appending F_v
The procedure exits.
Step 5: $P = N(N(v) \cap N(w))$
$U_v = U_v - N(w) - P$
$B = B - N(w)$
Discard the copy of packet p
if $B_v = \emptyset$ **then**
    Stop the transmission of packet p. The host v is now prohibited from rebroadcasting.
    The procedure exits.
else
    Resume waiting in Step 2
end if

The detailed analysis of this proposed algorithm using the scenario of Figure 2 is shown in Table 3.

Firstly, node 4 is chosen as it has the maximum degree and minimum ID among all nodes. Then the P set is created which is an empty set. Accordingly the U set and the B set is also created.

$$U_4 = N(N(4)) - \emptyset - N(4) - \emptyset = \{2, 7, 8, 10, 11\}$$
$$B_4 = N(4) - \emptyset = \{1, 3, 5, 6, 9\}$$

The timer is set. The B set is sorted in a manner where the node covering maximum number of node in U set comes first. So the sorted list is B=9, 6, 3, 5, 1. The nodes from B set covering maximum nodes of set U goes to the forwarding list which is 9, 6, 3. Now it will be checked if the nodes in the forwarding lists are neighbor or not. Here node 9 and node 6 are neighbors. So an R_9 will be computed first.

Table 3: Detailed analysis of the MCDS construction using EPDP algorithm in scenario in Figure 2

Chosen Node, v	*P*	*U*	*B*	*Time*	*Remarks*
4	$\emptyset$	2, 7, 8, 10, 11	1, 3, 5, 6, 9	0	Initial Forwarding List=9, 6, 3 Final Forwarding List=9, 5, 3
9	1, 3, 4, 5, 6, 7, 8, 9,10, 11	$\emptyset$	—	1	No Forwarding List
5	1, 2, 3, 4, 5, 6, 7, 8, 9	$\emptyset$	—	2	No Forwarding List
3	1, 2, 3, 4, 5, 6, 9, 10	$\emptyset$	—	3	No Forwarding List

$$R_9 = N(9) - N(all\ nodes\ except\ 9) = 11$$

The R_9 is not empty. So it cannot be deleted from the list. Then an R_6 will be computed.

$$R_6 = N(6) - N(all\ nodes\ except\ 6) = \emptyset$$

R_6 is empty so it can be removed and it will be replaced by node 5 which covers node 7 from the U set. So finally, the forwarding list is 4, 9, 5, 3.

So the algorithm intelligently selects nodes in forward list which mitigates the contention problem in a network.

Time Complexity analysis of Distributed CRCDS:
The major complexity here is to find the forwarding list. Let us assume that, the maximum degree of the network is Δ. Let we consider the worst case scenario. In the worst case for a node v, the size of B_v set can be equal to Δ. The new modification in this algorithm is to rebuild B_v if there is any node present in B_v which may create contention. So, in the worst case scenario, it may happen that we have to re build B_v. So it will take $O(\Delta^2)$ time in the worst case scenario to build B_v. Forwarding nodes F_v should be chosen from B_v in such a way that it covers all the nodes in U_v set. To do so, the loop of F_v will run Δ times in the worst case scenario. So, the final run time for constructing F_v will be $O(\Delta^3)$.

4 PERFORMANCE ANALYSIS

The examples from the previous section show that there exist situations when our Centralized CRCDS brings about an improvement over Sergiy's algorithm and our Distributed CRCDS brings about an improvement over EPDP. So, it is obvious that one would like to see how much of that improvement will materialize statistically in some of the typical networks. For doing so, we have carried out some simulation experiments in randomly generated networks by collecting the following performance measures:

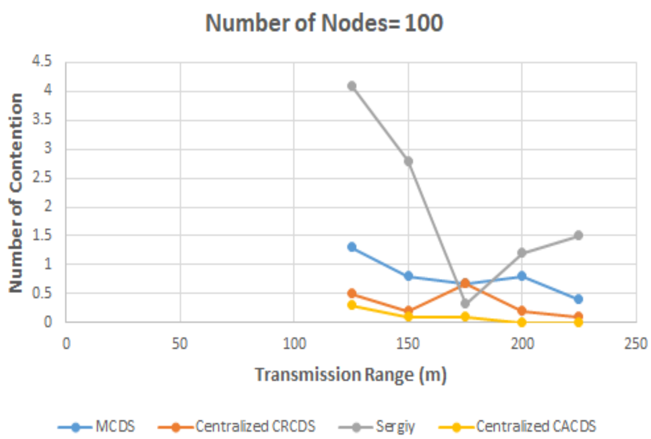

Figure 3: Performance comparison of MCDS, Sergiy, Centralized CRCDS and Centralized CACDS in term of number of contention occurs for a sparse network

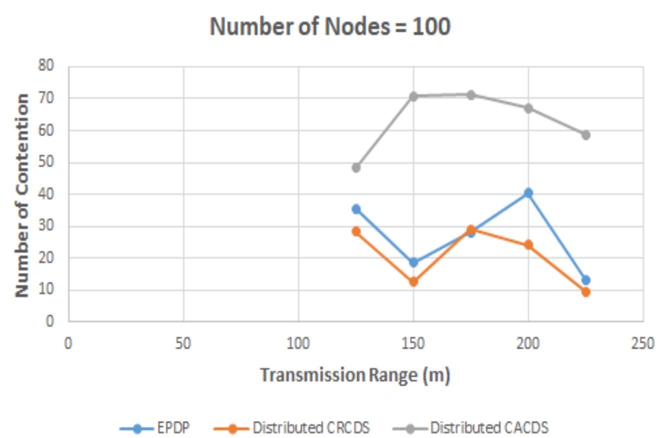

Figure 4: Performance comparison of EPDP, Distributed CRCDS and Distributed CACDS in term of number of contention occurs for a sparse network

(1) The number of forwarding nodes, i.e., the number of transmitting nodes required to propagate a broadcast packet within the network.
(2) The number of contentions, i.e., the number of nodes that will try to broadcast a packet using the same transmission area and will face *competition for resources*.

We have simulated the program for a network having transmission range between 125m to 225m with 100 to 500 nodes. In each scenario, we generated 10 different random networks to run the simulation and took the average value of the performance matrices to plot the graph. In the graph, we compared our Centralized CRCDS with traditional MCDS, Sergiy's and Centralized CACDS algorithm. On the other hand, we compared our Distributed CRCDS with EPDP and Distributed CACDS algorithm.

4.1 Performance Regarding Number of Contention Occurs

Figure 3 and Figure 4 shows the number of contentions for a sparse graph containing 100 nodes and Figure 5 and Figure 6 shows the number of contentions for a dense graph containing 500 nodes both ranging from 125m to 225m.

Figure 3 and Figure 5 shows the comparison result of global topology network algorithms. We see, our Centralized CRCDS gives better result than traditional MCDS and Sergiy's algorithm and gives more or equal same result as Centralized CACDS algorithm. In Figure 3 we can see that, the traditional MCDS algorithms generates 0.4 to 1.3 contentions in average, the Sergiy's algorithm generates 0.32 to 4.1 contentions in average while broadcasting a message in the network and our centralized CRCDS minimizes it .1 to .67 nodes in average.

On the other hand, Figure 4 and Figure 6 shows the comparison result of distributed network algorithms. Figure 4 shows that our algorithm avoids almost 7.4% to 20.1% and 16.2% to 42.8% contention occurrence in transmission range 125m and 200m respectively.

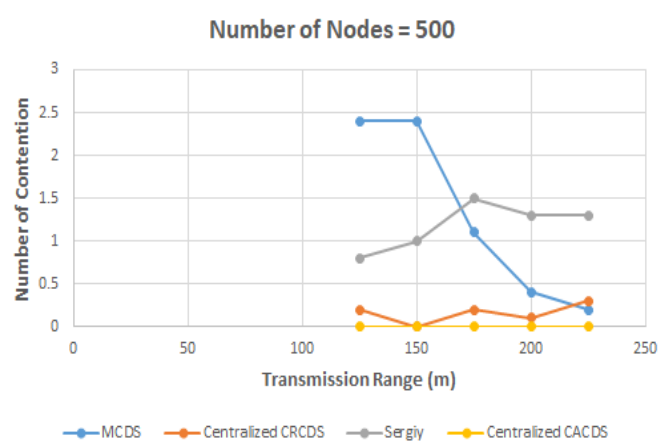

Figure 5: Performance comparison of MCDS, Sergiy, Centralized CRCDS and Centralized CACDS in term of number of contention occurs for a dense network

We can see the clear difference in the amount of contention in global topology network and distributed network by analyzing Figure [3-6]. We plotted the graph of Figure 3 and Figure 5 into one graph to visualize the difference properly which is shown in Figure 7. This is because centralized network algorithms use global topology network so, they produce less contention than that of distributed algorithms which use only 2-hop neighbor information.

4.2 Performance Regarding Number of Forwarding Nodes

While both of our approaches exhibits consistent superiority over the state-of-art algorithms in almost all case, it comes at the price of increased number of forwarding nodes.

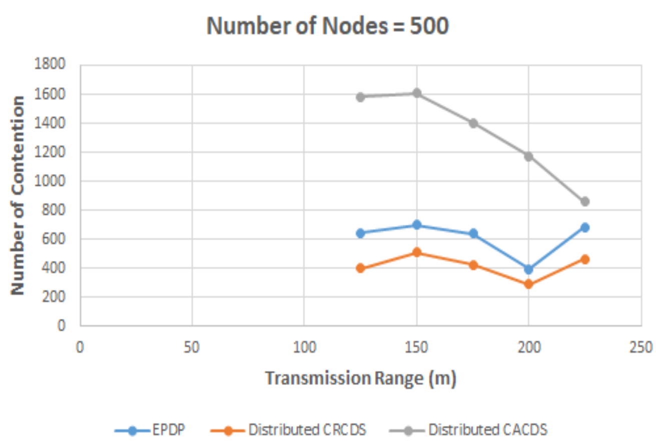

Figure 6: Performance comparison of EPDP, Distributed CRCDS and Distributed CACDS in term of number of contention occurs for a dense network

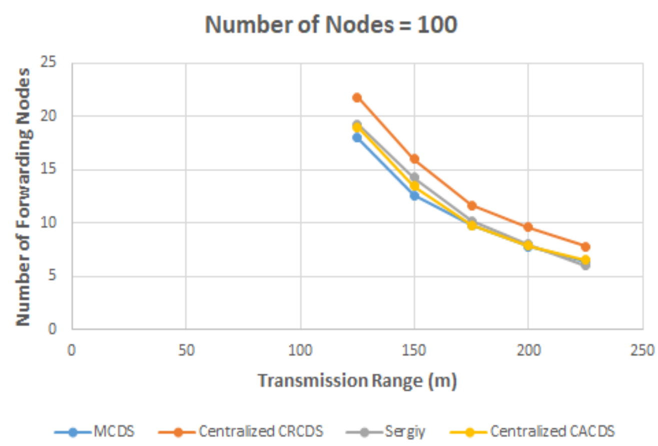

Figure 8: Performance comparison of MCDS, Sergiy, Centralized CRCDS and Centralized CACDS in term of number of forwarding nodes for a sparse network

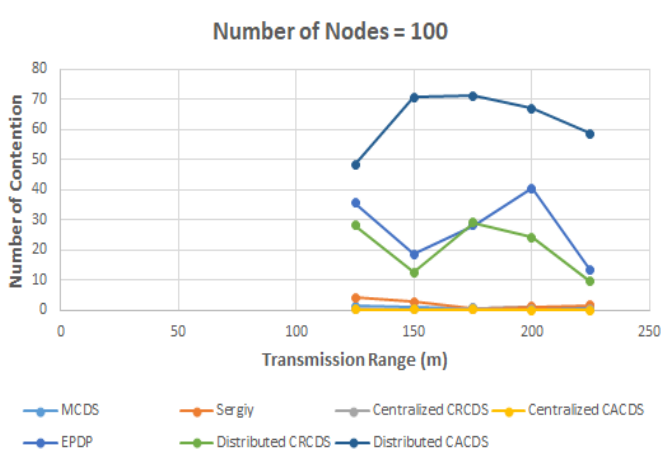

Figure 7: Performance comparison of MCDS, Sergiy, Centralized CRCDS and Centralized CACDS in term of number of contention occurs for a dense network

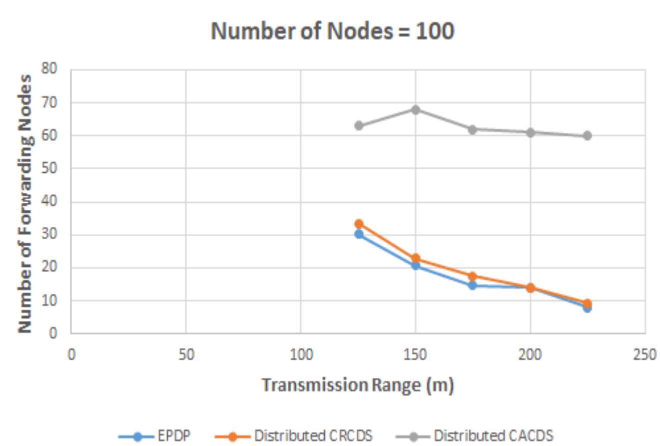

Figure 9: Performance comparison of EPDP, Distributed CRCDS and distributed CACDS in term of number of forwarding for a sparse network

Figure 8 and Figure 9 show the number of forwarding nodes for a sparse graph containing 100 nodes and Figure 10 and Figure 11 show those for a dense graph containing 500 nodes both ranging from 125m to 225m.

Here also can see the clear difference in the amount of forwarding nodes in global topology network and distributed network by analyzing Figure [8-11]. We plotted the graph of Figure 9 and Figure 11 into one graph to visualize the difference properly which is shown in Figure 12. Even in distributed network, it has less forwarding node than others.

Figure [8-11] show us that our algorithms have not increased the number of forwarding nodes much compared to the state-of-art algorithms rather they have less forwarding nodes in most of the cases compared to the state-of-art algorithms. So it turns out that, our approaches has reduced number of contentions without increasing the negative impact of increased forwarding nodes.

5 CONCLUSION

In this paper, we have discussed about new approaches for constructing connected dominating set to minimize contention in network which is an issue in broadcast storm problem. We presented a detailed simulation to analyze the behavior of the proposed algorithms and compared their performances with the state-of-art algorithms in terms of number of forwarding nodes and number of contention occurring among the nodes in CDS. The simulation result shows that our proposed algorithm for global topology network gives better result than the Sergiy's algorithm for constructing CDS and reducing contention. Our other proposed algorithm for

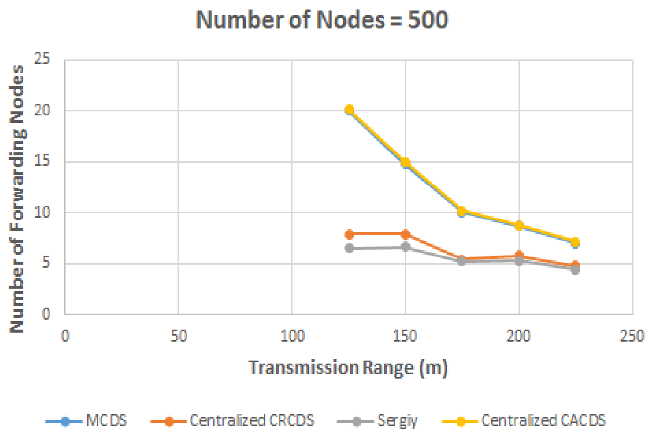

Figure 10: Performance comparison of MCDS, Sergiy, Centralized CRCDS and Centralized CACDS in term of number of forwarding for a dense network

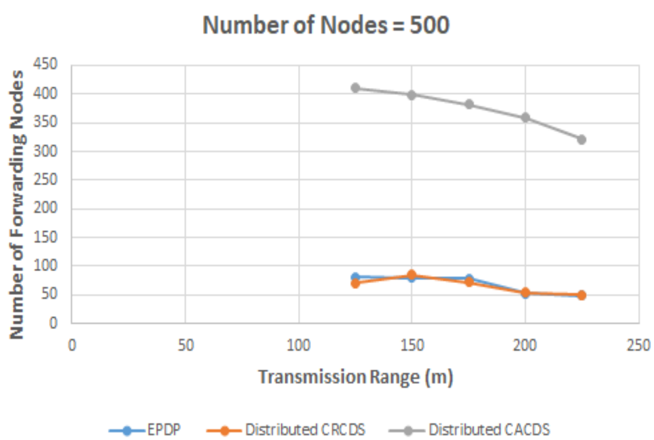

Figure 11: Performance comparison of EPDP, Distributed CRCDS and Distributed CACDS in term of number of forwarding for a dense network

distributed network also works better than EPDP algorithm in minimizing contention although the number of nodes increases in some cases.

The centralized algorithm works better than the distributed algorithm. In the distributed algorithm, some contention still remains, but in the centralized one we have not found any contention in our simulation. In future, we intend to work for making our distributed algorithm more efficient so that we can mitigate contention in a network. We also plan to construct the minimum connected dominating set (MCDS) and minimize contention in that. As we are minimizing contention, the execution speed of the network will also increase. In future, we will analyze the speed of execution of network as our performance metric to evaluate our proposed algorithms.

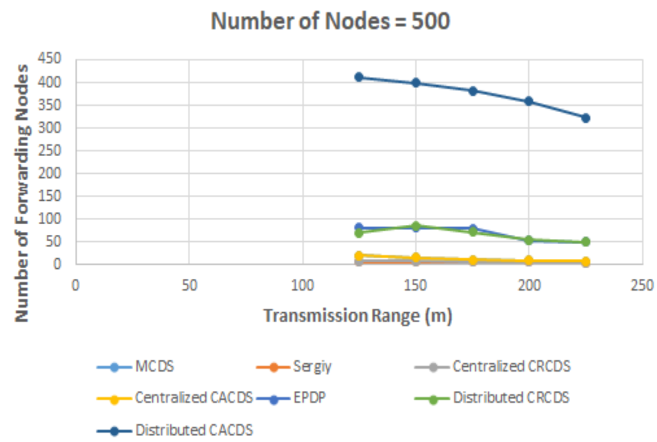

Figure 12: Performance comparison of MCDS, Sergiy , Centralized CRCDS, Centralized CACDS, EPDP, Distributed CRCDS and Distributed CACDS in term of number of forwarding for a dense network

REFERENCES

[1] Azzedine Boukerche. 2006. Handbook of Algorithms for Wireless Networking and Mobile Computing. Last accessed on September 28, 2019, at 03:31:00 PM. [Online], Available: "https://books.google.com.bd/books/about/Handbook_of_Algorithms_for_Wireless_Netw.html?id=CR7NBQAAQBAJ&redir_esc=y".

[2] Sergiy Butenko, Xiuzhen Cheng, Carlos A Oliveira, and Panos M Pardalos. 2004. A new heuristic for the minimum connected dominating set problem on ad hoc wireless networks. In *Recent developments in cooperative control and optimization*. Springer, 61–73.

[3] Sergiy Butenko, C Oliveira, and Panos M Pardalos. 2003. A new algorithm for the minimum connected dominating set problem on ad hoc wireless networks. *Proceedings of CCCT'03* (2003), 39–44.

[4] Nawrin Ferdous Chowdhury. 2017. Contention aware connected dominating set construction algorithm for wireless multi-hop networks. (2017).

[5] Sudipto Guha and Samir Khuller. 1998. Approximation algorithms for connected dominating sets. *Algorithmica* 20, 4 (1998), 374–387.

[6] Hyojun Lim and Chongkwon Kim. 2001. Flooding in wireless ad hoc networks. *Computer Communications* 24, 3-4 (2001), 353–363.

[7] Wei Lou and Jie Wu. 2003. On reducing broadcast redundancy in ad hoc wireless networks. In *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*. IEEE, 10–pp.

[8] Nicholas Loulloudes, George Pallis, and Marios Dikaiakos. 2008. *Information Dissemination in Mobile CDNs*. Vol. 9. 343–366. https://doi.org/10.1007/978-3-540-77887-5_14

[9] Stefan Pleisch, Mahesh Balakrishnan, Ken Birman, and Robbert Van Renesse. 2006. MISTRAL: efficient flooding in mobile ad-hoc networks. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 1–12.

[10] Raqeebir Rab, Shaheed Ahmed Dewan Sagar, Nazmus Sakib, Ahasanul Haque, Majedul Islam, and Ashikur Rahman. 2017. Improved Self-Pruning for Broadcasting in Ad Hoc Wireless Networks. *Wireless Sensor Network* 9, 02 (2017), 73.

[11] Ashikur Rahman, Md Endadul Hoque, Farzana Rahman, Sabuj Kumar Kundu, and Pawel Gburzynski. 2009. Enhanced Partial Dominant Pruning (EPDP) Based Broadcasting in Ad hoc Wireless Networks. *JNW* 4, 9 (2009), 895–904.

[12] Lu Ruan, Hongwei Du, Xiaohua Jia, Weili Wu, Yingshu Li, and Ker-I Ko. 2004. A greedy approximation for minimum connected dominating sets. *Theoretical Computer Science* 329, 1-3 (2004), 325–330.

[13] Yu-Chee Tseng, Sze-Yao Ni, Yuh-Shyan Chen, and Jang-Ping Sheu. 2002. The broadcast storm problem in a mobile ad hoc network. *Wireless networks* 8, 2-3 (2002), 153–167.

[14] Wikipedia contributors. 2018. Connected Dominating Set. https://en.wikipedia.org/wiki/Connected_dominating_set. Last accessed on December 10, 2018, at 08:03:00PM. [Online], Available: "https://en.wikipedia.org/wiki/Connected_dominating_set".