M.Sc. Engg. Thesis

# A Contention Aware Connected Dominating Set Construction Algorithm for Wireless Multi-Hop Networks

By

Chowdhury Nawrin Ferdous

Submitted to
Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
Master of Science in Computer Science and Engineering

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology (BUET)
Dhaka-1000

March 27, 2018

The thesis titled, **"A Contention Aware Connected Dominating Set Construction Algorithm for Wireless Multi-Hop Networks"**, submitted by Chowdhury Nawrin Ferdous, Roll No. 1014052060P, Session October 2014, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering and approved as to its style and contents. The examination held on March 27, 2018.

## Board of Examiners

1. ______________________

Dr. A.K.M. Ashikur Rahman — Chairman (Supervisor)
Professor
Department of CSE, BUET, Dhaka.

2. ______________________

Dr. Md. Mostofa Akbar — (Member) (Ex-Officio)
Head and Professor
Department of CSE, BUET, Dhaka.

3. ______________________

Dr. M. Kaykobad — Member
Professor
Department of CSE, BUET, Dhaka.

4. ______________________

Dr. Sadia Sharmin — Member
Assistant Professor
Department of CSE, BUET, Dhaka.

5. ______________________ 27-03-2018

Dr. Saifuddin Md. Tareeq — Member (External)
Professor
Department of Computer Science and Engineering
Science Complex
University of Dhaka, Dhaka-1000.

# Candidate's Declaration

This is to certify that the work presented in this thesis entitled "**A Contention Aware Connected Dominating Set Construction Algorithm for Wireless Multi-Hop Networks**" is the outcome of the investigation carried out by me under the supervision of Professor Dr. A.K.M Ashikur Rahman in the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology (BUET), Dhaka. It is also declared that neither this thesis nor any part thereof has been submitted or is being currently submitted anywhere else for the award of any degree or diploma.

Nawrin 27.03.2018

---

Chowdhury Nawrin Ferdous
Candidate

# Contents

# List of Figures

# List of Tables

# Acknowledgements

First of all, I would like to declare that all the appraisals belong to the Almighty. I would like to express my deep gratitude to my supervisor Professor Dr. A.K.M Ashikur Rahman for introducing me to the fascinating and prospective field of wireless multi-hop networks. I have learned from him how to carry on a research work, how to write, speak and present well. I thank him for his patience in reviewing my so many inferior drafts, for correcting my proofs and language, suggesting new ways of thinking, leading to the right way, and encouraging me to continue my research work. I again express my indebtedness, sincere gratitude and profound respect to him for his continuous guidance, suggestions and whole hearted supervision throughout the progress of this work. I convey my heartfelt reverence to my parents and other family members for giving their best support throughout my work to overcome the tedium of repetitive trials to new findings. Finally, every honor and every victory on earth is due to GOD, descended from Him and must be ascribed to Him. He has endowed me with good health and with the capability to complete this work. I deeply express my sincere gratitude to the endless kindness of GOD.

# Abstract

Efficient data dissemination among a set of mobile hosts is one of the most important concerns in wireless multi-hop networks. Broadcast and Multi-cast are two techniques that are being widely used for communicating among a group of nodes. The elementary flooding used for broadcasting requires each node to participate in forwarding task of a message that leads to notorious broadcast storm problem causing redundant transmissions, contentions and collisions. One convenient promising solution to overcome this problem is to construct a virtual backbone and connected dominating set (CDS). Several centralized (e.g. Minimum Connected dominating set (MCDS)) and distributed (e.g. Dominant Pruning (DP)) algorithms have been designed to compute CDS in order to reduce the number of rebroadcasting. However, none of the approaches aim at minimizing contentions. Contention is a phenomenon that occurs when a group of nodes want to transmit over a shared channel. During contention only one node gets access to the channel and the others defer their transmission for a later time. In this thesis, we propose a new heuristic dubbed as *Contention Aware Connected Dominating Set (CACDS)* to construct a CDS that will intelligently select the nodes to reduce contention. A centralized algorithm has been deduced first to construct CDS using the global topology information of a wireless network. Since it is difficult to gather the entire topology information, a distributed counterpart has also been presented where a node selects a subset of nodes from its immediate neighbors as forwarding nodes based on 2-hop neighborhood information and reduces contention. We have shown the detail analysis of how the proposed heuristics work better in terms of contention minimization compared to the state-of-the-art algorithms for constructing CDS. Though the total number of transmissions increases in the proposed approaches but the number of contentions among the forwarding nodes has been significantly reduced. In centralized environment, with the increase of 0-5% forwarding nodes, our heuristic generates almost 90-100% contention free CDS. In distributed environment, we need to increase the forwarding nodes by 1-5% to mitigate almost 4-19% contention in CDS.

# Chapter 1

# Introduction

In this chapter we provide a brief introduction to the problem that we solve in this thesis. We also highlight major contributions of the thesis.

## 1.1 Wireless Multi-hop Networks

A wireless Multi-Hop Network is a wireless network of nodes that uses multiple hops for data dissemination. In this network there exist(s) one or more intermediate nodes along the path that receive and forward packets via wireless links. Moreover, transmission over multiple "short" links might require less transmission power and energy than over "long" links. Multi-hop wireless networks avoid wide deployment of cables and can be deployed in a cost-efficient way. In case of dense multi-hop networks several paths might become available that can be used to increase robustness of the network. Four network paradigms can be classified as wireless multi-hop networks. These paradigms are: Mobile Ad-hoc Networks (MANETs), Wireless Sensor Networks (WSNs), Wireless Mesh Networks (WMNs), Vehicular Ad-hoc Networks (VANETs).

## 1.2 Wireless Ad-hoc Networks

A mobile ad hoc network (MANET) is a decentralized type of wireless network. The network is ad hoc because it does not rely on any pre-existing infrastructure, such as routers in wired networks or access points in managed (infrastructure-based) wireless networks. Instead, each node participates in routing by forwarding data for other nodes, so the determination of which nodes should forward data is made dynamically on the basis of network connectivity and the routing algorithm in use.

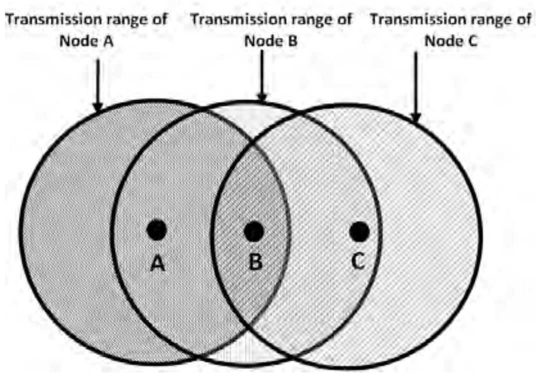



Figure 1.1: A sample wireless adhoc network

For example, in the scenario shown in Figure 1.1 node A and C both are within the transmission range of node B. If node A wants to send a packet to node C, it needs to rely on node B to forward the packet. In this case, node B will act as a router to relay the packet from node A to node C.

Due to considerations such as radio power limitation, channel utilization, and power-saving concerns, a mobile host may not be able to communicate directly with other hosts in a single-hop fashion. In this case, a multi-hop scenario occurs, where the packets sent by the source host are relayed by several intermediate hosts before reaching the destination host. As, it does not require any pre-existing infrastructure or any centralized management, the area where there is little or no communication infrastructure or existing infrastructure is inconvenient to use, wireless mobile users may still be able to communicate through formation of wireless multi-hop networks. Wireless mobile ad hoc networks are self-configuring, dynamic networks in which nodes are free to move. Wireless ad hoc networks lack the complexities of infrastructure setup and administration, enabling devices to create and join networks "on the fly" – anywhere, anytime. MANETs have applications in emergency search-and-rescue operations, decision making in the battlefield, data acquisition operations in inhospitable terrain, etc. where networks need to be deployed immediately but the base stations or fixed network or infrastructures are not available.

### 1.2.1 Broadcasting in MANET

As there is no centralized admin node, ad hoc networks face some difficulties that are required to be solved to establish a proper connection among the mobile nodes. Firstly, these mobile nodes are resource constrained devices with limited energy and power. If these nodes' power is not used in an organized way, the outcome could be devastating like it could lead to network partition. Another concern about wireless ad hoc network is, its topology is continuously changing as the node mobility is very high. Despite the mobility, it is often necessary that a piece of information needs to be reached throughout the entire network. Wireless ad hoc networks needs to broadcast messages for various services such as route discovery, periodic data dissemination, erasing an invalid route, locating a node, duplicate IP address detection or even for sending alarm signals in the entire network.

### 1.2.2 Flooding and Broadcast Storm Problem

An effortless approach to perform broadcast is by blind flooding. In blind flooding, when a host sends a message, its entire neighborhood will receive the message and upon receiving a message for the first time, a node rebroadcasts it. Though blind flooding ensures full coverage at high mobility but unfortunately, it results redundant transmissions, high energy consumption and finally leads to *broadcast storm problem* [1]. A host, on receiving a broadcast message for the first time, has the obligation to rebroadcast the message. Clearly, this costs n transmissions in a network of n hosts. In a CSMA/CA network, drawbacks of flooding include:

i **Redundant rebroadcasts:** When a node decide to broadcast a message but all of its neighbors have already received the message, then the broadcast of that node becomes unnecessarily redundant.

ii **Contention:** When a node broadcast a message, many of its neighbors need to rebroadcast the message to spread it in the whole network. When more than one nodes within the same transmission area try to broadcast a message at the same time, then their transmission may severely contend with each other, leads to contention problem.

iii **Collision:** Collisions are more likely to occur because of the insufficiency of back-off mechanism, the inadequacy of RTS/CTS dialogue, and the absence of collision detection etc. This can lead to more damage in the network.

## 1.3 Our Focus

The main focus of this work is to reduce contention that arises in blind flooding. Contention means *competition for resources*. The term is used especially in networks to describe the situation where two or more nodes attempt to transmit a message across the same medium at the same time. In a mobile 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.

One solution to overcome this problem is to compute a virtual backbone based on the physical topology, and run any existing routing protocol over the virtual backbone [2]. Connected Dominating Set (CDS) [3] is a widely used approach in this context. A CDS is generally constructed by at first modeling the entire network as a graph $G$ where all nodes form a vertex set $V$ and the communication links between nodes form an edge set $E$. Then, a CDS of a graph $G$ becomes a subset $D$ of the vertex set $V$ where any node in $D$ can reach any other node in $D$ by a path that stays entirely within $D$. That is, $D$ induces a connected sub-graph of $G$ and every vertex in $G$ either belongs to $D$ or is adjacent to a vertex in $D$. A minimum connected dominating set (MCDS) of a graph $G$ is a connected dominating set with the smallest possible cardinality among all connected dominating sets of $G$. In Figure 1.2, there is a graph of 6 nodes. Node 2 and 3 constructs minimum connected dominating set.

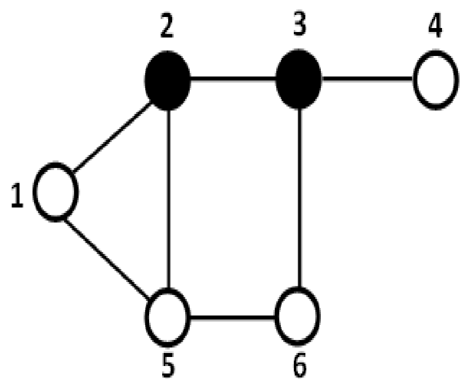



Figure 1.2: Minimum Connected Dominating Set

## 1.4 Our Contribution

A significant number of centralized and distributed algorithms [2,4,5] have been devised using the concept of MCDS to reduce the number of packet forwarding. Although MCDS based algorithms have been proposed to reduce redundancy, none of these works aim at minimizing contention. Thus, in this work we propose to fill this notable gap by introducing ***Contention-aware Connected Dominating Sets (CACDS)***. Our main contribution is to construct a connected dominating set that will select the forwarding nodes such that the contention among the forwarding nodes is minimized. The contribution of this work can be summarized as follows:

i We provide a centralized algorithm to construct contention aware connected dominating set (CACDS) using global topology information of the network as well as a distributed algorithm using 2-hop neighborhood information of each node.

ii A demonstration of the efficiency of the distributed algorithm using the centralized algorithm as a benchmark is also provided in this work.

iii Finally, we present a comprehensive simulation to analyze the behavior of the proposed algorithms and compare their performances with other state-of-the-art algorithms in term of number of forwarding nodes and number of contention.

## 1.5 Organization of the Thesis

The rest of the book is organized as follows- in chapter 2, we present the state-of-the-art research works regarding connected dominating set in both centralized and distributed environment of wireless multi-hop network. Chapter 3 presents the important terminologies along with the methodology of the new proposed algorithms. Chapter 4 deals with details analysis of the algorithms with suitable example. Chapter 5 demonstrates the simulation and performance evaluation and finally chapter 6 concludes our work with the contribution and limitations of our study and possible future works.

# Chapter 2

# Related Works

In this chapter, we will discuss on several approaches that many researchers have proposed to mitigate the *broadcast storm problem*. We will also focus on different ways of constructing connected dominating set for wireless multi-hop communication that has been proposed for efficient broadcasting.

A general solution to mitigate the broadcast storm problem is to construct a virtual backbone as the basis of routing and broadcasting. The idea to use connected dominating set (CDS) as a virtual backbone was first proposed by Ephermides in [6]. Since then many algorithms have been reported for using CDS construction to minimize the forwarding nodes to minimize the broadcast storm problems. The utility of CDSs in wireless ad hoc networks has been demonstrated in protocols that perform a wide range of communication functions. CDSs have formed an underlying architecture used by protocols including media access coordination [7, 8], multicast/broadcast [9, 10] and location-based routing [11]; energy conservation [12, 13]; and topology control [14, 15]. CDS can also be used to facilitate resource discovery in MANET [16]. This is also termed dominating set based routing [17] or Backbone based routing [18], or spine based routing [17].

The various methods are found in the literature on the CDS construction problem which can be classified as centralized algorithms [2, 19, 20] and distributed algorithms [21, 22] based on the network information.

## 2.1 Centralized Algorithms

The centralized algorithms need global information of the network to construct the CDS. Multicasting to all nodes in an ad hoc network is equivalent to broadcast. The problem of constructing optimal broadcast tree that minimizes the number of packet forwarding is very much similar to MCDS problem [23]. MCDS problem cannot be solved in polynomial time, so the optimal broadcast tree construction based on MCDS is proved to be an NP-complete problem [24]. So, researchers have been proposed several approximation algorithms and heuristics to find the optimal broadcast tree using the concept of MCDS.

### 2.1.1 Guha and Khullar's Algorithm

In [19], Guha and Khullar first proposed two greedy heuristic algorithms to construct CDS. In the first algorithm, the CDS is grown from one node outward. The algorithm begins by marking all the vertices (nodes) white. At first the algorithm chooses the node that has maximum number of white neighbors. The chosen node is colored black and all its neighbors are colored gray. The algorithm then iteratively scans the gray node and among the gray nodes with maximum number of white neighbors is selected next and marked it black. The algorithm terminates once all the nodes are marked as either gray or black and finally the black nodes create CDS.

In the second algorithm, a WCDS (Weakly connected dominating set) is constructed first, and then intermediate nodes are selected to create a CDS. The second algorithm also begins by marking all nodes white. A piece is defined to be either a connected black node, or a white node. The algorithm proceeds in two phases. The first phase iteratively selects a node that causes the maximum reduction of the number of pieces. Once a node is selected, it is marked black and its white neighbors are marked gray and it terminates when no white node is left. The second phase constructs a Steiner Tree that connects all the black nodes by coloring chains of two gray nodes black.

### 2.1.2 Ruan's Algorithm

By using a potential function, Ruan et al. [20] proposed a one-step greedy approximation algorithm to construct CDS. This algorithm also requires each node to be colored white at the beginning. If there exists a white or gray node such that coloring it black and its white neighbors gray would reduce the potential function, then choose the one that causes maximum reduction in the potential function which is defined in the way stated in [20].

### 2.1.3 Cheng's Greedy Algorithm

In [25], Cheng et al. proposed a greedy algorithm for MCDS in unit-disk graphs. This algorithm is based on an MIS (Maximal Independent Set). It is assumed that initially all nodes are colored white. The construction of a CDS contains four phases. In the first phase, an MIS is computed and all its members are colored red. In the second phase, a node that can decrease the maximum number of pieces is selected, where a piece is either a red node, or a connected black component. This node is colored black and all its non-black neighbors are colored gray. There is still some white node left after finishing the second phase. The third phase will compute a spanning tree for each connected component in the sub graph reduced by all white nodes. Connect each tree to the nearest black component with black nodes accordingly. All non-leaf tree nodes are colored black while leaf nodes are colored gray. The last phase will seek chains of two gray nodes to connect disjoint black components.

### 2.1.4 Min's Algorithm

In [26] Min et al. proposed to use a Steiner tree with minimum number of Steiner nodes (ST-MSN) [27] to connect a maximal independent set. This algorithm contains two phases. The first phase constructs an MIS with the property that every subset of the MIS is two hops away from its complement. Then color all nodes in the MIS black and color all other nodes gray. In the second phase, a gray node that is adjacent to at least three connected black components is colored black in each step. If no node satisfying this condition can be found, a gray node that is adjacent to at least two connected black components will be colored black.

### 2.1.5 Butenko's Algorithm

The heuristic proposed in [2,28] is pruning-based. In other words, the connected dominating set $S$ is initialized to the vertex set of graph $G(V, E)$, and each node will be examined to determine whether it should be removed or retained. At first it is assumed that all nodes in $S$ are colored white at the beginning and defined the effective degree of a node to be its white neighbors in $S$. Consider a white node $x \in S$ with minimum effective degree. If removing $x$ from $S$ makes the induced graph of $S$ disconnected, then retain $x$ and color it black. Otherwise, remove $x$ from $S$. At the same time, if $x$ does not have a black neighbor in $S$, color its neighbor with maximum effective degree in $S$ black. This procedure needs to be repeated until no white node left in $S$.

## 2.2 Distributed Algorithms

For wireless networks and MANETs, distributed CDS construction is more effective due to the lack of a centralized administration. On the other hand, the large topology size also prohibits the centralized CDS computation. There exist several distributed algorithms [17,29] for MCDS computation in the context of ad hoc wireless networking. The first one builds a rooted tree distributedly. Das and Bharghavan in [17] provided the distributed implementation of the two centralized algorithms given by Guha and Khuller in [19]. Both implementations suffer from high message complexities. The one given by Wu and Li in [29] has no performance analysis. It needs at least two-hop neighborhood information. The status of each host is assigned based on the connectivity of its neighbors.

Various heuristics using the neighborhood information have been proposed to minimize the redundant rebroadcasting. Mainly, all mechanisms can be divided into two categories which are given below:

### 2.2.1 Proactive Approaches

In this approach, upon receiving a packet, a receiver decides whether to forward the packet or not. Some of the algorithms that use proactive approaches are summarized below:

#### Self Pruning

In [21], Lim and Kim proposed a reactive approach named **Self Pruning (SP)**. In SP, a node, $u$ attaches its neighbor list, $N(u)$ while forwarding a packet. Another node $v$, in the transmission range of $u$ when receives the packet compares its own neighbor list, $N(v)$ with $N(u)$. If there are new nodes that $v$ can cover then its forwards the packet. Basically, if $N(v) - N(u) \neq \emptyset$ , then it forwards. However, since there could be several receivers of $u$ those cover the same node $w$ and the nodes take decision without coordinating among themselves, there will remain a lot of redundant re-transmissions.

#### Scalable Broadcast Algorithm

In [22], another algorithm named **Scalable Broadcast Algorithm** is proposed. It is achieved by delaying the rebroadcast for a random period. In this waiting period, when a node receives the duplicate messages from other nodes, it continues comparing its own neighbor list with the neighbor list of later sender nodes and updates the covering set.

#### Improved Self Pruning

In **Improved Self Pruning** [30], each node makes decision whether to forward or not based on 3-hop neighbor information and performs much better than traditional Self Pruning.

### 2.2.2 Reactive Approaches

While transmitting a packet, the transmitter node decides which of its neighbor nodes should forward the packet. The forward list is attached in the header of the broadcast packet. Then receiving a packet, a node which is requested to forward the packet again determines the forward list from its neighbors similarly; otherwise, does not construct any forward list and does not rebroadcast. The flooding ends when there is no more nodes to rebroadcast anymore. Recently different types of reactive approaches have been suggested by researchers.

#### Dominant Pruning

One of the most promising way is to use 2-hop neighborhood information before selecting a forwarding node which is done by Lim and Kim in [21] is known as Dominant Pruning (DP). When a node, $v$ receives a packet from a node $u$, then it selects minimum number of nodes from its 1-hop neighbors deducting the neighbors of $u$, $N(v) - N(u)$ to cover all the nodes of the set $U_{\text{v}} = N(N(v)) - N(v) - N(u)$. $N(N(v))$ is all the 2-hop neighbors of $v$. $N(v)$ is discarded as they will receive when $v$ will forward and $u$ has already forwarded, so $N(u)$ is also covered by $u$. This algorithm performs better than blind flooding and self pruning. Authors in [31], suggested two heuristics of dominant pruning- Partial Dominant Pruning and Total Dominant Pruning.

- Partial Dominant Pruning uses 2-hop neighborhood information and it works more effectively. Other than deducting $N(v)$ and $N(u)$ from $N(N(v))$, PDP also deducts the neighbor of common neighbors of node $u$ and $v$. This reduction of sets reduces number of forward list than DP.

- Total Dominant Pruning requires 2-hop neighbors of the immediate sender node to be piggybacked with the broadcast packet. Thus, a node $v$ receiving a packet from node $u$ also deducts $N(N(u))$ from $N(N(v))$.

**Enhanced Partial Dominant Pruning (EPDP)**

According to [32], EPDP is an extended version of PDP. Only difference between PDP and EPDP is, EPDP introduces a delay before forwarding a received broadcast packet. It takes advantage of the fact, that the same node may hear the same packet several times from its neighbors. Whenever the node receives the same packet from another neighbor, it uses the overheard information and keeps updating the $U$ set, it needs to cover according to the newly received packet. The defer time is selected according to the position of the node in the forward list. This solution brings better performance than DP and PDP. Basically, EPDP is combination of reactive and proactive approach. As the nodes in the forward list from the immediate sender defers its retransmission and after defer time the node itself decides whether to rebroadcast the message or not.

Throughout this chapter, we can see that there are several approaches to reduce redundant broadcast by using connected dominating set (CDS). Some of the algorithms use global topology information to minimize the redundancy. However, to give better performance in mobile environment, there are also distributed algorithms. But, CDS has never been computed keeping in mind the contention issue in any of the prior works. Therefore, we are proposing to fill up this notable gap by introducing contention aware minimum connected dominating set.

# Chapter 3

# Methodology

## 3.1 Network Model

A wireless ad hoc network is often represented by a graph in which vertices correspond to the communicating nodes, and a directed edge from one vertex to another indicates that the node corresponding to the former can send data directly to the node corresponding to the later one. It is common to assume that propagation conditions can be modelled simply by considering "transmission range" within which communication is possible, and outside of which it is impossible. If all nodes have equal transmission ranges, then the graph becomes undirected. We

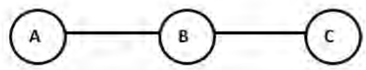



Figure 3.1: Representation of the network from Figure 1.1

use a simple graph $G(V, E)$ to represent an ad hoc network, where $V$ represents a set of wireless mobile hosts (nodes) and $E$ represents a set of edges. An edge $(u, v)$ indicates that both hosts $u$ and $v$ are within the transmission range of each other. Figure 1.1 shows an example of wireless ad hoc network consisting of three nodes A, B and C. Here, A and C both are in transmission range of B. Again, B is in transmission range of A and C. The circle around a node indicates its transmission range. The nodes within the circle are considered as the neighbors of that node. That means, when a node forward a packet, all the nodes within this transmission range receives the packet. As, node A and node B both are within each other's transmission range, they can communicate with each other and in the network graph there will be an edge between them. The same case also applies for node B and node C.

Figure 3.1 is the conversion of the graph of the network from Figure 1.1.

## 3.2 Auxiliary Definition

In order to develop the algorithm, we state some definition and introduce some terminology relevant to the research.

i **Dominating Set:** Dominating Set for a graph $G = (V, E)$ is a subset $D$ of the Vertex Set $V$ such that each vertex $u \in V$ is either in $D$ or adjacent to some vertex $v$ in $D$. The elements of dominating set are called dominators. Examples of dominating sets in a graph $G$ are given below.

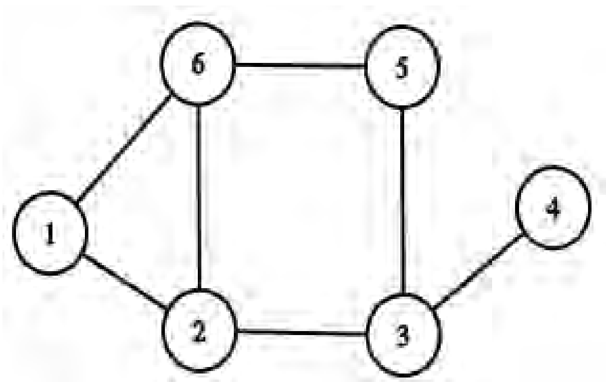



Figure 3.2: {1,3},{2,3,5},{1,2,3,4} are dominating sets

ii **Connected Dominating Set:** A Connected Dominating Set (CDS) of a graph $G = (V, E)$ is a set of vertices with two properties:

- $D$ is a dominating set in $G$.
- $D$ induces a connected sub graph of $G$.

In figure 3.2 {2,3} and {1,2,3,4} are connected dominating sets.

iii **Minimum Connected Dominating Set:** A minimum Connected Dominating Set (MCDS) is a connected dominating set with the smallest possible cardinality among all the CDSs of

$G$. As in Figure 3.2, the Minimum Connected Dominating Set is {2,3}.

iv $ColorW$**:** $ColorW$ set consists of all nodes $v \in V$. Whenever a node is selected as forwarding node, the node and all its 1-hop neighbors are discarded from $ColorW$ set.

v $ColorB$**:** The nodes consisting in $ColorB$ set are the forwarding nodes of the network.

vi $ColorG$**:** When a node $u \in ColorB$ forwards any message, all of its 1-hop neighbors receive the message and become member of $ColorG$ set. So, $ColorG = \bigcup_{u \in ColorB} N(u)$

vii $Candidate\_Set$**:** The nodes in $ColorG$ set which are eligible (who has minimum number of neighbors belongs to $ColorB$ set) for becoming a forwarding node are the members of $Candidate\_Set$.

viii **Neighborhood of a node:** Suppose, $v$ is an arbitrary node of the network and it has received a broadcast packet from node $u$ for forwarding. Node $v$ will create a forward list and append the list in the packet header before rebroadcasting.

- $N(v)$**:** $N(v)$ is the set of all 1-hop neighbors of node $v$. The nodes that are in the transmission range of $v$ are member of this set. Note that, node $v$ itself also is a member of this set, $\{v\} \in N(v)$.
- $N(N(v))$**:** $N(N(v))$ is the set of all nodes within 2-hop of node $v$, $\{v\} \subseteq N(v) \subseteq N(N(v))$.
- $N(N(v)) - N(v)$**:** The nodes that are exactly 2-hop away from $v$ are included in this set.
- $F_{\text{v}}$**:** The list of 1-hop neighbors of $v$ that are selected for forwarding by node $v$, $F_{\text{v}} \subseteq N(v)$.
- $B_{\text{v}}$**:** The set of one hop neighbors of node $v$ that are eligible to be included in the forwarding list $F_{\text{v}}$. When node $v$ receives a packet from node $u$ and $v \in F_{\text{u}}$, it selects its own forward list. Node $u$ and node $v$ may have some common neighbors, so while selecting forwarding nodes, node $v$ does not need to consider those common neighbors as they were already considered by node $u$. Thus, $B_{\text{v}} = N(v) - N(u)$ and node $v$ selects forwarding nodes from $B_{\text{v}}$. That means, $F_{\text{v}} \subseteq B_{\text{v}}$.

- **$U_v$:** $U_v$ is the set of nodes that need to be covered by using nodes from $B_v$ while $v$ creates its forwarding list $F_v$ .

## 3.3 Considered Algorithms

As mentioned earlier, we are proposing a centralized algorithm as well as distributed algorithm to minimize contention problem in wireless multi-hop network. The problem of selecting the broadcasting nodes in order to reduce redundant broadcast is very much similar to MCDS problem. Only the nodes in MCDS will broadcast the message in order to distribute it to the whole network. MCDS construction is considered to reduce the problem of redundancy that was created by blind flooding ignoring the fact of contention and collision. We are proposing a way to construct connected dominating set with no or minimal contention based on MCDS algorithm. However, the major drawback of this algorithm is it needs global network topology information to select the broadcasting nodes efficiently. In wireless environment where nodes may move freely, it is nearly impossible to have global topology information. Therefore, it becomes difficult to utilize this centralized algorithm in a distributed environment. So, there are two heuristic algorithms named- self pruning and dominant pruning to flood packets more effectively than blind flooding in wireless networks. Both of these algorithms are designed to reduce redundancy of broadcasting using neighborhood information, neither of the algorithms consider the contention issues. We have considered the dominant pruning algorithm to modify in order to minimize contention problem in distributed environment. In this section we will discuss about the state-of-art algorithms those are used as a base of our proposed algorithms.

### 3.3.1 MCDS Construction Algorithm

At the start of the algorithm, all nodes in the network are colored white. The node with maximum cardinality is then selected and colored black. All the one-hop neighbors of that node are colored gray. A gray node having maximum number of white neighbors is then selected. The selected gray node is then colored black and all its white neighbors are colored gray. The selection process recursively runs until no white node exits. The set with all the black nodes are the resultant nodes that makes MCDS.

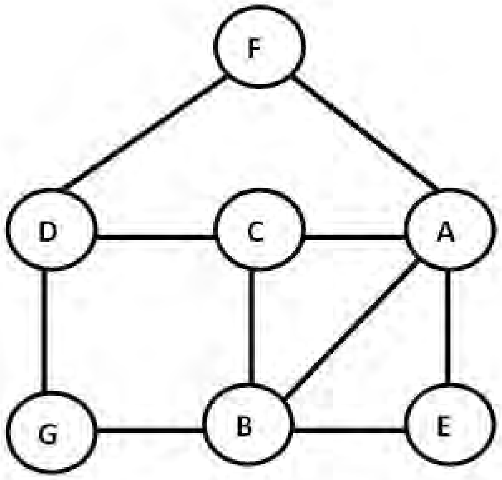



Figure 3.3: A sample network with 7 nodes

In figure 3.3, there are 7 nodes. At first either node A or node B could be selected because they have the same maximum number of white neighbors (i.e., 4). Suppose Node A is selected first. Nodes $B, C, E, F$ are colored gray and all becomes candidate for selection in the next phase. However, nodes B, C and F has 1 white neighbors each and node E has none. So any of them except E could be selected. Suppose Node B is selected next. So G becomes gray node. At this stage either of nodes C, G and F can be selected because each has 1 white neighbor each. So, let's select node C. Node D becomes gray node and no more white nodes exist in the network at this stage; the algorithm stops. Therefore,

$$MCDS = \{A, B, C\}$$

### 3.3.2 Dominant Pruning Algorithm

Dominant pruning (DP) uses the neighborhood information to select forwarding nodes from its neighbors. It extends the range of neighborhood information into two-hop apart nodes. This two-hop neighborhood knowledge can be obtained by exchanging the adjacent node list with the neighbors. The sender nodes selects the adjacent nodes to relay the packets to complete the broadcast in dominant pruning. The IDs of the selected adjacent nodes are recorded in the packet as a forward list. An adjacent node that is requested to relay the packet again determines the forward list. This process is iterated until broadcast is complete.

Now let us discuss, how dominant pruning selects forwarding nodes from its neighbors. Each node $u$ determines its forward list as a subset of its one-hop neighbors whose transmission will cover all two-hop neighbors of $u$. Then node $u$ selects some nodes from $B_u$ to cover all the nodes that belong to $U_u$. The whole scenario can be mapped as a set cover problem as indicated in Figure 3.4.

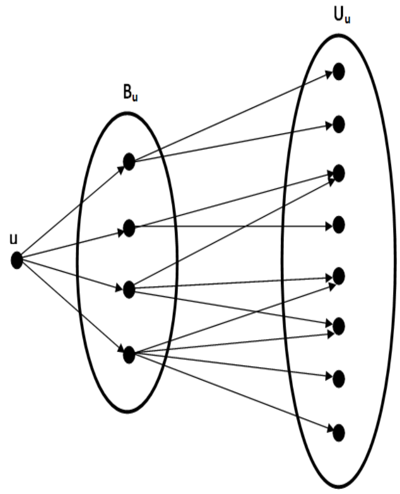



Figure 3.4: Connectivity among node u, $B_u$, $U_u$

Suppose, a node $v$ receives a packet from node $u$. The sender node $u$ also sends a *forwarding list ($F_u$)* with the packet header. If $v \notin F_u$, the node will not participate in rebroadcasting and if $v \in F_u$, then the node $v$ will rebroadcast and will create its own forward list ($F_v$) and insert it to the header of rebroadcast copy. The node then start constructing $U_v$ which is all uncovered two-hop neighbors of $v$ ($N(N(v))$).

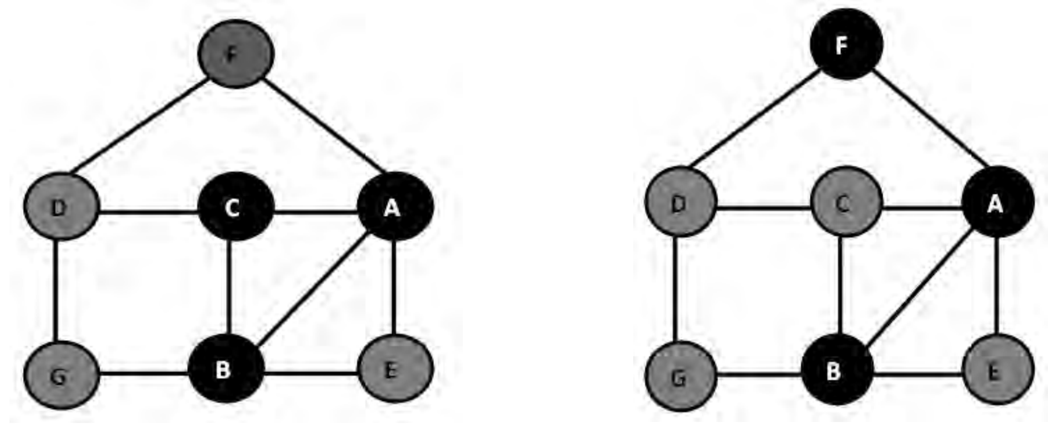
$$U_v = N(N(v)) - N(v) - N(u)$$

One hop neighbors of $v$ ($N(v)$) will receive packet when $v$ will broadcast. As every node knows its two hop neighbors, so, $N(u)$ is known to $v$. then $v$ sets $F_v = \emptyset$ and

$$B_v = N(v) - N(u)$$

The set $B_v$ represents those neighbors of $v$ which are possible candidates for inclusion in $F_v$. Then, in each iteration, $v$ selects a neighbor $w \in B_v$, such that $w \notin F_v$ and the list of neighbors of $w$ covers the maximum number of nodes in $U_v$ , i.e $|N(w) \cap U_v|$ is maximized. Next $v$ includes $w$ in $F_v$ and sets $U_v = U_v - N(w)$ . The iterations continue for as long as $U_v$ becomes empty or no more progress can be accomplished. Let us illustrate forwarding list creation in dominant pruning by the example scenario of Figure 3.3. Assume node A is the broadcast initiator here,

$$N(N(A)) = \{A,B,C,D,E,F,G\}$$




(a): A CDS that has contention (b): A CDS without any contention

Figure 3.5: CDS construction with and without contention

$$N(A) = \{A,B,C,E,F\}$$

$$U_A = \{D,G\}, B_A = \{B,C,E,F\}$$

It is now need to choose a node from $B_A$ which is {B,C,E,F} to cover nodes in $U_A$ which is {D,G}. It will pick node B to cover node G and node C to cover node D.

$$F_A = \{B, C\}$$

## 3.4 The Proposed Algorithms

As stated earlier, the main goal of this thesis is to construct the CDS in such a way so that when the nodes in CDS forward the message, it will reach to all nodes in the network as well as the contentions among the nodes in CDS are as minimum as possible. For example, in Fig 3.5(a), if the CDS is constructed in usual traditional method, node A, B and C will be selected to construct CDS which will lead to contention problem. Upon receiving message from node A, if two of its neighbors node B and node C start to rebroadcast it, they have to contend with each other first for gaining access to the shared channel as they are within the transmission range of each other. So, the selection of the nodes should be done intelligently to minimize this type of contentions. Figure 3.5(b) presents a scenario, where node F is selected instead of node C to avoid the contention. In this section, we propose a new algorithm for constructing CDS that reduces the contention problem. We are proposing one centralized algorithm based on MCDS construction and one distributed algorithm based on dominant pruning algorithm.

### 3.4.1 Centralized Contention aware Connected Dominating Set (Centralized CACDS)

The first algorithm is a centralized one because it needs the global topology information of a network for constructing CDS. At the start of the algorithm, all the nodes are colored white. The node with the maximum cardinality is then selected and colored as black and all the neighbors of that nodes are colored as gray. To minimize the contention problem, the major modification of the MCDS algorithm, is done while selecting forwarding nodes among the gray nodes. Among the gray nodes, the nodes that have minimum number of black neighbors are selected and placed in a set called $Candidate\_Set$. The black nodes are already in CDS. Therefore, by selecting a gray node with minimum black neighbors reduces the chance of contention. Among the nodes in $Candidate\_Set$ (gray nodes with minimum number of black neighbors), the node which has maximum number of white neighbors is finally selected and colored as black and all its white neighbors are colored as gray. A recursive selection process runs till there is no white node left in the network. The details is stated in Algorithm 1.

Consider Figure 3.6, the network contains 7 nodes labeled from A to G. Among all the nodes node A has the maximum cardinality (which is 4 here), so, according to the algorithm, node A is colored black and all its neighbors (node B,C,E and F) are colored gray (illustrated in Figure 3.6(b)). In the next step, all the gray nodes have only 1 black neighbor that is node A and each of the gray nodes B, C and F has 1 white neighbor and node E has none. So we can select either of Node B, C , F. Suppose node B is selected to cover node G and it is colored as black and node G is colored as gray (illustrated in Figure 3.6(c)). The only white node remaining in the network at this stage is node D. The gray node C has now two black neighbors (node A and node B), gray node G and node F each has one black neighbor, so either of node G and node F can be selected. Suppose node F is selected to cover node D ( In Figure 3.6(d)). So, the final Contention aware Connected Dominating Set consists of node A,B and F.

$$CACDS = \{A, B, F\}$$

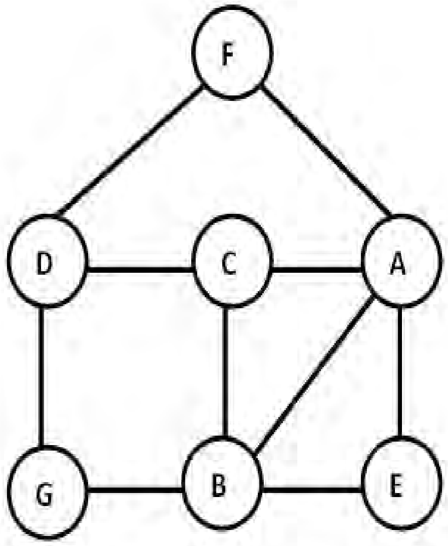
In the traditional MCDS algorithm, node C is selected to be a member of CDS to reduce

**Algorithm 1** Centralized CACDS

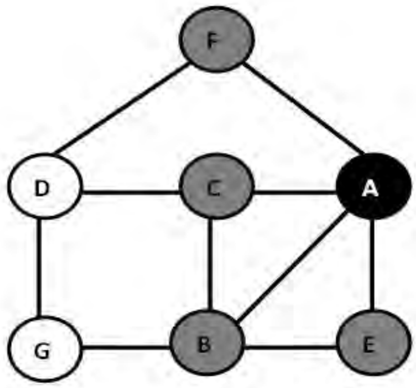
INPUT: G(V,E)

RESULT {CACDS}

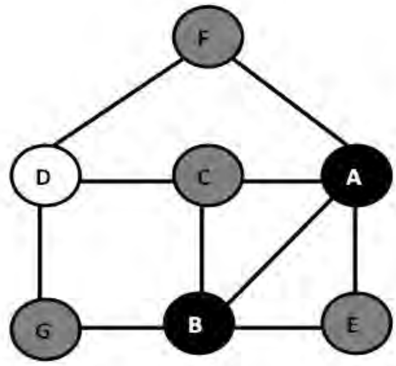
```
ColorB = ∅, ColorG = ∅, CACDS = ∅,
ColorW = all nodes v ∈ V;
Select a node v ∈ V with max(degree(v));
ColorB = {v};
ColorG = N(v) − {v};
ColorW = ColorW − N(v);
while ColorW ≠ ∅ do
    MaxWhite = −1, MinBlack = 10000, Candidate_Set = ∅;
    for all node u ∈ ColorG do
        BlackCount = ‖N(u) ∩ ColorB‖;
        if BlackCount < MinBlack then
            MinBlack = BlackCount;
        end if
    end for
    for all node u ∈ ColorG do
        BlackCount = ‖N(u) ∩ ColorB‖;
        if BlackCount == MinBlack then
            Candidate_Set = Candidate_Set ∪ {u};
        end if
    end for
    for all node w ∈ Candidate_Set do
        WhiteCount = ‖N(w) ∩ ColorW‖;
        if WhiteCount > MaxWhite then
            MaxWhite = WhiteCount ;
            selectedNode = w;
        end if
    end for
    if MaxWhite > 0 then
        ColorB = ColorB ∪ {selectedNode};
        ColorW = ColorW − N(selectedNode);
        ColorG = ColorG ∪ (N(selectedNode) − ColorB);
    else
        ColorG = ColorG − Candidate_Set;
    end if
end while
CACDS = ColorB;
```



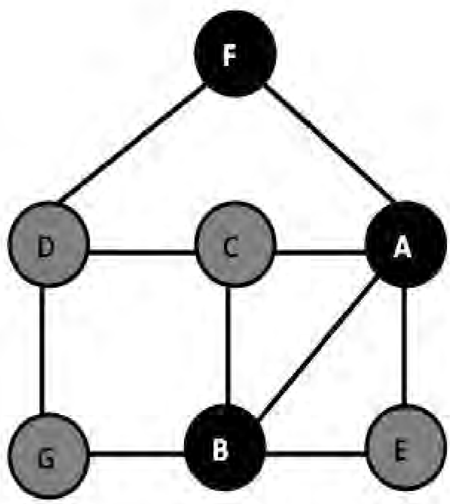

(a): All nodes are colored WHITE




(b): Node A is selected and colored BLACK and all its neighbors are colored GRAY




(c): Node B is selected and the process continues




(d): Finally node A,B,F constructs the CACDS

Figure 3.6: Step by step construction of Centralized Contention aware Connected Dominating Set (Centralized CACDS).

redundancy. However, if node C is selected, there will be contention between node B and node C. So once node A broadcasts a packet, both node B and node C try to rebroadcast the message and end up with having contention with each other for accessing the shared channel as those are within the transmission range of each other. So, by selecting node F instead of node C, the CACDS avoids contention in the network.

### 3.4.2 Complexity Analysis of Centralized CACDS

In the algorithm, the iteration of the "$WhileLoop$" from line 7-35 continues until $ColorW$ set does not become empty. At first set $ColorW$ consists of all nodes in the network. So, the loop body will run in $O(V)$ times where $V$ is the total number of nodes in the network. Inside the $WhileLoop$, there are loops that are used to find the nodes which have minimum number of black neighbors and maximum number of white neighbors which will again run in $O(V)$ times in the worst case scenario. So, the run time complexity of Centralized CACDS is $O(V^2)$.

### 3.4.3 Theoretical Correctness of CACDS

*Lemma 1:* $CACDS$ is a connected dominating set.

*Proof:* The proof is by contradiction. Suppose the vertex set of CACDS is,

$$V_{CACDS} = \{v_1, v_2, v_3 ... v_m\}$$

Assume node $v_{\mathrm{i}} \in V_{CACDS}$ cannot connect with other nodes in $CACDS$. According to the algorithm, the black nodes are selected among the gray nodes. When a node becomes gray, that means it has at least one black neighbor in the network. As $v_i$ is a member of $CACDS$, the node must be black. When $v_i$ was selected, it was among one of the gray nodes and each gray node is connected to one of the black nodes. Therefore, each black node has a path to connect other nodes in $CACDS$. The result contradicts with the hypothetic premises. So, $v_i$ must have a path to connect other nodes in the $CACDS$. Hence each node in $CACDS$ must be connected.

*Lemma 2:* The connected dominating set constructed by $CACDS$ covers all the nodes in the network.

*Proof:* Assume that, $U$ is the set consisting of all the nodes in the network, $U = \{x_1, x_2, ...x_{\mathrm{n}}\}$, $n$ is the number of nodes in the network, and vertex set of CACDS is $V_{CACDS} = \{v_1, v_2, v_3...v_m\}$, where $m$ is the total number of nodes in the $CACDS$. $N(v_{\mathrm{i}})$ is the set of nodes which become gray after selecting $v_{\mathrm{i}}$ as a member of $CACDS$ i.e., $N(v_{\mathrm{i}})$ represents the set consisting of all adjacent nodes of $v_{\mathrm{i}}$.

$$N = N(v_1) \cup N(v_2) \cup N(v_3) \cup ...N(v_{\mathrm{m}}) \tag{3.1}$$

In order to prove, $CACDS$ covers all the nodes in the network, we have to prove that $U = N$. As $CACDS$ is a connected dominating set, so every node $x_{\mathrm{i}} \in U$ either same as $v_{\mathrm{j}}$ or is adjacent to $v_{\mathrm{j}}$ for some $j$. In other words,

$$\forall i \Big[ \exists j \Big[ x_i \in N(v_j) \Big] \Big]$$

Thus,

$$\{x_1, x_2, ...x_{\mathrm{n}}\} \subseteq N(v_1) \cup N(v_2) \cup N(v_3) \cup ...N(v_{\mathrm{m}}) \tag{3.2}$$

From Equation 3.2,

$$\{x_1, x_2, ...x_{\mathrm{n}}\} \subseteq N \tag{3.3}$$

By definition,

$$\{x_1, x_2, ...x_{\mathrm{n}}\} \subseteq U \tag{3.4}$$

From Equation 3.3 and 3.4, by the axiom of extensionality, we can say that,

$$\forall x_i (x_i \in U \iff x_i \in N) \implies U = N \tag{3.5}$$

### 3.4.4 Distributed Contention aware Connected Dominating Set (Distributed CACDS)

Like dominant pruning, in the proposed distributed algorithm it is assumed that, each nodes knows its 2-hop neighborhood information. The assumption is not unreasonable because most of the routing algorithms are designed based on the neighborhood information. Every node periodically sends "whoami" packet to its neighbors to inform its presence in the network.

When a sender node sends a packet it also piggybacks its neighbor list in the packet header, so it becomes easy to derive two-hop information $N(N(u))$ by the other nodes of $N(u)$. In the proposed algorithm, a node uses its two-hop neighbor information to select the adjacent forwarding nodes.

The core difference of the distributed CACDS algorithm with the dominant pruning algorithm is that, it selects the adjacent forwarding nodes in such a way so that the contention between the forwarding nodes is as minimum as possible when they will rebroadcast the packet upon arrival from the sender. As discussed in the previous section, in DP, if a node $v$ is selected to rebroadcast a message by node $u$, it starts to make its own *forward list* ($F_{\text{v}}$) from a subset of its one-hop neighbors($B_{\text{v}}$) to cover its uncovered two-hop neighbors ($U_{\text{v}}$). Also in our algorithm, after receiving a packet from node $u$, if node $v$ finds itself in node $u$'s *forward list*, it starts to make its own *forward list* ($F_{\text{v}}$) from a subset of its one-hop neighbors ($B_{\text{v}}$) to cover its uncovered two-hop neighbors ($U_{\text{v}}$). The sets $U_v$ and $B_v$ is calculated using the following equations:

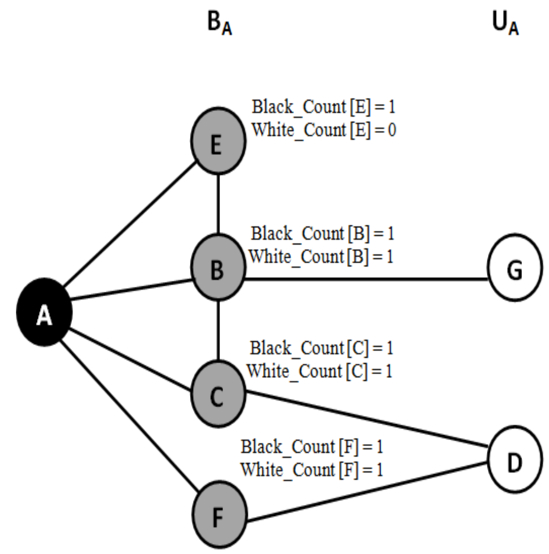
$$U_{\text{v}} = N(N(v)) - N(v) - N(u) \tag{3.6}$$

$$B_{\text{v}} = N(v) - N(u) \tag{3.7}$$

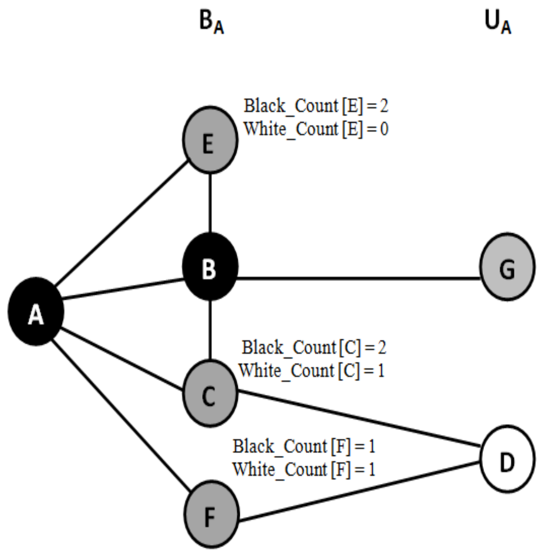
Figure 3.7 is the redrawing of Figure 3.6 for better visualization in order to determine the *forward_list* of node A. $B_{\text{A}}$ consists of node B, C, E and F and $U_{\text{A}}$ consists of node D and G. In DP, node A will ask node B to forward the message to cover node G and ask node C to cover node D. And there will arise contention between node B and node C when they will start to rebroadcast as they are both in the transmission range of each other. Our proposed algorithm aims at minimizing this type of contention. The algorithm will not select node C to cover node D once node A decides to choose node B to cover node G. It will ask node F instead to cover node D to avoid contention.

The motivation of our proposed algorithm is as follows: when a node $v$ receives a packet from node $u$, node B makes the $F_{\text{v}}$ from set $B_{\text{v}}$ to cover set $U_{\text{v}}$ so that there exists minimum or no contention among the forwarding nodes. The selection process of *forward list* of the new approach is stated in Algorithm 2.

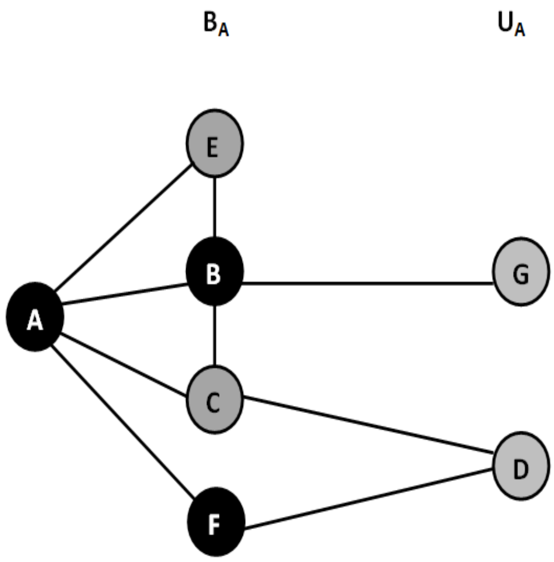
In the algorithm, an array named $Black\_Count$ is used so that each node $w \in B_{\text{v}}$ can keep record of its adjacent nodes in $B_{\text{v}}$ that are already selected as a candidate of *forward_list*. At




(a): Connection of node A with $B_A$ and $U_A$




(b): Node B is selected for $F_A$




(c): Node F finally selected for $F_A$

Figure 3.7: Step by step construction of Forward_list of node A

the start of constructing $F_v$, the corresponding value in the $Black\_Count$ array of all the nodes $w \in B_v$ will be 1. It will be increased by 1 whenever one of its neighbors in $B_v$ will be selected as a candidate of $F_v$. The nodes whose corresponding value of the $Black\_Count$ is minimum among all the nodes in $B_v$ will be placed to $Candidate\_Set$. Then in each iteration, a node $p \in Candidate\_Set$ is selected who covers maximum number of nodes in $U_v$ i.e $\|N(p) \cap U_v\|$ is maximized. Next, node $v$ includes node $p$ in its $F_v$ and the corresponding value of the $Black\_Count$ of the nodes in $B_v$ adjacent to $p$ will be increased by 1. Node $v$ will then set $U_v = U_v - N(p)$ and $B_v = B_v - p$. This process is iterated until all nodes in $U_v$ is covered or no change in $B_v$ set is possible.

Consider the illustration of the previous example, In our proposed algorithm when node A receives a packet and starts to making its forward list, The two hop neighbor of node A is:

$$N(N(A)) = \{A, B, C, D, E, F, G\}$$

The one hop neighbor of node A is:

$$N(A) = \{A, B, C, E\}$$

The U and B set of node A is:

$$U_A = \{D, G\} \tag{3.8}$$

$$B_A = \{B, C, E, F\} \tag{3.9}$$

The black_count of node B,C,E and F is

$$Black\_Count[B] = 1$$

$$Black\_Count[C] = 1$$

$$Black\_Count[E] = 1$$

$$Black\_Count[F] = 1$$

The scenario is illustrated in Figure 3.7(b). So, node B,C,E and F will make the $Candidate\_Set$.

$$Candidate\_Set = \{B, C, E, F\} \tag{3.10}$$

Among them node B covers node G in $U_A$. So B is selected and included in the forward list $F_A$. The adjacent nodes of B in $B_A$ is node C. Therefore, the counter of node C will be increased by 1 and node B is deducted from $B_A$ and node G will be deducted from $U_A$. Now, node C,E and

**Algorithm 2** Forward_list of a node

FORWARD_LIST (node v)

$F_v = \emptyset$, $size\_of\_forward\_list = 0$,
**for all** node $p \in B_v$ **do**
  $Black\_Count[p] = 1$;
**end for**
**while** $U_v \neq \emptyset$ or $B_v$ remains unchanged **do**
  $maximum = -1$, $minimum = 100000$, $Candidate\_Set = \emptyset$;
  **for all** node $q \in B_v$ **do**
    **if** $Black\_Count[q] < minimum$ **then**
      $minimum = Black\_Count[q]$;
    **end if**
  **end for**
  **for all** node $r \in B_v$ **do**
    **if** $Black\_Count[r] == minimum$ **then**
      $Candidate\_Set = Candidate\_Set \cup \{r\}$;
    **end if**
  **end for**
  **for all** node $s \in Candidate\_Set$ **do**
    **for all** node $t \in U_v$ **do**
      **if** node $t \in N(s)$ **then**
        $White\_Count[s] = White\_Count[s] + 1$;
      **end if**
    **end for**
  **end for**
  **for all** node $i \in Candidate\_Set$ **do**
    **if** $White\_Count[i] > maximum$ **then**
      $maximum = White\_Count[i]$;
      $x = i$;
    **end if**
  **end for**
  **if** $maximum > 0$ **then**
    $F_v[size\_of\_forward\_list] = \{x\}$;
    $size\_of\_forward\_list = size\_of\_forward\_list + 1$;
    $U_v = U_v - N(x)$;
    **for all** node $y \in (B_v \cap N(x))$ **do**
      $Black\_Count[y] = Black\_Count[y] + 1$;
    **end for**
    $B_v = B_v - \{x\}$;
  **end if**
**end while**

F remains in $B_{\text{A}}$ set to cover node D remains in $U_{\text{A}}$. The remaining $U_{\text{A}}$ and $B_{\text{A}}$ after selecting node B,

$$U_{\text{A}} = \{D\} \tag{3.11}$$

$$B_{\text{A}} = \{C, E, F\} \tag{3.12}$$

corresponding value of the $Black\_Count$ of the nodes in $B_{\text{A}}$

$Black\_Count[C]$ = 2

$Black\_Count[E]$ = 1

$Black\_Count[F]$ = 1

The corresponding value of node E and F in $Black\_Count$ is minimum, but E does not cover any node in $U_{\text{A}}$ so, node F is selected to cover node D.

$$U_{\text{A}} = \{\emptyset\} \tag{3.13}$$

So, the *forwarding list* of node A is:

$$F_{\text{A}} = \{B, F\} \tag{3.14}$$

In this way, node C will never select to be a part of node A's *forward list* to cover node F when there is an option to cover that node by selecting another forwarding node for avoiding the contention.

### 3.4.5 Complexity Analysis of Distributed CACDS

Constructing the $forwarding\_list$ of a node $v$ is very much similar to set cover problem. We have to choose nodes from set $B_{\text{v}}$ to cover the nodes belongs to set $U_{\text{v}}$. In the worst case scenario, the $Candidate\_Set$ equals to set $B_{\text{v}}$. So, we have to execute the loop from line 17-23 for $O(|B_{\text{v}}|)$ times and for each node in $B_{\text{v}}$, we have to check each node in $U_{\text{v}}$ to determine the reachability among the nodes with the loop executes from line 18-22 which will again run in $O(|U_{\text{v}}|)$ times in worst case scenario. So, the final run time of constructing the $forwarding\_list$ of node $v$ is $O(|U_{\text{v}}||B_{\text{v}}|)$.

# Chapter 4

# Detailed Analysis of the Algorithms

In this chapter, we analysis the state-of-art algorithms and our proposed algorithms based on a sample network of 10 nodes (Node A - Node J). At first, we analyze the selection process of the centralized algorithms- MCDS and Centralized CACDS construction and then we proceed further with the distributed algorithms- DP and Distributed CACDS. The one and two hop neighbors of each node is given in the following Table 4.1.

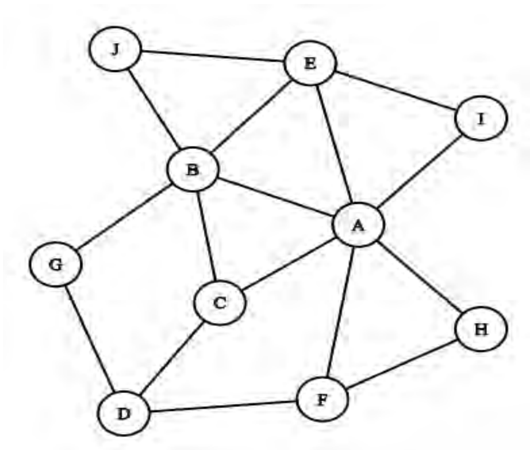



Figure 4.1: A random example scenario of 10 nodes

## 4.1 Analysis of the Centralized Algorithms

In this section, we first present the steps of constructing the MCDS as well as our new proposed Centralized CACDS.

Table 4.1: 1-hop and 2-hop neighbors of each node of the scenario in Figure 4.1

| $v$ | $N(v)$ | $N(N(v))$ |
|---|---|---|
| A | {A,B,C,E,F,H,I} | {A,B,C,D,E,F,G,H,I,J} |
| B | {A,B,C,E,G,J} | { A,B,C,D,E,F,G,H,I,J} |
| C | { A,B,C,D} | { A,B,C,D,E,F,G,H,I,J} |
| D | {C,D,F,G} | {A,B,C,D,F,G,H} |
| E | {A,B,E,I,J} | {A,B,C,E,F,G,I,J} |
| F | {A,D,F,H} | {A,B,C,D,E,F,G,H,I} |
| G | {B,D,G} | {A,B,C,D,E,F,G,J} |
| H | {A,F,H} | {A,B,C,D,E,F,H,I} |
| I | {A,E,I} | {A,B,C,E,F,H,I,J} |
| J | {B,E,J} | {A,B,C,E,G,I,J} |

### 4.1.1 Details Analysis of MCDS Algorithm

Table 4.2 illustrate the step by step construction of a MCDS for a given scenario of Figure 4.1. At first assume all nodes are colored WHITE and node A in selected and colored as BLACK as the degree of A is maximum among all the nodes and all its one hop neighbors {B, C, E, F, H, I} are colored as GRAY. Among the GRAY nodes, node B has the maximum number of WHITE neighbors which is 2 here, so, node B is selected and colored as BLACK and all its WHITE neighbors {G, J} will be colored as GRAY. Among the GRAY nodes, node C is then selected and colored BLACK to cover node D. After selecting node C, there is no WHITE node left in the network, so the algorithm terminates and the finally node A, B and C construct MCDS.

### 4.1.2 Detailed Analysis of Centralized CACDS Algorithm

At the start of the algorithm, All the nodes are colored WHITE. As node A has the maximum node degree among all so, node A will be colored as BLACK and all its one hop neighbors {B, C, E, F, H, I} are colored as GRAY. All the GRAY nodes has only one BLACK neighbor which

Table 4.2: Details analysis of MCDS algorithm for the scenario in Figure 4.1

| BLACK nodes | GRAY Nodes | Number of WHITE Neighbors of GRAY nodes | Selected Node | Remarks |
|---|---|---|---|---|
| A | B | 2{G,J} | B | |
| | C | 1{D} | | |
| | E | 1{J} | | |
| | F | 1{D} | | |
| | H | 0 | | |
| | I | 0 | | |
| A,B | C | 1 {D} | C | Node F or G also meets the criteria, here node C is selected as it has the lowest ID among all |
| | E | 0 | | |
| | F | 1{D} | | |
| | G | 1{D} | | |
| | H | 0 | | |
| | I | 0 | | |
| | J | 0 | | |
| A,B,C | D | 0 | N/A | No White node exists, so the algorithm terminates |
| | E | 0 | | |
| | F | 0 | | |
| | G | 0 | | |
| | H | 0 | | |
| | I | 0 | | |
| | J | 0 | | |
| The forwarding nodes are {A,B,C} | | | | |

is A so node B, C, E, F, H, I will be temporary selected and among the temporary selected nodes, node B has the maximum number of WHITE neighbors which is 2 here, so, node B will be selected and colored as BLACK and all its WHITE neighbors {G,J} will be colored as GRAY. Now, among the gray nodes, node F, G and J has minimum number of BLACK neighbors (which is 1 here) whereas node C and node E has two black neighbors each. So, node F, G and J is forwarded for next consideration. The remaining WHITE node D is reachable from either node F or node G. Node F is selected. After selecting node F and coloring all its neighbors GRAY, there is no WHITE node left in the network. Finally the BLACK nodes together will create the CACDS. Table 4.3 illustrate the step by step construction of a CACDS for a given scenario of Figure 4.1.

### 4.1.3 Analysis of MCDS and Centralized CACDS

If we compare Table 4.2 and Table 4.3, we can see that the nodes selecting to construct the connected dominating set is different. In tradition MCDS, node A, B and C is selected as a forwarding nodes and in Centralized CACDS, node A, B and F is selected for a sample scenario of Figure 4.1.

Because of selecting node C as a member of CDS after selecting node A and B, there occurs contention between node B and C as node B and C are within their transmission range. As we know, after broadcasting a message if, many of its neighbors try to rebroadcast it, if they are withing their transmission range, they will content with each other for the media. So, we can surely say that, the nodes selected by MCDS algorithm, will content when they will broadcast the message.

Our proposed algorithm Centralized CACDS, selects node F instead of node C to construct the CDS. Because of selecting node F, we can easily avoid the contention which was occurring because of selecting node C in MCDS. As node F is not within the transmission range of both node A and node B, so there will occur no contention when the nodes will broadcast the message.

Table 4.3: Details analysis of Centralized CACDS algorithm for the scenario in Figure 4.1

| **BLACK nodes** | **GRAY Nodes** | **Number of BLACK Neighbors of GRAY nodes** | **Candidate Set** | **Number of White neighbors of the temporarily selected nodes** | **Selected Nodes** | **Remarks** |
|---|---|---|---|---|---|---|
| A | B | 1 | {B,C,E,F,H,I} | $N(B) \cap ColorW$ =2 {G,J} | B | |
| | C | 1 | | $N(C) \cap ColorW$ =1 {D} | | |
| | E | 1 | | $N(E) \cap ColorW$ =1 {J} | | |
| | F | 1 | | $N(F) \cap ColorW$ =1 {D} | | |
| | H | 1 | | $N(H) \cap ColorW$ = $\emptyset$ | | |
| | I | 1 | | $N(I) \cap ColorW$ = $\emptyset$ | | |
| A,B | C | 2 | {F,G,J} | | F | Node G can also meets the criteri, here node F is selected as it has the lowest ID among all |
| | E | 2 | | $N(F) \cap ColorW$ =1 {D} | | |
| | F | 1 | | $N(G) \cap ColorW$ =1 {D} | | |
| | G | 1 | | $N(J) \cap ColorW$ = $\emptyset$ | | |
| | J | 1 | | | | |
| A,B,F | C | | | | | No White node exists, so the algorithm terminates |
| | D | | | | | |
| | E | | | | | |
| | G | | | | | |
| | H | | | | | |
| | I | | | | | |
| | J | | | | | |
| The forwarding nodes are {A,B,F} | | | | | | |

## 4.2 Analysis of the Distributed Algorithms

In this section, we will analysis the dominant pruning algorithm and our proposed distributed algorithm, and will present that our how our algorithm chooses the forward nodes to avoid contention which was not kept in mind while designing the DP algorithm. We will analysis both the algorithms for a scenario given in Figure 4.1 and the neighbor information of the aforementioned topology is shown in Table 4.1.

### 4.2.1 Detailed Analysis of DP Algorithm

As we know, in dominant pruning each node make its own *forward list* based on its two hop neighbor information before broadcast a message. Figure 4.1 shows as example of network. Suppose, node A is the broadcast initiator. Now, let us find out the forwarding list of node A.

$$U_{\text{A}} = N(N(A)) - N(\emptyset) - N(A) = \{D, G, J\}$$

$$B_{\text{A}} = N(A) - N(\emptyset) = \{B, C, E, F, H, I\}$$

Therefore, the forwarding list of node A will be $F_{\text{A}} = \{B, C\}$

Similarly, we can find a complete list of forwarding node list for all other nodes. Table 4.4 shows the complete analysis. Node A,B,C,G,D and node F will forward the packet, comprising a total of 6 forwarding to complete the broadcast.

Table 4.4: Detailed Analysis of Dominant Pruning algorithm for the scenario in Figure 4.1

| $u$ | $v$ | $U_{\text{v}}$ | $B_{\text{v}}$ | $F_{\text{v}}$ |
|---|---|---|---|---|
| $\emptyset$ | A | {D,G,J} | {A,B,C,E,F,H,I} | {B,C} |
| A | B | {D} | {G,J} | {G} |
| A | C | {G,J} | {D} | {D} |
| B | G | {F} | {D} | {D} |
| C | D | {H} | {F,G} | {F} |
| D | F | {B} | {A,H} | {A} |

### 4.2.2 Detailed Analysis of Distributed CACDS Algorithm

As dominant pruning, our proposed algorithm, each node make its own *forward list* based on its two hop neighbor information before broadcast a message. Like previous example, suppose node A starts the broadcast. Now, let us find out the forwarding list of node A based on proposed method.

$$U_{\text{v}} = N(N(A)) - N(\emptyset) - N(A) = \{D, G, J\}$$
$$B_{\text{v}} = N(A) - N(\emptyset) = \{B, C, E, F, H, I\}$$

The number of black neighbors of all the nodes in $B_{\text{A}}$ is 1, as node B covers maximum nodes in $U_{\text{A}}$, so node B is selected first. Then the number of black neighbors of rest of the nodes in $B_{\text{A}}$, which are adjacent to node B in updated. Among the nodes which can cover node D,node F has the minimum number of black neighbors, so node F is selected next.

Therefore, the forwarding list of node A will be $F_{\text{A}} = \{B, F\}$

Similarly, we can find a complete list of forwarding node list for all other nodes. Table 4.5 shows the complete analysis. Node A,B,F,G,D and node C will forward the packet, comprising a total of 6 forwarding to complete the broadcast.

Table 4.5: Detailed Analysis of Distributed CACDS algorithm for the scenario in Figure 4.1

| $u$ | $v$ | $U_{\mathbf{v}}$ | $B_{\mathbf{v}}$ | $F_{\mathbf{v}}$ |
|---|---|---|---|---|
| $\emptyset$ | A | {D,G} | {A,B,C,E,F,H,I} | {B,F} |
| A | B | {D} | {G,J} | {G} |
| A | F | {G} | {D} | {D} |
| B | G | {F} | {D} | {D} |
| F | D | {H} | {C,G} | {C} |
| D | C | {H,I,J} | {A,B} | {A,B} |

### 4.2.3 Analysis of DP and Distributed CACDS

If we analyze Table 4.4 and Table 4.5, we can see that, in both the cases, a total of 6 nodes need to forward the message to reach it to the whole network.

In Table 4.4, while selecting the $forward\_list$ of node A, DP selects node B and node C whereas our proposed algorithm selects node B and node F. After forwarding a message by node A, all the nodes in its $forward\_list$ will rebroadcast the message. If node B and node C try to rebroadcast the message simultaneously (as they are in the forward list of node A selected by DP), they will contend with each other for the medium as they are within each other's transmission range.

Therefore, in our proposed algorithm, by selecting node F, instead of node C , this contention is minimized. Selecting node B and node F for $forward\_list$ of A, there will occur no contention when the nodes will rebroadcast the message upon receiption of node A.

If we, carefully analyze Table 4.4 and Table 4.5, we can see that, the selected forwarding nodes are same in both the cases. Node A, B, C, D, F and G is selected as forwarding nodes in both the cases, but they will forward in different sequence. To be precise, node C is in the $forward\_list$ of node D in our proposed algorithm whereas node C is in the $forward\_list$ of node A in DP. If node C forward with node B simultaneously after receiving node A, then there will arise contention. However, if node C forward the message later in the network it will not face any contention. So, we can see that, by intelligently selecting nodes in $forward\_list$ we can easily mitigate the contention problem.

# Chapter 5

# Simulation and Performance Evaluation

To see the effectiveness of the proposed algorithms, the algorithms:- Centralized and Distributed CACDS are implemented along with MCDS and Dominant Pruning algorithm. The centralized algorithm is considered as a benchmark for evaluating the performance of the distributed algorithm. Finally we perform a comparative analysis based on the simulation results.

## 5.1 Simulation Environment

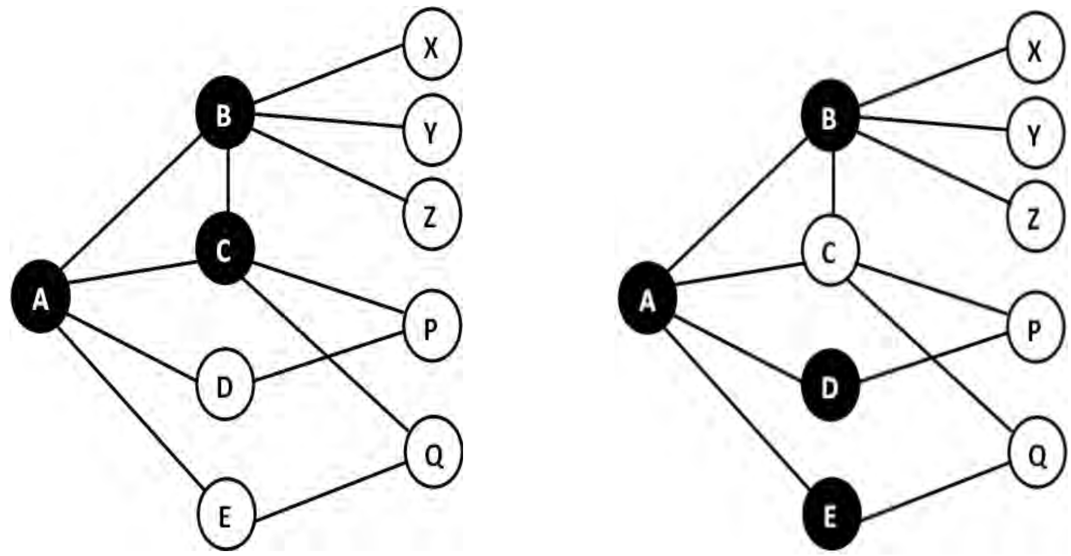
To evaluate the performance, we simulate randomly deployed networks of 100-500 nodes over a fixed 625m X 625m square region. The network generated each time is a connected network. Depending on the node density, the random scenarios can be broadly categorized into sparse, moderately dense and dense networks. The maximum transmission radius is limited between 125m to 225m. Each node is placed randomly in the simulation space for testing purpose. The nodes remained in their position for the whole time. We choose node 1 each time to initiate the broadcast packet for distributed algorithms. Any random node could have been selected to initiate the broadcast, but this does not matter as the position of node 1 is randomly placed.The simulation code-base was built using C++ programming language.

## 5.2 Performance Metrics

While evaluating the algorithms, we choose number of forwarding nodes and number of contention occured as performance metrics.

- **Number of forwarding nodes:**

  Number of forwarding nodes can also be depicted as size of the CDS (Connected Dominating Set). We know, use of CDS as a virtual backbone of a network was introduced in order to reduce the redundancy problem. In our proposed algorithms, we are constructing CDS such a way to minimize contention, we may need to sacrifice a node which will cover the maximum nodes in the network. For example, suppose in a network there are 5 remaining nodes which need to be covered by 4 temporarily selected nodes. Among the temporarily selected nodes, may be there is a node which can cover the remaining uncovered nodes alone, but if that node is selected, contention may arise. On the other hand, the other two nodes from temporarily selected nodes can jointly cover all the remaining nodes, so to avoid contention we will select the other two nodes as a member of CACDS instead of that one node. Therefore, the number of forwarding node will increase in order to mitigate contention. So, we will inspect the total number of forwarding nodes to analyze how many extra transmissions are needed to mitigate contention. The scenario is depicted in Figure 5.1




(a): Selection of forwarding nodes to minimize redundancy (b): Selection of forwarding nodes to minimize contention

Figure 5.1: Effect of algorithms in term of forwarding nodes

If we see the following figure, node A will select node B to cover node X,Y and Z. After selecting node B, it will select node C to cover node P and Q to minimize the number of forwarding nodes. So, the number of forwarding nodes is 3 here. However, because of selecting C, there will occur contention between node B and node C as they are within their transmission range. So, our algorithm will choose node D to cover node P and node E to cover node Q. So, the number of forward nodes of node A increases to 4 in our algorithm.

- **Number of Contention:** Upon receiving a packet from node $u$, if node $v \in F_{\text{u}}$ , the node $v$ starts to make its own forward list. If one member of the $F_v$ is connected (within the transmission range) to another ones, then those nodes will face contentions while they participate in forwarding the packet. The number of contention occurrence can be determined as

$$NumberofContention = \sum_{w \in F_{\text{v}}} \left( \left| (N(w) - \{w\}) \cap F_{\text{v}} \right| \right) \tag{5.1}$$

In Figure 5.1(a), because of selecting node B and C for the $forward\_list$ of node A, there occurs 1 contention, but in Figure 5.1(b), by selecting node B, D and E minimizes the contention and construct a CDS without contention.

## 5.3 Experimental Results regarding Number of Forward list

### 5.3.1 Effect of transmission range

In Figure 5.2 and Figure 5.3, we can see the effect of centralized and distributed algorithms in a sparse networks respectively. For this scenarios, we have run the simulation for 100 nodes having transmission range between 125m to 225m. For each scenario 10 different networks were generated and the mean value of the number of forwarding nodes are taken to plot in the graphs. The X-axis represents transmission range in meter and Y-axis represents the number of forwarding nodes. For better readability of the results, we have separated the graphs of centralized (MCDS and Centralized CACDS) and distributed (DP and distributed CACDS) algorithms. Traditional MCDS needs minimum number of transmission to broadcast a message in the whole

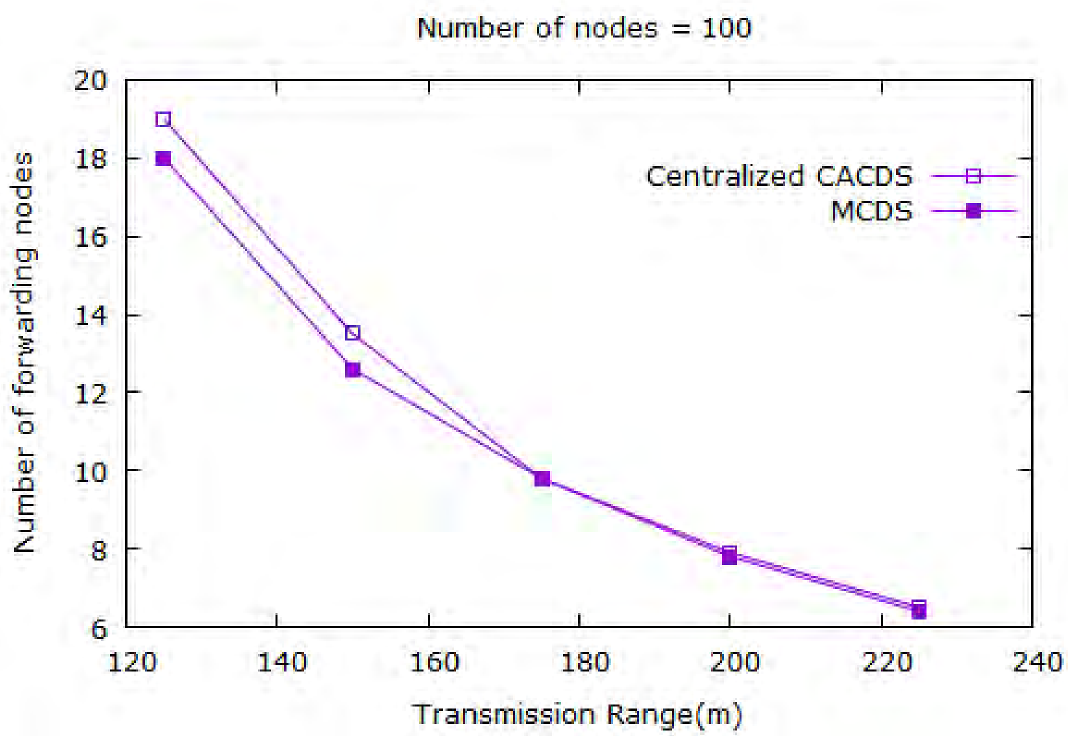



Figure 5.2: Performance comparison of MCDS, Centralized CACDS in term of number of forwarding for a sparse network

network. In Figure 5.2, we can see that MCDS needs only 6-18 broadcasts in average to forward the message in the whole network. Our Centralized CACDS also needs 6-19 broadcasts in average to deliver a message in the whole network. So, the size of CACDS and MCDS is almost same for this scenario. We can say that, the difference between the forwarding nodes are very little in these two scenarios.

Figure 5.3 depicts are result of distributed algorithms. Clearly, the result of using 2-hop information can not match the one using global network information. In DP, it needs almost 55-65 transmissions in average for forwarding a message in the whole network whereas our proposed algorithm needs 59-68 transmissions in average which means with having only 4-5 extra transmissions in average, we can minimize contention for a network of 100 nodes.

As the transmission range increases, mainly the number of forwarding decreases for all methods. It is expected because with the increasing transmission range, a node can cover more neighbors with a single forwarding than smaller transmission range.

Figure 5.4 and Figure 5.5 shows the effect of the algorithms for 300 nodes. Again, 10 scenarios were generated for each transmission range and the average value is taken to plot the graphs. In, Figure 5.4, when the transmission range in 125m, MCDS needs 20 transmissions

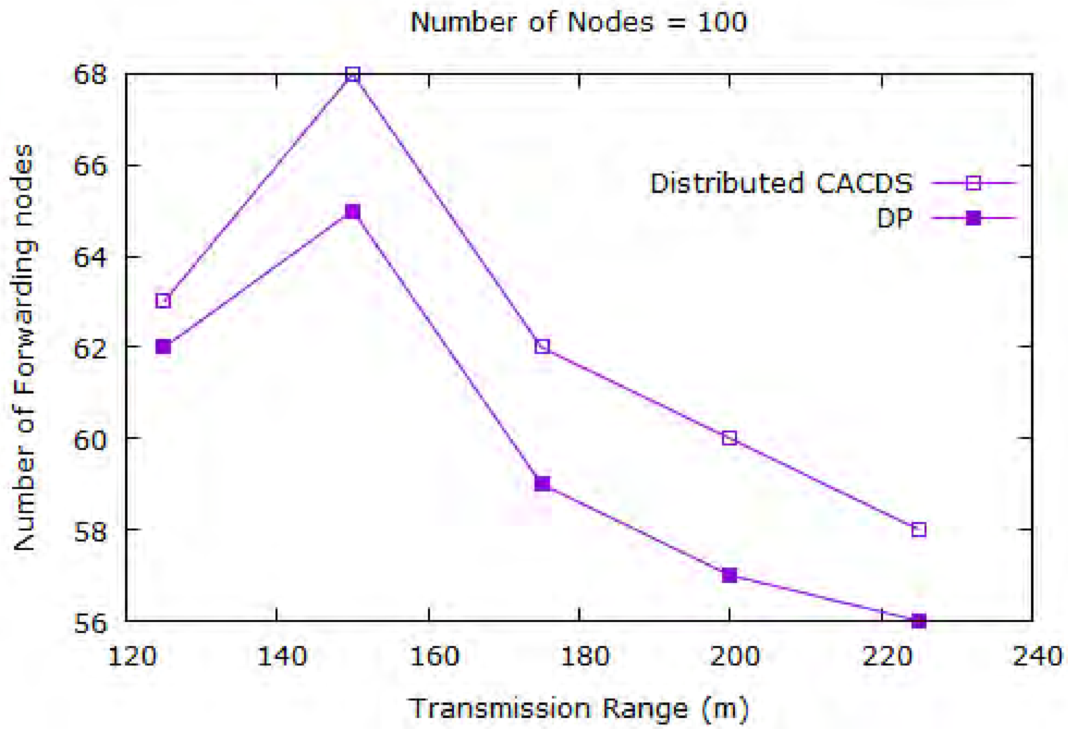



Figure 5.3: Performance comparison of DP, Distributed CACDS in term of number of forwarding for a sparse network

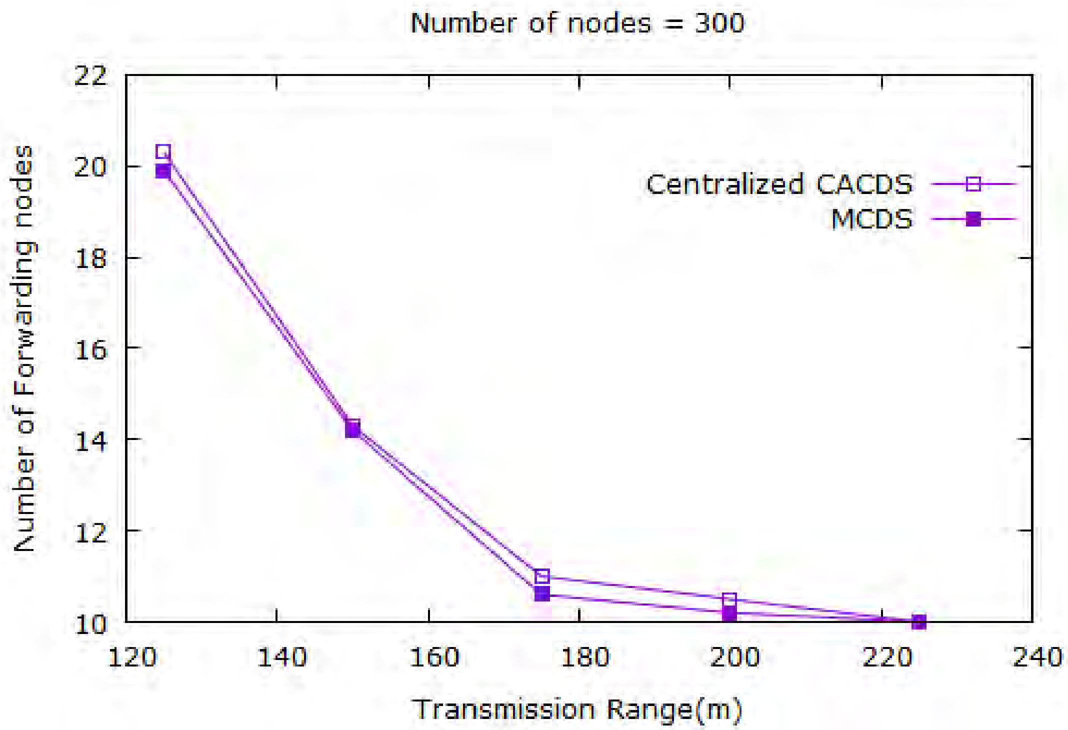



Figure 5.4: Performance comparison of MCDS and Centralized CACDS in term of number of forwarding for a moderately dense network

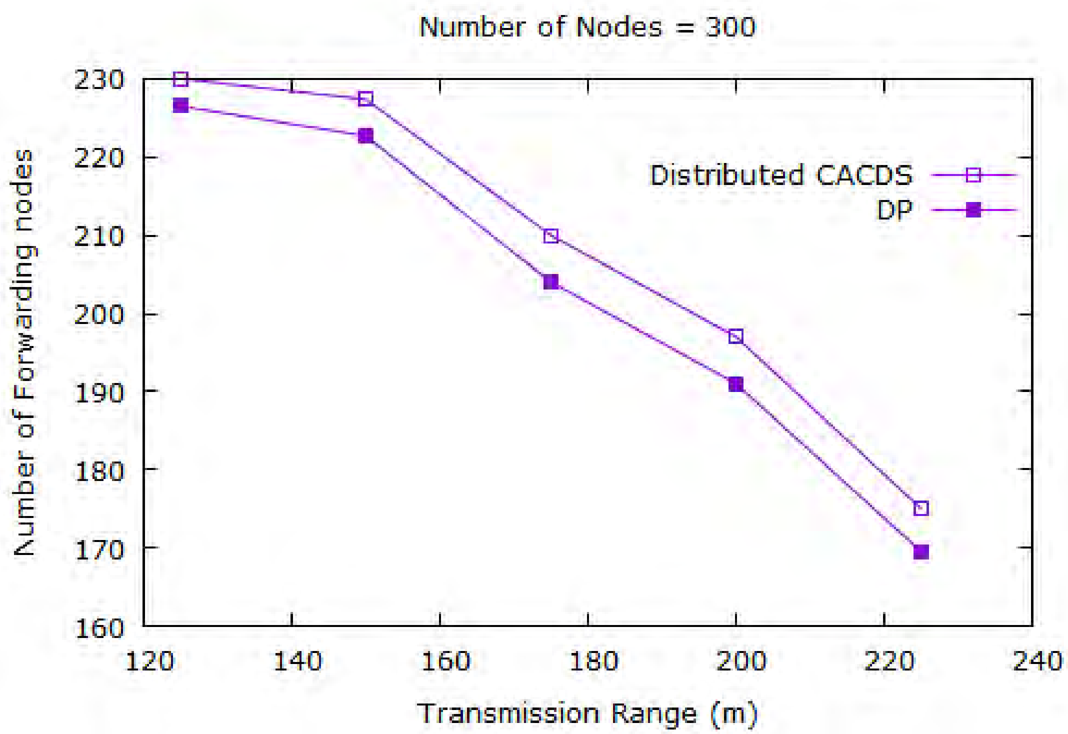



Figure 5.5: Performance comparison of DP and Distributed CACDS in term of number of forwarding for a moderately dense network

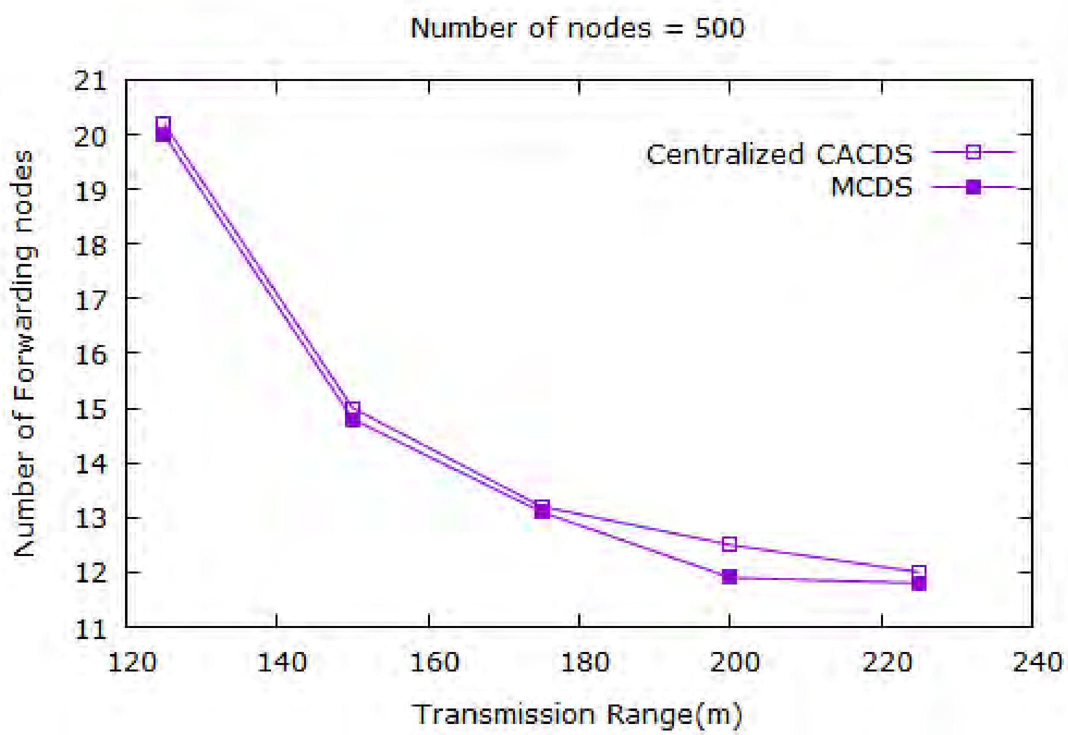



Figure 5.6: Performance comparison of MCDS and Centralized CACDS in term of number of forwarding for a dense network

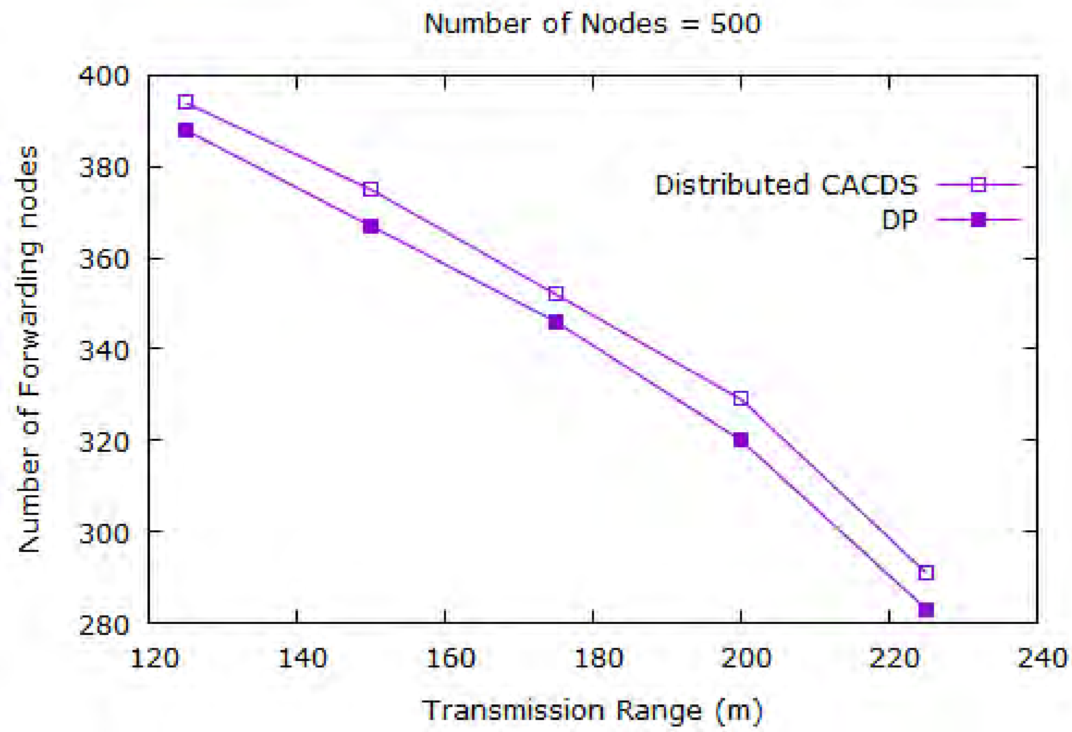



Figure 5.7: Performance comparison of DP and Distributed CACDS in term of number of forwarding for a dense network

where centralized CACDS needs 21 (only 1 extra) transmissions to broadcast a message in the whole network. From Figure 5.5, we can see that 225 transmissions are needed for DP and 230 transmissions are required in our proposed method. Again the difference between the number of forwarding nodes is not so high between the two algorithms.

Now, to see the effect of the methods, we have increased the total number of nodes by 500. Figure 5.6 and Figure 5.7 presents the effect of network with 500 nodes. In Figure 5.6, we can see that again the difference between the number of nodes are pretty much similar for centralized algorithms. For distributed algorithms, DP needs 280 transmissions where our proposed algorithm needs almost 288 transmissions for transmission range 225m to minimize contention.

We have seen the performance in term of number of forwarding nodes for all the algorithms in sparse, moderately dense and dense network. In order to minimize the contention, the increment in the number of forwarding nodes is clearly visible in the scenarios. However, for centralized algorithms, the difference is so little that it can easily be neglected. For distributed algorithms the difference between the number of forwarding nodes is also not very high. After analyzing all the scenarios, we can see that, there is a noticeable difference between the number

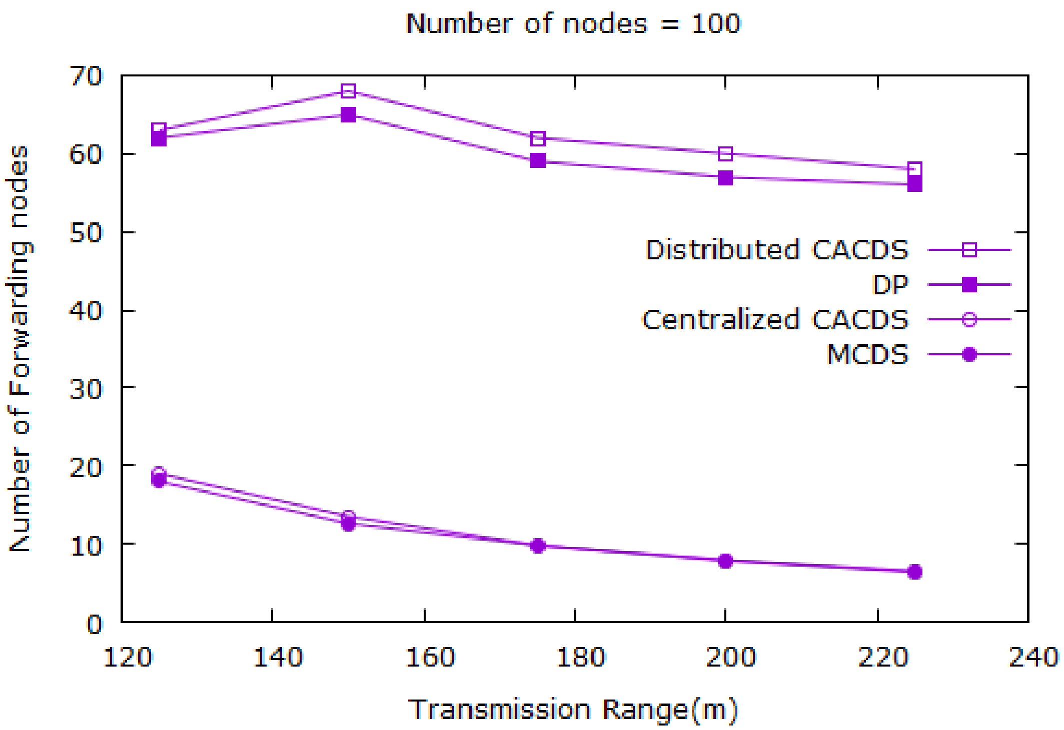



Figure 5.8: Performance comparison of MCDS, Centralized CACDS, DP and Distributed CACDS in term of number of forwarding for a sparse network

of forwarding nodes of centralized and distributed algorithms. Centralized algorithms need the whole topology to decide the forwarding nodes. So, the whole topology information is known, it is easy to decide which node should forward and which node should not. Therefore, it is so obvious that, centralized algorithms will always perform better than that of distributed algorithms. Clearly, the result of using 2-hop neighborhood information can not match with the one using global network information. Figure 5.8 depicts result of all methods for a network with 100 nodes. In centralized CACDS, it needs only 19 transmission, whereas is distributed CACDS it needs 63 transmissions to broadcast the message in the whole network with minimum contention for transmission range 125m.

### 5.3.2 Effect of Node Density

Figure 5.9 and Figure 5.10 shows the effect of node density on centralized and distributed methods respectively. In this case, the transmission range is kept fixed at 125m for all nodes. Each scenario has different number of nodes ranging from 100-500. The number of forward nodes increases for all the methods as the number of nodes increases in the network. The growth

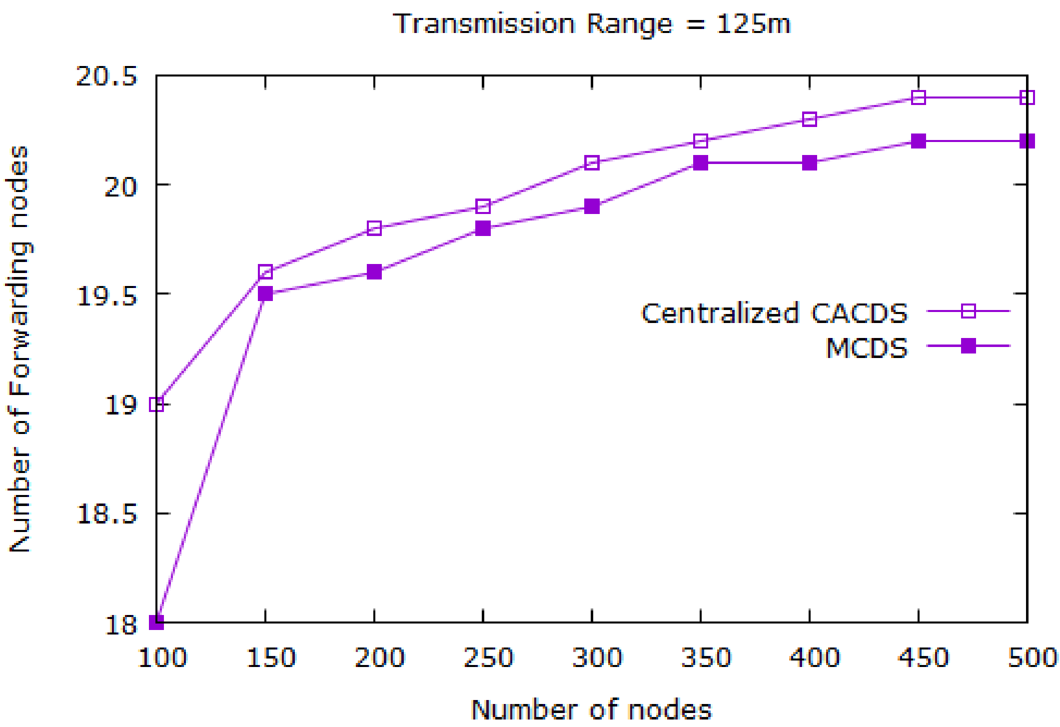



Figure 5.9: Performance comparison for forwarding nodes of MCDS, Centralized CACDS in term of node density

rate is almost linear but the difference of growth rate between two method is very small. The same scenario is shown in Figure 5.11 and Figure 5.12 setting the transmission range to 225m. In summary, in term of number of forwarding nodes our proposed algorithms shows almost similar performance with the state-of-art algorithms. The increase of forwarding nodes between the centralized algorithms is easily negligible and for distributed algorithms the increase in the number of forwarding list is also not very high.

## 5.4 Experimental Results regarding Number of Contention occurs

### 5.4.1 Effect of transmission range

Figure 5.13 and Figure 5.14 present the state of contention occurrence for a sparse network with 100 nodes for both the centralized and distributed algorithms respectively. This gives clear idea of how much efficient our algorithms work than that of state-of-art algorithms. Again, we

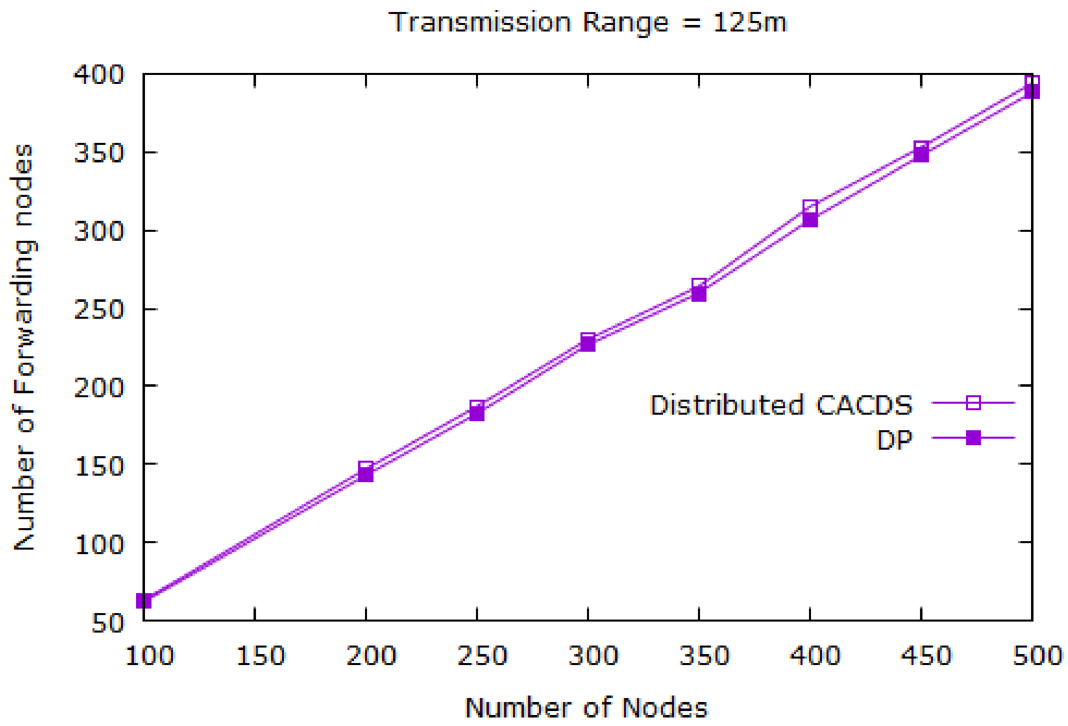



Figure 5.10: Performance comparison for forwarding nodes DP and Distributed CACDS in term of node density

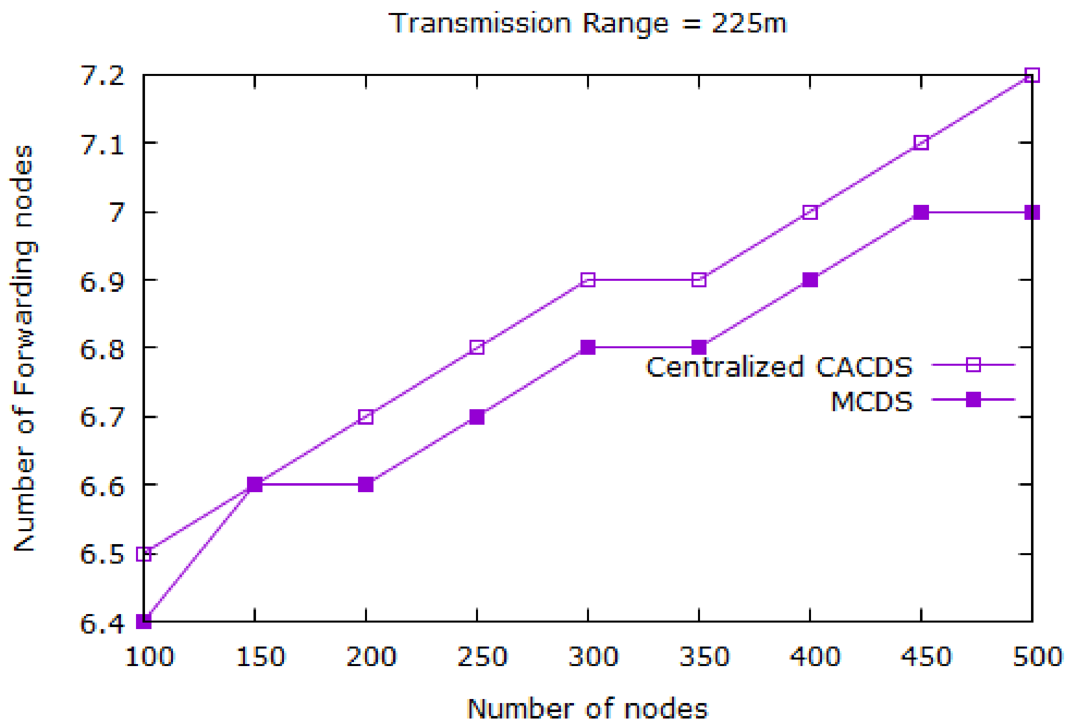



Figure 5.11: Performance comparison for forwarding nodes of MCDS, Centralized CACDS in term of node density

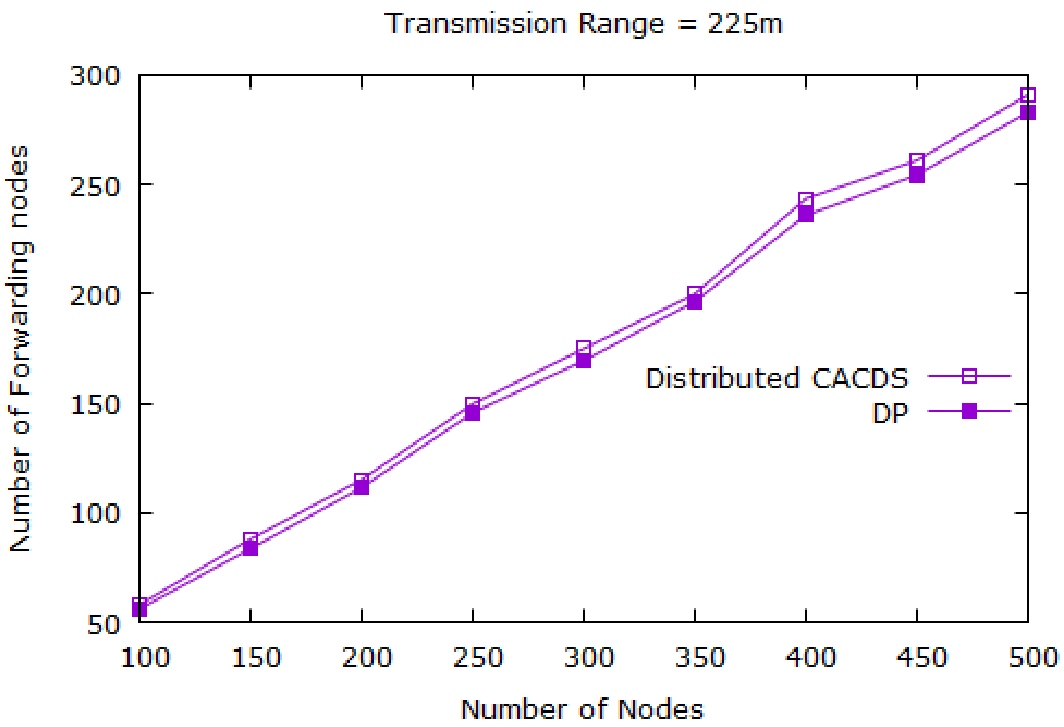



Figure 5.12: Performance comparison for forwarding nodes DP and Distributed CACDS in term of node density

have run the simulation for 100-500 nodes having transmission range between 125m to 225m. For each scenario 10 different networks were generated and the average value is taken of the number of occurred contention to plot in the graphs. The X-axis represents transmission range in meter and Y-axis represents the number of contention.

Figure 5.13 shows that the number of contention varies from 0.9 to 3.3 (in average) between the forwarding nodes in MCDS algorithm whereas it is minimized to 0-0.3 (in average) for our Centralized CACDS. It is noticeable that for transmission range 225m, our algorithm generates totally contention free CDS. The distributed scenario is presented in Figure 5.14. In distributed environment also, our proposed method performs better than that of DP. Number contention in forwarding nodes in DP varies from 51 to 65. In our method it decreases to 47 to 57. It almost avoids 6.75% contention occurrence in transmission range 175m and about 9.3% when the transmission range is 225m.

Figure 5.15 and Figure 5.16 shows the effect of methods for moderately dense network of 300 nodes. Our centralized algorithm generates contention free connected dominating set showing in Figure 5.15. In distributed environment it also shows quite good performance. For transmission range 150m, the number of occurred contention among the forwarding nodes is 760 in DP

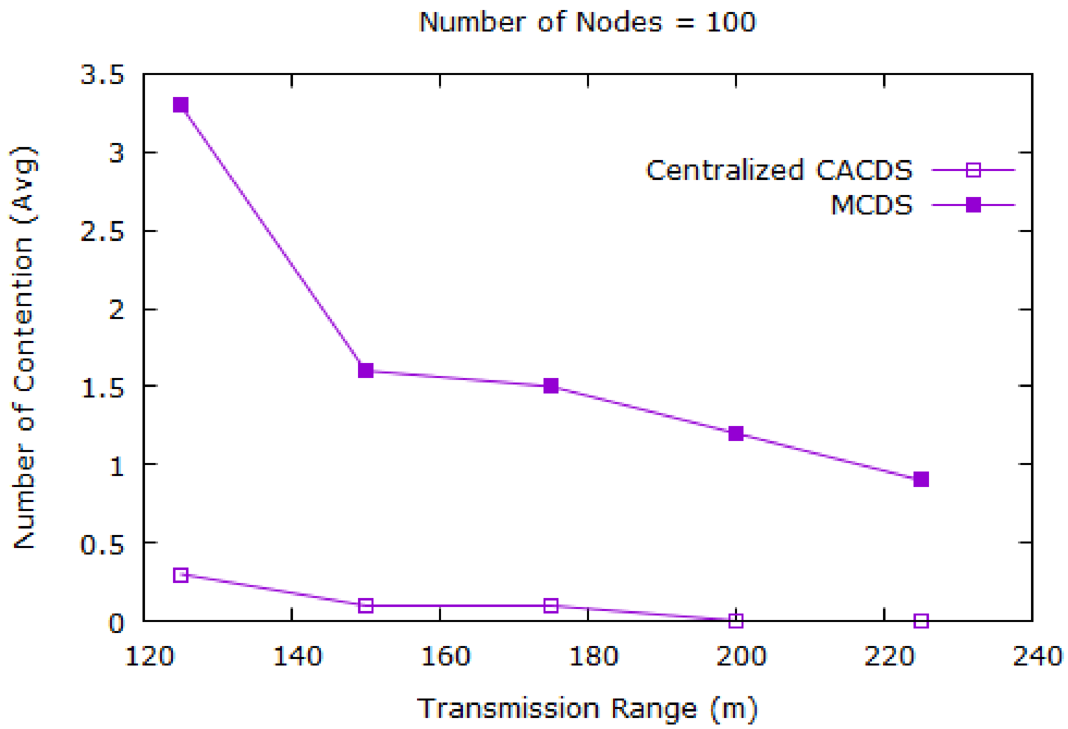



Figure 5.13: Performance comparison of MCDS and Centralized CACDS in term of number of contention occurs for a sparse network

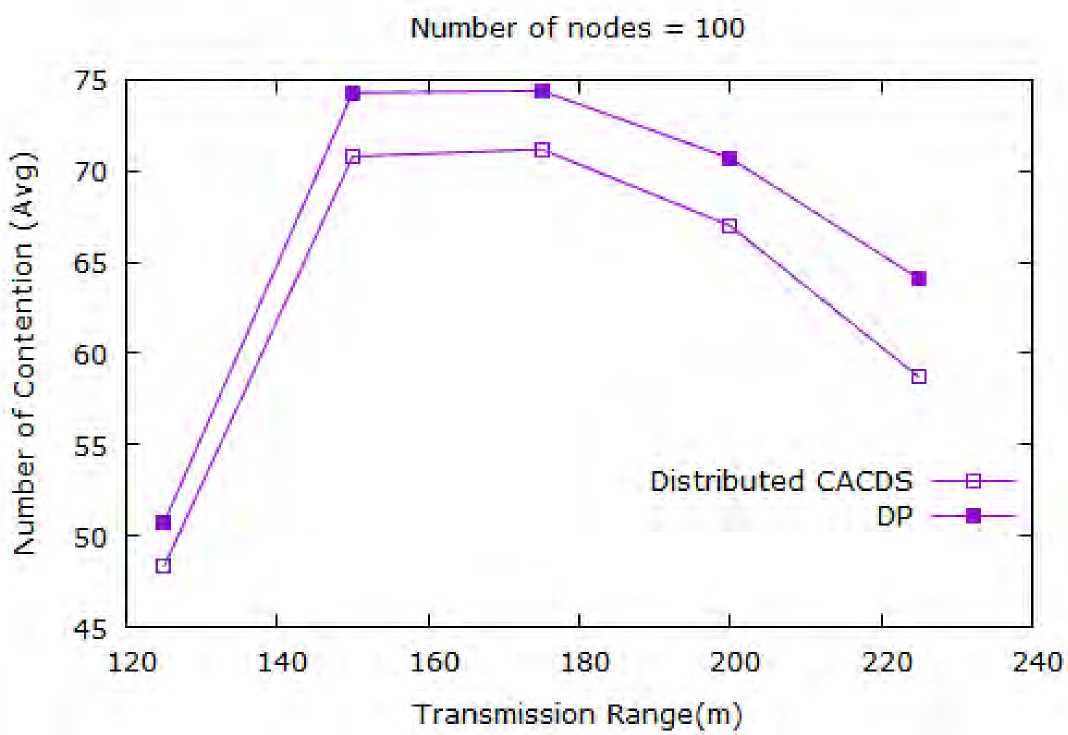



Figure 5.14: Performance comparison of DP and Distributed CACDS in term of number of contention occurs for a sparse network

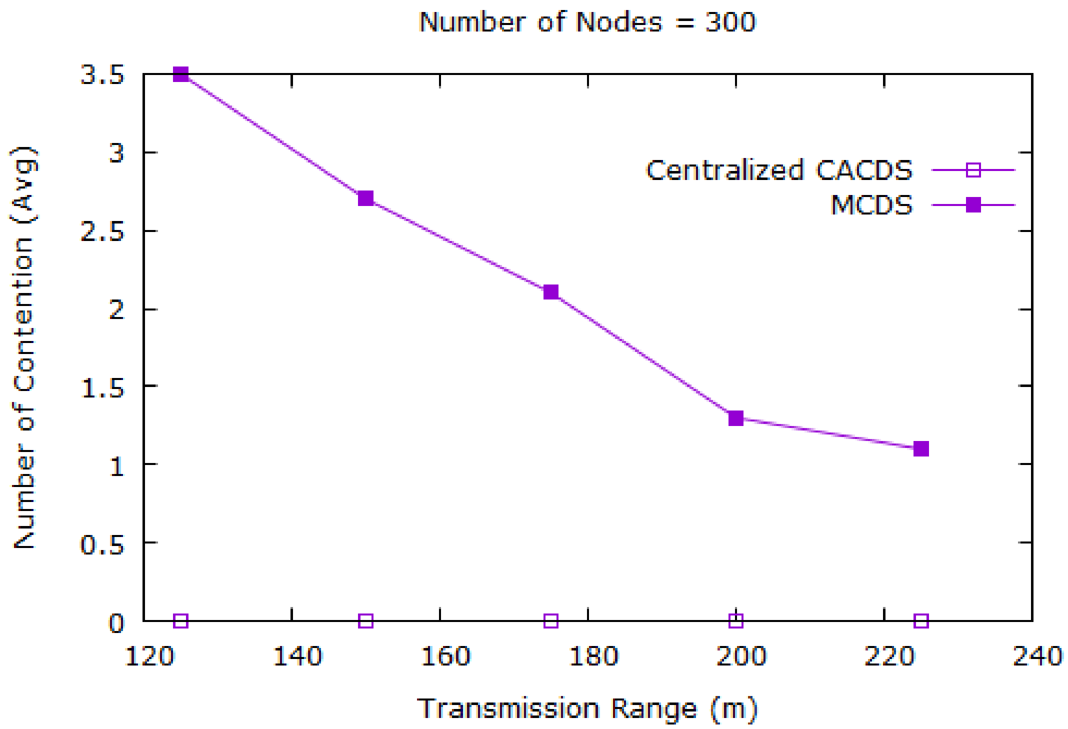



Figure 5.15: Performance comparison of MCDS, Centralized CACDS in term of number of contention occurs for a moderately dense network

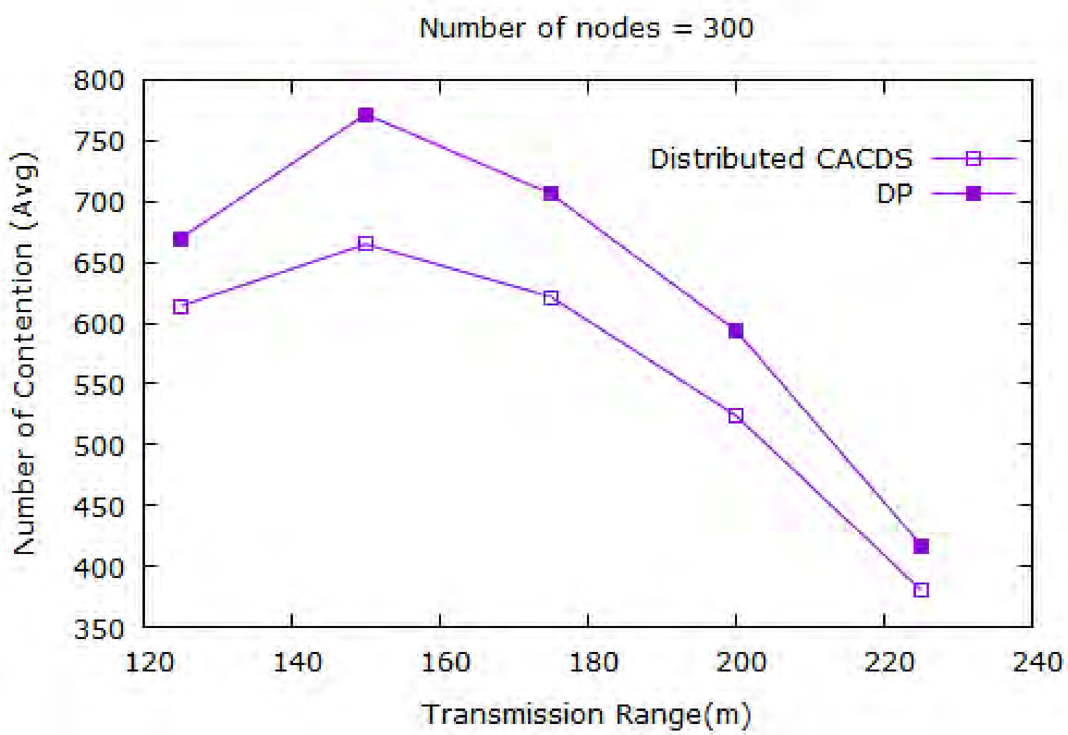



Figure 5.16: Performance comparison of DP, Distributed CACDS in term of number of contention occurs for a moderately dense network

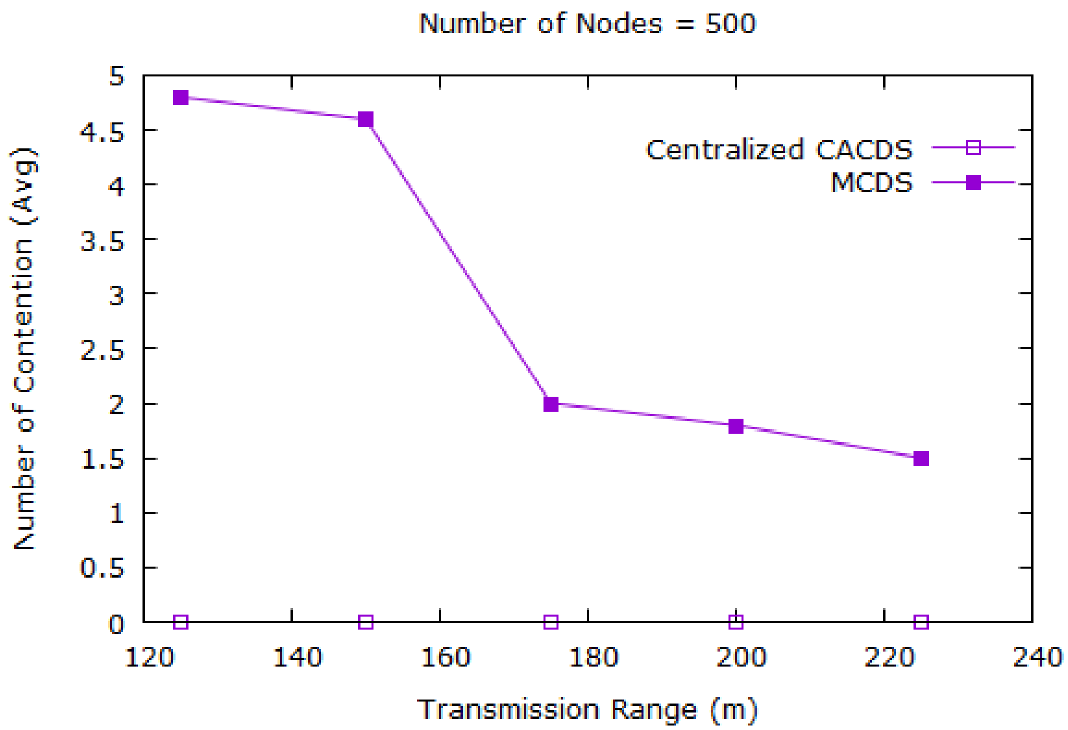



Figure 5.17: Performance comparison of MCDS, Centralized CACDS in term of number of contention occurs for a dense network

and 650 in our algorithm. It reduces the contention by almost 13%. The effects of the methods for a dense network is presented in Figure 5.17 and Figure 5.18. Our proposed algorithms performs better in both the environment. It minimizes almost 200 chances of contentions between the forwarding nodes in transmission range 150m.

Figure 5.19 illustrates a complete scenario how four algorithms work in a sparse network. Both the proposed methods perform better in term of minimizing contention. Centralized CACDS performs very well as it knows the whole topology information of the network, so it can easily make its choice of selecting forwarding nodes such a way to avoid contention. Distributed CACDS also performs better than DP to minimize contention.

### 5.4.2 Effect of Node Density

Figure 5.20 and Figure 5.21 show the effect of node density for centralized and distributed methods respectively. In this case, the transmission range is kept fixed to 125m for all nodes but each scenario has different number of nodes ranging from 100 to 500. It is clearly evident as the number of nodes increases in a network, number of contention also increases linearly for all the methods. Though distributed CACDS also gives linear increase with the increasing number of

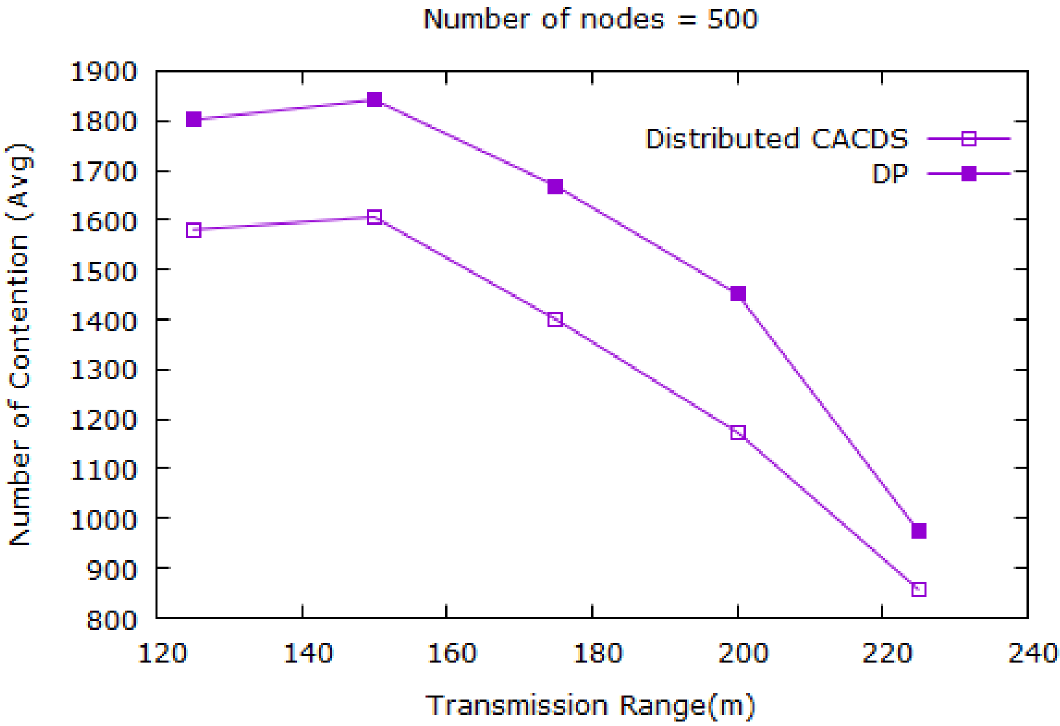



Figure 5.18: Performance comparison of DP, Distributed CACDS in term of number of contention occurs for a dense network

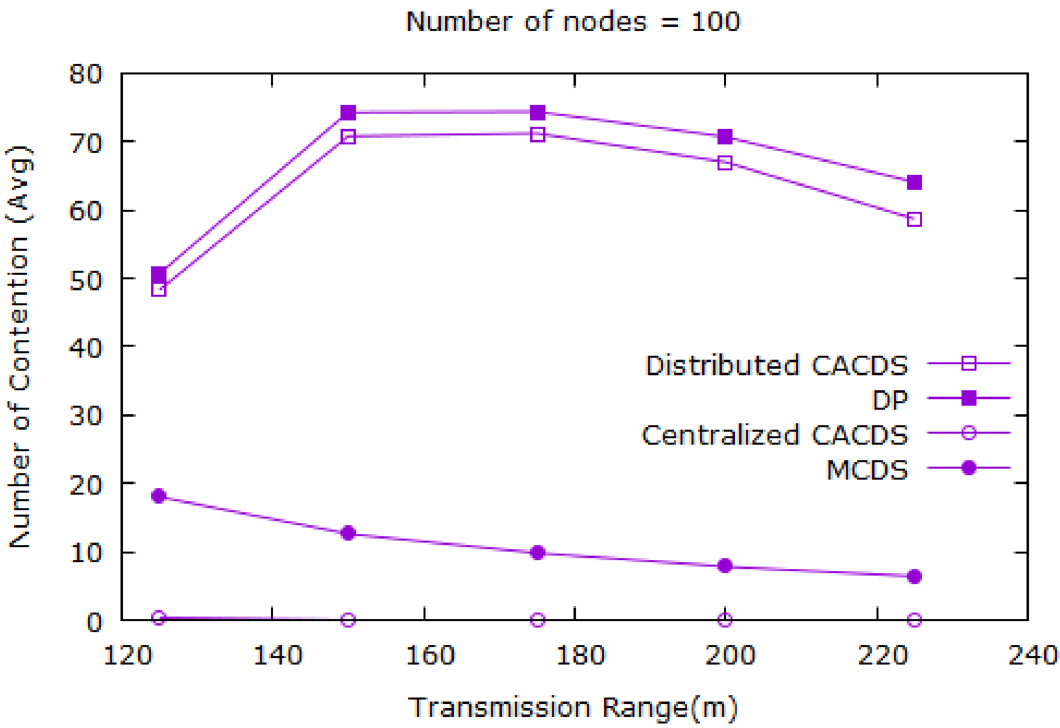



Figure 5.19: Performance comparison of MCDS, Centralized CACDS, DP and Distributed CACDS in term of number of contention for a sparse network

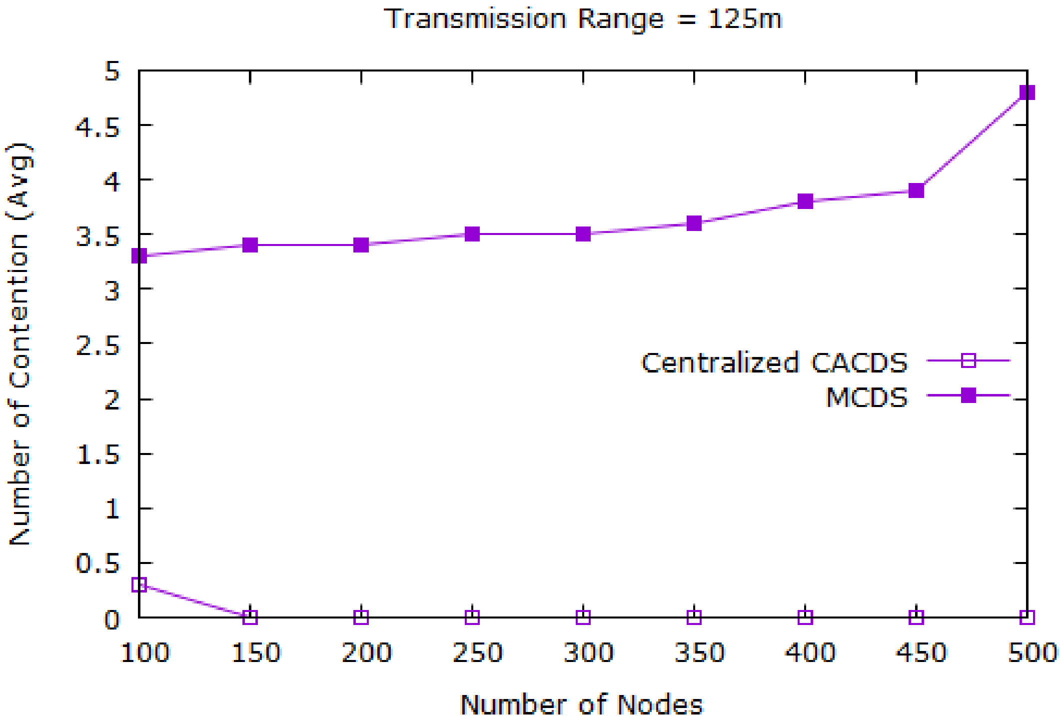



Figure 5.20: Performance comparison of MCDS and Centralized CACDS in term of number of contention occurs for transmission range 125m

nodes, its growth is less than that of DP. Thus, in case of node density out approach gives better result than DP. The same scenario is shown in Figure 5.22 and Figure 5.23 having transmission range set to 225m. Here also, our approach perform well.

In summary, the proposed method, both centralized and distributed gives better performance than MCDS and DP respectively by reducing number of contention.

## 5.5 Complexity Analysis of Centralized CACDS

In Figure 5.24, we have shown the execution time of Centralized CACDS and compare it with traditional graph of $O(N^2)$. We can see that, the growth of execution time of centralized CACDS with the increase of the number of forwarding nodes is almost same as that of $O(N^2)$.

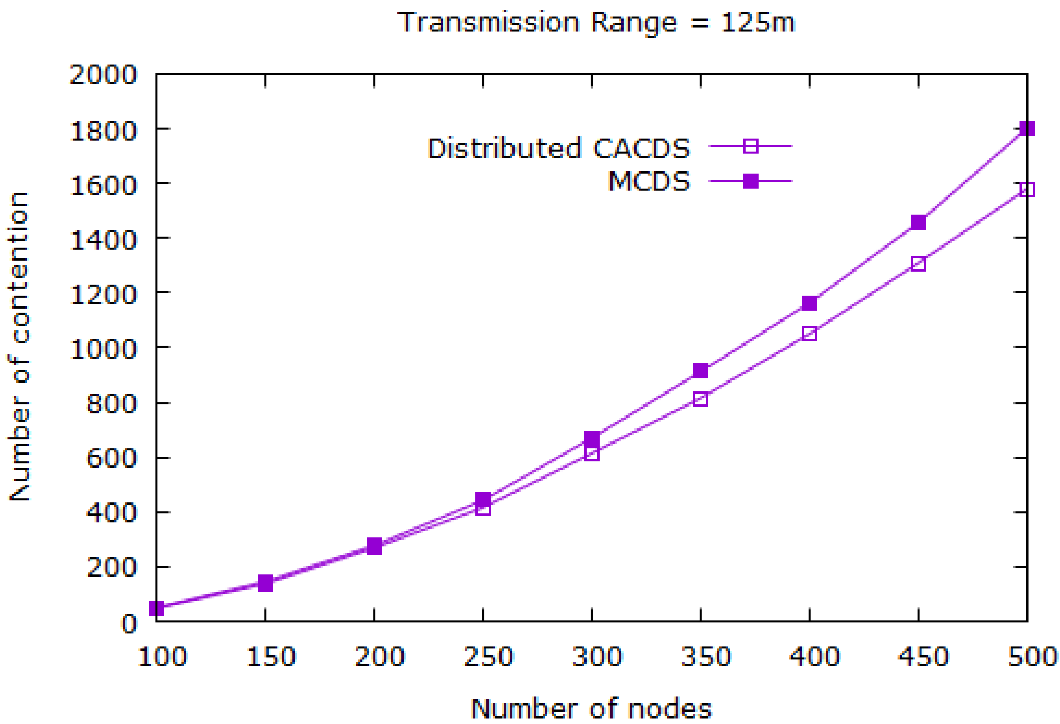



Figure 5.21: Performance comparison of DP and Distributed CACDS in term of number of contention occurs for transmission range 125m

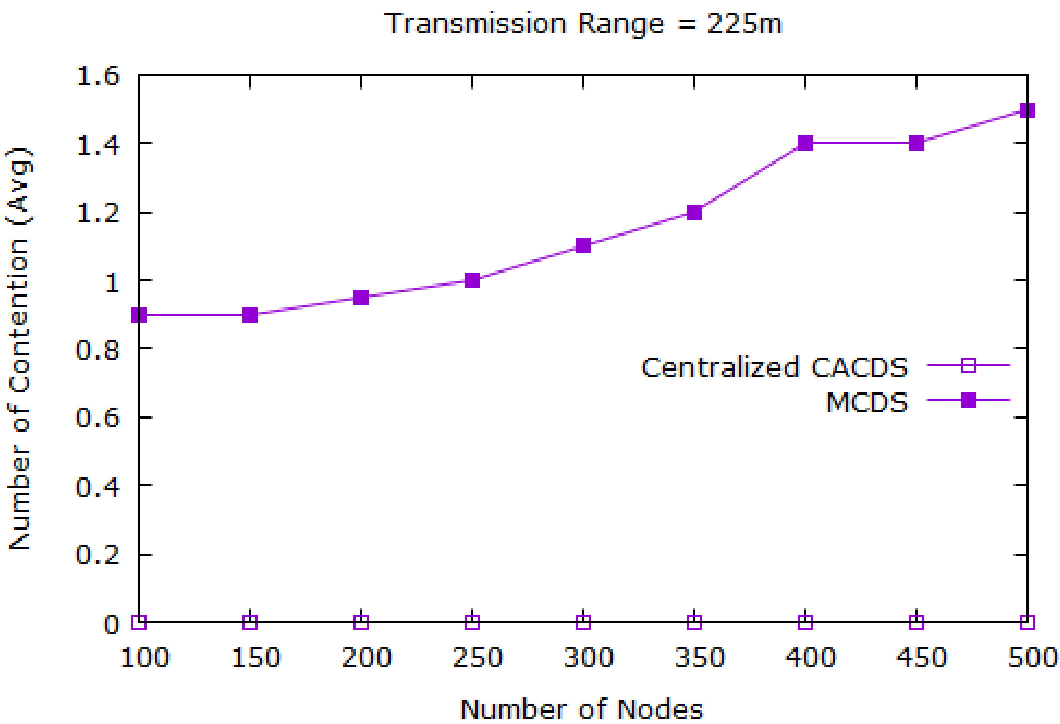



Figure 5.22: Performance comparison of MCDS and Centralized CACDS in term of number of contention occurs for transmission range 225m

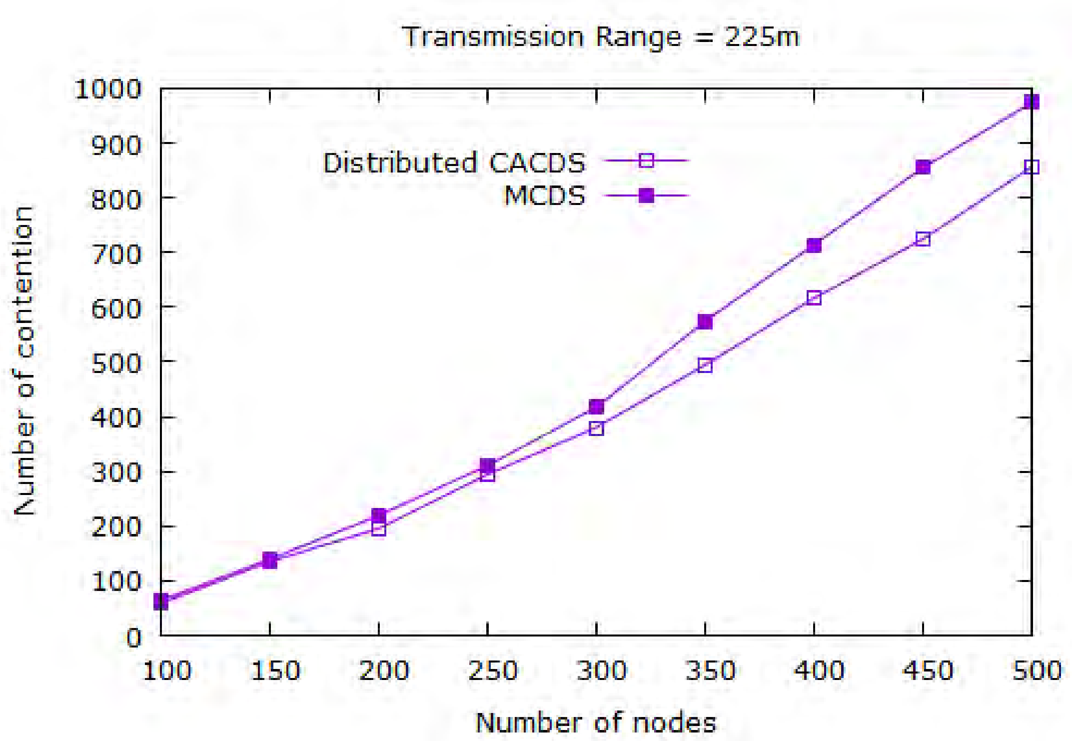



Figure 5.23: Performance comparison of DP and Distributed CACDS in term of number of contention occurs for transmission range 225m

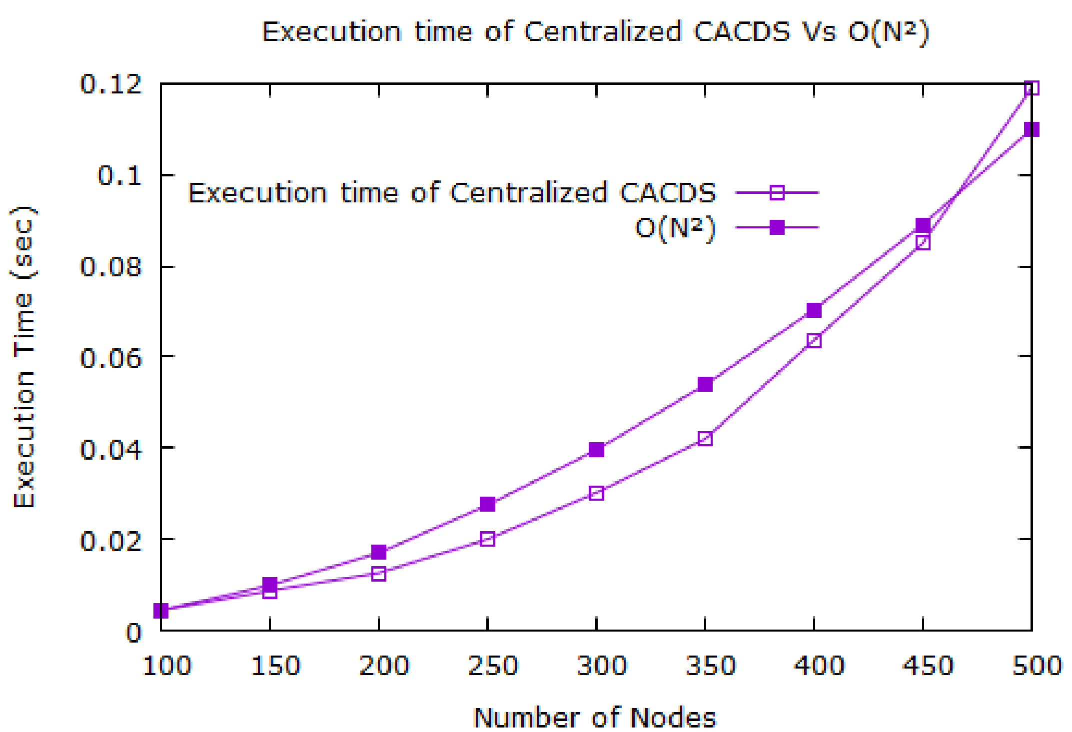



Figure 5.24: Execution time comparison of Centralized CACDS and $O(N^2)$

# Chapter 6

# Conclusion and Future Work

In this thesis, a new approach has been introduced to construct connected dominating set in order to minimize contention issue of broadcast storm problem. Both centralized and distributed version of the algorithm has been designed to analyze how they behave when they have entire topology information and how they behave when they have only limited 2-hop neighborhood information. As it is the first work done on contention aware CDS, so our centralized algorithm has been used as a bench mark to observe the effectiveness of our distributed one. 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 term of number of forwarding nodes and number of contention occurs among the nodes in CDS. We have done the simulation on 100-500 nodes in different transmission range. Our simulation result shows that, the centralized algorithm perform better than MCDS in order to reduce the contention and in most of the cases it constructs CDS without contention. The distributed algorithm also perform well than DP in this regard. Although the number of forwarding nodes increases a little bit while minimizing contention but this increment rate is not so high.

Though our centralized algorithm work better than the other centralized algorithm and the distributed one perform better than other distributed ones, but there still occurs a lot of contention in the distributed version compared to the centralized one. In future, we will work on minimizing this gap between the two versions to make our distributed algorithm more efficient. Because of contention, data sent from one node to another is delayed and it prolongs the execution speed of the network. As we are minimizing contention, so this execution speed of the network will also increase. In the future, we will also analyze the speed of execution of network as our performance metric to evaluate our proposed algorithms.

# Bibliography

[1] 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, no. 2, pp. 153–167, 2002.

[2] S. Butenko, X. Cheng, C. A. Oliveira, and P. M. Pardalos, "A new heuristic for the minimum connected dominating set problem on ad hoc wireless networks," *Recent developments in cooperative control and optimization*, vol. 3, pp. 61–73, 2004.

[3] J. P. Mohanty, C. Mandal, C. Reade, and A. Das, "Construction of minimum connected dominating set in wireless sensor networks using pseudo dominating set," *Ad Hoc Networks*, vol. 42, no. C, pp. 61–73, 2016.

[4] J. Yu, N. Wang, G. Wang, and D. Yu, "Connected dominating sets in wireless ad hoc and sensor networks–a comprehensive survey," *Computer Communications*, vol. 36, no. 2, pp. 121–134, 2013.

[5] G. Purohit and U. Sharma, "Constructing minimum connected dominating set: Algorithmic approach," *Applications of graph theory in wireless ad hoc networks and sensor networks*, vol. 2, no. 3, pp. 59–66, 2010.

[6] A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," *Proceedings of the IEEE*, vol. 75, no. 1, pp. 56–73, 1987.

[7] B. An and S. Papavassiliou, "A mobility-based clustering approach to support mobility management and multicast routing in mobile ad-hoc wireless networks," *International Journal of Network Management*, vol. 11, no. 6, pp. 387–395, 2001.

[8] R. Sivakumar, P. Sinha, and V. Bharghavan, “Cedar: a core-extraction distributed ad hoc routing algorithm,” *IEEE Journal on Selected Areas in communications*, vol. 17, no. 8, pp. 1454–1465, 1999.

[9] J. Wu and B. Wu, “A transmission range reduction scheme for power-aware broadcasting in ad hoc networks using connected dominating sets,” in *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, vol. 5, pp. 2906–2909, IEEE, 2003.

[10] J. Wu, B. Wu, and I. Stojmenovic, “Power-aware broadcasting and activity scheduling in ad hoc wireless networks using connected dominating sets,” *Wireless Communications and Mobile Computing*, vol. 3, no. 4, pp. 425–438, 2003.

[11] S. Datta, I. Stojmenovic, and J. Wu, “Internal node and shortcut based routing with guaranteed delivery in wireless networks,” *Cluster computing*, vol. 5, no. 2, pp. 169–178, 2002.

[12] B. Chen, “An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks,” *Proceedings of ACM/IEEE MOBICOM’01*, 2001.

[13] J. A. Shaikh, J. Solano, I. Stojmenovic, and J. Wu, “New metrics for dominating set based energy efficient activity scheduling in ad hoc networks,” in *Local Computer Networks, 2003. LCN’03. Proceedings. 28th Annual IEEE International Conference on*, pp. 726–735, IEEE, 2003.

[14] B. Deb, S. Bhatnagar, and B. Nath, “Multi-resolution state retrieval in sensor networks,” in *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pp. 19–29, IEEE, 2003.

[15] M. Ding, X. Cheng, and G. Xue, “Aggregation tree construction in sensor networks,” in *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, vol. 4, pp. 2168–2172, IEEE, 2003.

[16] U. C. Kozat and L. Tassiulas, “Service discovery in mobile ad hoc networks: an overall perspective on architectural choices and network layer support issues,” *Ad Hoc Networks*, vol. 2, no. 1, pp. 23–44, 2004.

[17] B. Das and V. Bharghavan, "Routing in ad-hoc networks using minimum connected dominating sets," in *Communications, 1997. ICC'97 Montreal, Towards the Knowledge Millennium. 1997 IEEE International Conference on*, vol. 1, pp. 376–380, IEEE, 1997.

[18] S. R. Bevan Das and V. Bharghavan, "Routing in ad-hoc networks using a virtual backbone," in *Proceedings of the 6th International Conference on Computer Communications and Networks (IC3N'97)*, pp. 1–20, 1997.

[19] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, no. 4, pp. 374–387, 1998.

[20] L. Ruan, H. Du, X. Jia, W. Wu, Y. Li, and K.-I. Ko, "A greedy approximation for minimum connected dominating sets," *Theoretical Computer Science*, vol. 329, no. 1-3, pp. 325–330, 2004.

[21] H. Lim and C. Kim, "Flooding in wireless ad hoc networks," *Computer Communications*, vol. 24, no. 3, pp. 353–363, 2001.

[22] W. Peng and X.-C. Lu, "On the reduction of broadcast redundancy in mobile ad hoc networks," in *Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*, pp. 129–130, IEEE Press, 2000.

[23] C. J. Colbourn and L. K. Stewart, "Permutation graphs: connected domination and steiner trees," *Discrete Mathematics*, vol. 86, no. 1-3, pp. 179–189, 1990.

[24] D. Lichtenstein, "Planar formulae and their uses," *SIAM journal on computing*, vol. 11, no. 2, pp. 329–343, 1982.

[25] X. Cheng, M. Ding, and D. Chen, "An approximation algorithm for connected dominating set in ad hoc networks," in *Proc. of International Workshop on Theoretical Aspects of Wireless Ad Hoc, Sensor, and Peer-to-Peer Networks (TAWN)*, vol. 2, 2004.

[26] M. Min, H. Du, X. Jia, C. X. Huang, S. C.-H. Huang, and W. Wu, "Improving construction for connected dominating set with steiner tree in wireless sensor networks," *Journal of Global Optimization*, vol. 35, no. 1, pp. 111–119, 2006.

[27] D. Chen, D.-Z. Du, X.-D. Hu, G.-H. Lin, L. Wang, and G. Xue, "Approximations for steiner trees with minimum number of steiner points," *Journal of Global Optimization*, vol. 18, no. 1, pp. 17–33, 2000.

[28] S. Butenko, C. Oliveira, and P. M. Pardalos, "A new algorithm for the minimum connected dominating set problem on ad hoc wireless networks," *Proceedings of CCCT'03*, pp. 39–44, 2003.

[29] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*, pp. 7–14, ACM, 1999.

[30] R. Rab, S. A. D. Sagar, N. Sakib, A. Haque, M. Islam, and A. Rahman, "Improved self-pruning for broadcasting in ad hoc wireless networks," *Wireless Sensor Network*, vol. 9, no. 02, p. 73, 2017.

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

[32] A. Rahman, M. E. Hoque, F. Rahman, S. K. Kundu, and P. Gburzynski, "Enhanced partial dominant pruning (EPDP) based broadcasting in ad hoc wireless networks.," *JNW*, vol. 4, no. 9, pp. 895–904, 2009.