

Relative Complement Set Based For Connected Dominating Set Algorithm For the Ad Hoc Network

Mao-Lun Chiang

Chaoyang University of Technology

Hui-Ching Hsieh (✉ luckyeva.hsieh@gmail.com)

Hsing Wu University <https://orcid.org/0000-0002-2666-3115>

Chia-Fen Hsieh

Chaoyang University of Technology

Wei-Ling Lin

Chaoyang University of Technology

Xin-Zhi Lin

Chaoyang University of Technology

Research

Keywords: Connected Dominating Set, Ad-Hoc Network, Routing Algorithm, MANET

Posted Date: October 25th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-993551/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Relative Complement Set Based for Connected Dominating Set Algorithm for the Ad Hoc Network

Mao-Lun Chiang¹, Hui-Ching Hsieh^{2*}, Chia-Fen Hsieh³, Wei-Ling Lin¹ and Xin-Zhi Lin³

¹Bachelor Degree Program of Artificial Intelligence, National Taichung University of Science and Technology, Taichung City, Taiwan ROC
{skyjustfly@gmail.com, will.wlink@gmail.com}

²Department of New Media Communication Hsing Wu University New Taipei City, Taiwan ROC
luckyeva.hsieh@gmail.com

*Corresponding Author

³Department of Information and Communication Engineering, Chaoyang University of Technology, Taiwan, R.O.C.
{cfhsieh@cyut.edu.tw, s10430610@gm.cyut.edu.tw}

Abstract

As mobile devices flourish, users can conveniently use the network services through the mobility of mobile devices. The most popular environment is the Mobile Ad Hoc Networks (MANETs), which consists of wireless hosts that communicate with each other in the absence of a fixed infrastructure. Data transmission for MANET is mainly achieved through peer-to-peer communication and forwarding technology without restrictions on the direction or extent of movement. This type of flexible architecture is suitable for emergency relief and military applications. As a result, a set of network nodes, called connected dominating set (CDS), is thus proposed as the virtual backbone of MANET. It mainly forms a CDS through a group of gateway nodes, and through this CDS for message delivery and routing management to reduce overall power consumption. Therefore, in the past, many scholars have proposed improved CDS construction algorithms to achieve a minimum number of CDS.

In this paper, the Improved Relative Complement Suffrage CDS (IRCS-CDS) algorithm is proposed to effectively find out the stable and minimum CDS. Besides, the IRCS-CDS can adapt to the changeability of MANET even in a more loosely distributed network environment. The complexity of the proposed IRCS-CDS algorithm is $O(2n^2 + 3n)$, which is lower than other algorithms. Therefore, the proposed IRCS-CDS algorithm is more stable and efficient even in a network environments with loose nodes or small node coverage radius.

Keywords: Connected Dominating Set, Ad-Hoc Network, Routing Algorithm, MANET.

1 Introduction

With the continuous development and vigorous development of mobile network technology, users can now easily obtain services through the mobility of mobile devices and the convenience of the Internet.

In general, the wireless network architecture can be divided into client/server wireless network (Infrastructure) and Ad-Hoc wireless network (Non-Infrastructure) [1][2]. The client/server wireless network requires a base station infrastructure to assist in transmission of data, such as Access Point (AP). However, when the base station fails, the network range in which it covers will be paralyzed. In contrast, the Ad-Hoc wireless network is a peer-to-peer network [3][4], and the data transmission method under such an architecture is by applying the forwarding technique between nodes. Besides, under the Ad-Hoc wireless network topology, the movement direction and the transmission range of nodes can be changed at any time. Thus, this network structure is more suitable to be deployed for disaster relief or military systems. However, there still has some problems in this network environment, such as limited network bandwidth, memory capacity, power consumption, packet collisions, redundant message delivery, changeability of network topology and so on [5]. Therefore, how to solve these problems to improve the overall performance and stability becomes a very important issue.

Since the nodes in Ad Hoc Network have the characteristics of mobility, the network topology status and routing information will be changed accordingly. Besides, the transmission capacities of the nodes can also affect the routing path. For example, when the transmission range of nodes is getting larger in the network, the network topology will also be denser. In such a circumstance, the candidate of the routing path for transmitting data will become more flexible. Also, the bandwidth resources of the network and battery capacity of mobile devices are limited. Hence, when the bandwidth or the power is exhausted, the node will no longer be able to transmit data, and the entire network topology is destroyed.

To solve the above problems, some hierarchical [6][7][8][9][10][11] routing protocols have been proposed under the Ad-Hoc wireless network environment. The primary concept is that some specific nodes play the role of gateways to create a virtual backbone for message communication and routing [12][13]. The Dominating-Set-based (DS) routing protocol proposed by Wu et al [14] is a hierarchical routing protocol that can effectively broadcast the message through the transmission paths which are based on the DS structure and can ensure that all nodes in the network can send or receive packets correctly. Besides, when the number of gateways in the DS is reducing, the number of nodes that are required to store the routing path table will also be reduced. This scheme can help not only to avoid redundant transmitted messages but also avoid the situations of packet collisions, broadcast storms, electricity consumption, or other problems.

Therefore, the goal of the traditional DS (Dominating Set) [5][15][16][17] algorithms are to find

the minimum number of gateways to construct a smaller CDS (Connected Dominating Set) backbone. The main concept of the algorithm is using the status of branching degrees, node *id*, power, or the number of the node that are not covered by the CDS to decide which nodes will play the role of the gateway. Unfortunately, the system still cannot get better performance due to different factors, such as the deployment of node, node *id* or energy ability. These circumstances cause the overall performance of the network to decrease.

To solve the mentioned problem, the new Improved Relative Complement Selection Connected Dominating Set (IRCS-CDS) algorithm has been proposed in this article. By the experiment results, we can prove that the IRCS-CDS algorithm can effectively reduce the number of gateways to get better performance and utilization for the overall network.

The rest of this paper is organized as follows: Section 2 gives the related works of the basic wireless network architecture and the CDS algorithms proposed by other scholars. Section 3 describes the details of the proposed IRCS-CDS algorithm. To help to understand the proposed IRCS-CDS algorithm, an example is proposed in Section 4. Section 5 presents the results of the simulation of the algorithm. Finally, Section 6 gives the conclusions.

2 Related Works

In this section, the basic architecture of wireless Ad-Hoc networks is described first. The related route-related algorithms will be discussed in section 2.2. Finally, the various methods of the Connected Dominating Set (CDS) will be introduced and discussed in depth in section 2.3.

2.1 Introduction of Wireless Ad-Hoc Network

Compared to the architecture which applies the concepts of link state [18] or distance vector [19] to maintain and update the network status of traditional wired networks, Wireless Ad-Hoc Network (WANET) architecture is more flexible. Basically, the WANET architecture [20][21] is a peer-to-peer autonomous network architecture that does not require to plan and distribute the point in advance and can be automatically organized and constructed by passing information about themselves between nodes. Besides, each node in the Ad-Hoc network plays the role of host and router. When nodes request for data transfer, the source nodes will take the multi-hop transport mode to transmit the data from the adjacent nodes to the destination node. The overall transmission will be completed through several relay nodes. Base on the mentioned concepts, the Mobile Ad-Hoc Network (MANET) architecture has been proposed.

In this MANET architecture shown in Figure 1, and the system can maintain a certain level of service even when all nodes are moving in the network arbitrarily [5][16][22] under the multi-hop mode. So far, some routing protocols which transmit data by applying the concept of multi-hop have been proposed, and the flooding [5][23][24] method is one of the basic transmission protocols. Unfortunately, this method can easily generate the problems of redundant packet transmission, broadcast storms. It will not only reduce the overall operation efficiency and the life cycle but also increase the power consumption of the nodes for the network. To sum up, using the flooding method wastes resources and is not efficient for the network. To solve these problems, the hierarchical routing method have been widely used, and the main character is to construct the network with a Virtual Backbone [25] architecture. Here, all data will be sent through this virtual backbone, hence, it can avoid extra data be sent and reduce the problem of power consumption. More details of the routing protocols for wireless networks will soon be described in the next section.

Figure1 Here

2.2 Routing Protocols for MANET

The routing protocols in MANET environments can be classified into three modes: proactive mode [23], responsive mode [23], and hierarchical mode [14]. For the proactive routing mode, it will create a path in advance, and each node must send the routing information at regular intervals to update the routing table. The Wireless Routing Protocol [26] and Destination Sequenced Distance Vector (DSDV) protocol [27] are these kinds of proactive routing protocols. Besides, the main advantage of this routing protocol is that once the packet has been sent out, the system can get the information of the routing path from the sender to the destination immediately. However, the method of sending the routing information at regular intervals to update the routing table is a waste of the bandwidth and power for the nodes in the network. To overcome this problem, the system must extend the interval to broadcast the routing information. Unfortunately, this solution will cause the routing table not to be updated on time and generate the problem of package loss. Hence, it leaves many problems to be overcome.

For the responsive routing modes, the system will launch the route discovery procedure to create the routing information for the routing table when the system cannot find out the route to the destination for transmitting the package. The popular responsive routing modes including Dynamic Source Routing (DSR) [28] and Ad-Hoc On-Demand Distance Vector Routing (AODV) [29]. Although the responsive routing mode can efficiently save the use of network bandwidth, it still has the longest average latency

problem in transferring packages.

In the last mode, the main idea for the hierarchical routing mode [39] is to divide the entire network into several hierarchical network architectures, and each new architecture has its routing scheme. This mode can help to solve the routing problem under large-scale networks. Besides, in the Internet architecture, wireless networks can typically be divided into three tiers: the backbone, the middle tier and the subnet. Here, the backbone network is responsible for packet routing for the middle tier network, and the middle tier network is responsible for packet routing for the subnet. The internal network routing scheme for the subnet will be proceeded by itself, and this is the basic cooperating method for the overall hierarchical Internet architecture. Since the hierarchical routing mode can respond to the changes in the network variabilities efficiently, it becomes a widely used network architecture so far [30]. Dominating-Set-Based Routing is one of the most representative algorithms based on the hierarchical routing mode.

The basic concept of the dominating set applies the graph theory in which the network can be seen as a graph which is defined as $G(V, E)$ (V represents the nodes in the wireless network, E represents a collection of all edges in the wireless network). The dominating set is a subset dominating set (DS) of V such that every vertex not in DS is adjacent to at least one member of DS. Furthermore, the nodes which are in the DS are called gateway nodes which will construct a backbone network to transmit information. On the contrary, the remaining nodes, which are not in the DS, are called non-gateway. Basically, the package sent from these non-gateway nodes will be transmitted rely on the nodes in the DS. The main advantage of using this dominating set to build MANET is that the routing information will only be stored by the gateway nodes, and this can help to reduce the resource consumption of nodes and to manage the network more easily.

So far, many scholars have proposed some related dominating set based routing algorithms [14] [15][5][16][22][31] to improve the performance of the system. For example, an efficient dominating set based routing algorithm [14] is proposed by Wu. et al., the main idea of the proposed routing method can be divided into the following three steps:

- 1). When the source node (Source) who is not the gateway node sends the packet out, the packet will be sent to the adjacent gateway node. At this time, the node that receives this packet will become the source gateway.
- 2). The source gateway will be seen as a new source and it will then find out the routing path for transmitting the packet through the constructed CDS members. The package will be sent to the

next gateway node based on the routing information continuously.

- 3). The packet will finally be sent to the destination gateway. If the destination gateway is not the final receiver, its adjacent node will be. Since the package will be sent to the receiver by the destination gateway, and the overall transmission procedure is completed.

Here is an example to help to understand the idea, and the Ad-Hoc network topology is shown in Figure 2. Here, each gateway (CDS Member) retains its non-gateway members (Covered node) and the routing table. Furthermore, in this topology, nodes 2, 3, 4, 5, 7 and 15 are the gateway nodes that will construct a CDS topology. Take node 2 for example, its dominant member includes nodes 19 and 20, and the related routing table for node 2 is shown in **Error! Reference source not found.**. In the routing table, it contains information about the gateways (nodes 3, 4, 5, 7, and 15), each gateway's covered nodes and the distances to reach the destination (hops) number. Assuming node 19 wants to send packets to node 16, the package will be passed to node 2 (become a source gateway) first, and then to node 4. Continuously, Node 4 will send the package to node 15. At this time, node 15 will become a destination gateway since node 16 is its an adjunct node and is the destination. Finally, the packet is sent to destination node 16 through node 15.

Figure 2 Here

Table 1 Here

2.3 Related Algorithms of Constructing Connected Dominating Set

So far, some algorithms for constructing CDS have been proposed, and the first related algorithm called Wu's CDS algorithm was proposed by Wu and Li [14]. The main idea of the proposed algorithm is to create CDS nodes as the virtual backbone to reduce the consumption of messaging across the network. However, this algorithm will also generate a larger number of CDS members. Therefore, the ID-based CDS (ID-CDS) [22] algorithm has been proposed to improve this problem. This algorithm will select the proper CDS nodes based on each node's ID. However, its algorithm cannot significantly reduce the number of CDS when the nodes are not distributed well in the network. Following, the MaxD-CDS algorithms [5][15][16][22][31] have been proposed to solve the problem mentioned. The main idea of the MaxD-CDS algorithm is to consider whether the nodes are effectively covered by the CDS members. Unfortunately, the performance is not significant under the loose network environment. Hence, the Reducing Connected Dominating Set (RCDS) algorithm [32] which will reassign the ID and execute two remark procedures to reduce the number of CDS has been proposed. However, the complexity of this

RCDS algorithm will be greatly increased and resulting in a decrease in overall performance. Besides, the MaxS-CDS algorithm [33] is proposed to use the Suffrage Value (SV) to reduce more CDS members. Its main idea is to calculate the node's SV by using the degree of the node, the coverage status of the CDS and id value. This calculated SV is the key judgment for selecting the CDS members and can help to reduce the number of CDS. Still, this algorithm cannot get a smaller CDS backbone when the nodes are unevenly distributed in the network. To help to understand the key concepts of the CDS algorithm, some details will then be described as follow.

The purpose of Wu's CDS algorithm [14] is to construct the CDS structure as the virtual backbone network for transmitting packages. It is to treat the entire MANET as a graph $G = (V, E)$, where V represents a collection of all nodes and E represents the collection of all edges. Besides, there are two processes in Wu's CDS algorithm: the marking process and the re-marking process. Meanwhile, each node $v(v \in V)$ is marked with $m(v)$ action, and then classified as T (gateway) or F (non-gateway). The details of these two processes are described below:

- The marking process

- 1). Mark each node in the connected MANET as F .
- 2). Each node exchanges the messages about its neighbors with all 1-hop nearest neighbor nodes ($N(v)$).
- 3). Each node will determine whether its neighbor nodes are covered by its neighboring nodes or not with 1-hop nearest neighbor nodes. If the result is none, mark itself as T. After finished the marking process is completed, all nodes which are mark if T will be in the set V' where $V' = \{v | v \in V, m(v) = T\}$. The nodes in V' will then construct a new subgraph G' , where $G[V']$.

At this time, the number of CDS members that are marked through the marking process is still quite large. To effectively reduce the number of CDS members, the system will perform a re-marking process. In the re-marking process, the system can remove redundant gateway nodes by considering the cover status of the message of the neighbors, and the rules for the re-marking process are shown as follows:

- The re-marking process

There are two rules in the re-marking process:

- Rule1. Consider two nodes v and u in G' , and if $N[v] \subseteq N[u]$ and $id(v) < id(u)$, the mark for v is changed to F if node v is marked, and G' will also be changed to

$$G' - \{v\}.$$

Rule2. Suppose u and w are two adjacent marked nodes of marked v in G' . If $N(v) \subseteq N(u) \cup N(w)$ is presented in G and $id(v) = \min\{id(v), id(u), id(w)\}$, the mark for v is changed to F .

To sum up, Wu's CDS algorithm is first to find out the CDS member for MANET through a marking process, and then reduce the number of CDS by using id , degree, and coverage status between nodes in the re-marking process. The purpose of the re-marking process is to improve the problem that the number of CDS nodes is too large at the initial marking process.

Later, some algorithms which are based on Wu's CDS algorithm have been proposed to construct a smaller CDS [14][33][34][35][36][37]. One of them is the ID-CDS [22] algorithm proposed by Velumylyum et al. In ID-CDS algorithms, nodes are classified into two groups: CDS-Node-List and Covered-Node-List, where CDS-Node-List represents all CDS node members in the network and Covered-Node-List represents all nodes that are adjacent to the CDS node. The main concept of the ID-CDS algorithm is to select the node as the CDS member when the node has the largest id in MANET, and the detailed steps are listed as follows:

- 1). Find the node with the largest id in the overall network as a starting point to select the CDS node.
- 2). The nodes selected as CDS members are added to the CDS-Node-List and Covered-Node-List, and their neighbor nodes are added to the Covered-Node-List.
- 3). Select the node with the largest id from the Covered-Node-List. For the node, if there has one more neighbor that is not in the Covered-Node-List group, this node will then become the next node to join the CDS.

Here, steps 2 and 3 of the algorithm will then be executed repeatedly until all nodes in the network are added to the Covered-Node-List. Basically, the key point of the ID-CDS algorithms is to determine the CDS by the id values of the nodes while executing. Furthermore, the id values of the nodes are not repeatable, hence the system will not pick up two nodes as the CDS at the same time. ID-CDS algorithms may reduce the number of CDS, however, the system may sacrifice the nodes whose coverage abilities are better.

Similar to the ID-CDS algorithm, Meghanathan and other scholars have proposed the MaxD-CDS [5][15][16][22][31] algorithms to find the CDS nodes. The main idea of the MaxD-CDS algorithm is to calculate the number of uncovered neighbors for each node based on the node's neighbor message and

the coverage relationship, and then select the nodes with the largest number of uncovered neighbors to be the CDS members. This algorithm can help to construct a smaller number of CDS network more quickly, and the executing steps are listed as follows:

- 1). Find out the node that has the largest number of uncovered neighbors in the network as the starting point for the algorithm.
- 2). The nodes selected as CDS members are added to the CDS-Node-List and Covered-Node-List, and their neighbor nodes are added to the Covered-Node-List.
- 3). Select the nodes with the largest number of uncovered neighbors from Covered-Node-List as the next to join the CDS.

Also, steps 2 and 3 of the MaxD-CDS algorithm will then be executed repeatedly until all nodes in the network are added to the Covered-Node-List, and this means that all nodes are covered by the CDS nodes. Here, the MaxD-CDS algorithm mainly considers the number of uncovered neighbors of each node while selecting CDS nodes. Hence, the node with greater coverage ability can be effectively selected as the next CDS node. In a practical situation, once there are more than one nodes that have the same number of uncovered neighbors, the system may select a non-proper node to be the CDS node.

Another CDS algorithm called the RCDS algorithm [32] has been proposed by Jea et al. In the RCDS algorithm, there are two processes: the reassignment process and the advance process to find the smallest CDS nodes. The reassignment process calculates the new *id* for the node by using the parameters includes the maximum *id* value of 2-hop of the neighbor node, degree and *id* value, and new *id* will be marked as *id'*. With the *id'* value, the greater the chances of becoming a CDS member. After that, the advance process which concept is similar to the k-rules algorithm proposed by Wu Scholar [40] will be executed to reduce the number of CDS. The key difference from k-rules is that when creating the subgraph, the advance process will exclude the criteria for comparing the *id* value of nodes. It's simply considered the coverage relationship of the nodes to obtain a smaller CDS. Basically, the number of CDS is smaller than the algorithm proposed by Wu and Li. Furthermore, there are three main steps in the RCDS algorithm, and the details are listed below:

- 1). Execute the marking process proposed by Wu et al.
- 2). Execute the reassignment process to reduce the number of gateways for CDS.
- 3). Execute the advance process to obtain a smaller CDS.

The RCDS algorithm uses local (2-hop) information to reduce the number of CDS. When the node

has a relatively larger transmission radius, the reduction is more significant. Therefore, the RCDS algorithms can reduce not only the number of CDS but also the amount of information about routing and bandwidth consumption. The only fly in the ointment is that the RCDS algorithm requires two remark processes to reduce the number of CDS, and that will increase the complexity of the algorithm.

Another CDS algorithm called the MaxS-CDS algorithm [33] was proposed to improve the power consumption problem while the number of CDS is getting larger. Here, the MaxS-CDS algorithm contains two main processes: the sufferage CDS selection process and the remark process. In the sufferage CDS selection process, the Sufferage Value (SV) value [38] is calculated by using the parameters such as *id*, degree, and the number of uncovered neighbors of the node in MANET. With the SV value of a node, the greater the chances of becoming a CDS member. Then, the remark process is executed to reduce the number of CDS again. Besides, the concept of the remark process in the MaxS-CDS algorithm is similar to the concept in the advance process [32] proposed by Chen. The key difference from the advance process is that when creating the subgraph, the remark process will exclude the criteria for comparing the *id* value of nodes. It's also simply considered the coverage relationship of the nodes to obtain a smaller CDS. The detailed concepts of MaxS-CDS algorithm are shown as follow:

In the sufferage CDS selection process of the MaxS-CDS algorithm, nodes will be divided into three primary groups:

- CDS Gateway List (CGL): The collection of CDS members in MANET.
- Covered-Node-List (CNL): The node which is already a CDS node or is covered by at least one CDS node.
- Priority List (PL): The node which will be in the CNL and has at least one uncovered neighbor node.

Subsequently, the main steps of the sufferage CDS selection process are shown as follows:

- 1). Find out the nodes that have the largest number of degrees in the MANET as the initially selected CDS, and then add these nodes to CGL and CNL.
- 2). The neighbor node of the CGL is then added to the CNL. In the CNL, the nodes that do not include in the CGL and have an uncovered node will be included in the PL based on their degrees. After that, the node which is in the PL and has the largest degree will be preferred to be added into the CGL. Noticeably, if the node is already in the CGL, its degree information will not be stored in the PL.

3). While selecting the next node to be added into the PL, the system will calculate the SV value once there has one more node that has the same degree. In the algorithm, the node which has the larger SV value will be selected.

In the sufferage CDS selection process, steps 2 and 3 will then be executed repeatedly until all nodes in the network are added to the CNL. Basically, the MaxS-CDS algorithm can generate fewer CDS members in a loose network environment than other algorithms and can form a minimal CDS virtual backbone topology efficiently. However, the MaxS-CDS algorithm may still not select the optimal nodes as the CDS member when the nodes are unevenly distributed in the network, and the details are described below.

In general, when there have some nodes with the same degree in the network, the MaxS-CDS algorithm will calculate the SV value of each node based on the formula (1) to determine whether they will become the CDS member or not, and the example of the process is shown in Figure 3. In the example, there are three numbers above each node, and the meanings of these three numbers are the degree, the number of the uncovered node and the SV value sequentially.

$$SV = \frac{\text{node degree} * \text{number of uncovered node}}{\text{node id}} \quad (1)$$

Figure 3 Here

In this topology, the SV value requires to be calculated since node 1 and node 2 have the same maximum degree. Once the SV values calculated are the same, the system will randomly select one node among node 1 and node 2, and then adds it to the CDS group. Now, if node 1 has been selected as the CDS member, the network will generate more CDS members than the result of selecting node 2. Hence, the MaxS-CDS algorithm is unstable and the smallest CDS still cannot be found.

To solve the mentioned problems, an Improved Relative Complement Sufferage CDS (IRCS-CDS) algorithm is proposed to find the minimum number of CDS members and help the network stay stable. The details will be described as following.

3 Methods

To reduce the number of CDS nodes effectively, the Improved Relative Complement Sufferage CDS (IRCS-CDS) algorithm has been proposed in this paper. The followings are the detailed descriptions of the IRCS-CDS algorithm.

3.1 Definitions of Noations

The definitions of the noations in this paper are listed below:

- 1). $G = (V, E)$: The network will be seen as a simple graphic $G = (V, E)$, where V represents a collection of all nodes and E represents the collection of all edges.
- 2). (u, v) : There are two nodes (u and v) connected in the network, and both of them are within the transmission radius of each other.
- 3). V' : The set of the nodes that are marked as CDS nodes, where $V' = \{v | v \in V, m(v) = T\}$.
- 4). G' : The subgraph form with V' based on G .
- 5). $N(u) = N_1(u) = \{v | (u, v) \in E\}$: The collection of the neighbor nodes of node u within 1-hop, and $N(u)$ is known as the open neighbor set of node u . Besides, $N_2(u)$ represents the collection of the neighbor nodes of node u within 2-hop.
- 6). $N[u] = N_1[u] = N(u) \cup \{u\}$: The collection of the neighbor nodes of node u within 1-hop and node u itself, and $N[u]$ is known as the close neighbor set of node u . Besides, $N_2[u]$ represents the collection of the neighbor nodes of node u within 2-hop and node u itself.
- 7). RC_u : The Relative Complement (RC) value that is calculated by node u .
- 8). SV'_u : The sufferage value of node u . It represents the weight value calculated by node u .
- 9). Message Exchange: The information, including the *id*, degree, power of the node and the *id* information of the neighbors, that are required to be exchanged among the nodes.
- 10). CML (CDS Member List): A list of all CDS members in the network.
- 11). CNL (Covered Node List): A list of all nodes which are covered by at least one CDS member or the nodes which are the CDS member themselves in the network.
- 12). uncovered_node: The nodes that are not yet covered by CDS members in the network.
- 13). PML (Priority Member List): A list of the node that is the CNL member and has at least one neighbor node which is the uncovered_node. Besides, The list is sorted according to the degrees of the nodes in descending order.

3.2 The procedure of IRCS-CDS Algorithm

Here, the IRCS-CDS algorithm will pick out the nodes which have higher weight values, and then add them to become the CDS member. After that, the system will utilize the cross-coverage relationships

between nodes to reduce the number of CDS members. This procedure can help to decrease the impact caused by the poor locations of the nodes. Besides, the proposed IRCS-CDS algorithm is divided into two main procedures, namely the Relative Complement CDS Selection procedure and the Gateway Remark procedure, the following will be described in detail.

For the Relative Complement CDS Selection program, all nodes will periodically connect to their 1-hop neighbors and exchange message with them in the MANET environment. Meanwhile, the system will execute the Relative Complement CDS Selection procedure, and the detailed steps are shown as follows:

- 1). Calculate the degree of each node, and then select the node which has the largest degree to join the CNL and the CML. These selected nodes will be treated as the initial CDS members in the MANET network.
- 2). Add the neighbors of the CML member into the CNL. In the CNL, the nodes which are not in the CML and have 1-hop uncovered neighbors that are not added to the CNL are picked out to PML. Here, the PML will be updated when the CML or the CNL is updated.
- 3). Select the node which has the largest degree in the PML as the next CDS member. If several nodes have the same degree value, the system will calculate the RC_u value of each node u as a parameter to calculate the SV' value. This calculated SV' value will become the weight to decide which node will become the CDS member. Here, the node which has the higher SV' value will become the CDS member. The related formulas are shown in formula (2) and formula (3). The Relative Complement CDS Selection process will be executed until all nodes in the MANET environment are added to the CNL.
- 4). The nodes in the CML will be the CDS members, and these CDS members will form a new subgraph denoted as G' .

$$RC_v = uN[v] - same_{uN[v]} \dots \dots \dots (2)$$

$uN[v]$: The number of uncovered neighbor nodes of node v .

$same_{uN[v]}$: The $uN[v]$ nodes which are also covered by the neighbors of node v .

$$SV' = \frac{node\ degree * RC}{node\ id} \dots \dots \dots (3)$$

Here, the higher the RC value means that this node has more uncovered neighbors than other nodes, so it will be selected as CDS members to reduce the number of CDS members. Subsequently, the SV' value is calculated by above two formulas to find out the most suitable CDS members. Figure 4 shows the algorithm for the Relative Complement CDS Selection procedure.

Figure 4 Here

Subsequently, in the Gateway Remark procedure, the proposed protocol does not remark the redundant gateway node by using the node id . On contrary, the algorithm applies the strong connection pass method [33] to ensure the connectivity of CDS, and the concept of strong connectivity is defined as follows:

Assume that the entire network is treated as a graph $G = (V, E)$. Here, V' represents a collection of the gateway (T) marked with $m(v)$ in V where $V' = \{v | v \in V, m(v) = T\}$ and V'_+ represents a strongly connected component in V' . $V'_+ = \{w | w \in V' \cap N(v)\}$. The Gateway Remark procedure is shown as follows:

- 1). For each node $v \in V'$, create the sub-graph $G[V'_+]$ where $V'_+ = \{w | w \in V' \cap N(v)\}$.
- 2). If there is a set of strongly connected nodes V'_{ci} ($1 \leq i \leq l$) and $N(v) \subseteq N(V'_{ci})$, the node v will be marked ($m(v)$) as the non-gateway (F).
- 3). Output the remark result and update the CDS member as a new G' .

After the Remark procedure, the new G' will also be the connected set. Subsequently, the IRCS-CDS algorithm can obtain a smaller number of CDS members. To help to understand the proposed IRCS-CDS algorithm, an example will be given in the next section.

4 Example of the IRCS-CDS Algorithm

In this example, we assume that there are 20 nodes with the same transport radius in the network. When the distance between two nodes is less than the transport radius, these two nodes can be connected directly. The overall procedure is described below.

4.1 The IRCS-CDS Selection procedure

In the IRCS-CDS algorithm, the system will execute the Relative Complement CDS Selection procedure to select the CDS members. As shown in Figure 5, the number in each circle represents the

node *id* of each node. There are four numbers (denoted as A, B, C, D for description) defined in the parentheses around the node where A represents the degree of the node, B is the number of uncovered neighbors of the node, C is the calculated *RC* value of the node (The initial value is 0) and D is the *SV'* value of the node (The initial value is 0).

Figure 5 Here

At first, the Relative Complement CDS Selection procedure will select the node with the largest degree as the initial node. In this example, node 4 is selected as the starting CDS node and will be added to the CML list since it has the largest degree in the network. At the same time, nodes 1, 2, 3, 6, 13, 15 and 17 will be added into the PML and nodes 1, 2, 3, 4, 6, 13, 14, 15, 17 and 18 will be added into CNL according to the second step of the Relative Complement CDS Selection procedure. The detailed results are shown in Figure 6.

Figure 6 Here

In the second round of the selection procedure, the system will select the node that has the largest degree in the priority list to join the CDS member. Therefore, node 3 will be added to the CML. So far, the CDS member includes nodes 3 and 4, and nodes 1, 2, 5, 7, 13, 15 and 17 are in the PML. For the CNL, it includes nodes 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 14, 15, 17 and 18. The results are shown in Figure 7.

Figure 7 Here

Figure 8 shows the results of the third round of the selection procedure. Here, node 13 has the largest degree, and it will be added to the CML. At the end of this round, the CML includes nodes 3, 4 and 13. Nodes in the PML includes node 1, 2, 5, 7, 15 and 17. Node 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 14, 15, 17, 18 and 21 are in the CNL.

Figure 8 Here

In the fourth round of the selection procedure, since node 5 and node 7 have the same degree, the system must calculate the *RC* and *SV'* values for these two nodes. In this example, the *RC* value is 2 (2-0=2) and *SV'* value is $2(\frac{5*2}{5} = 2)$ for node 5. Meanwhile, the *RC* value is 2 (2-0=2) and *SV'* value is $0.7(\frac{5*2}{7} = 0.7)$ for node 7. The calculated results are shown in Figure 9. Based on the results, the system will select node 5 as the CDS member since it has a larger *SV'* value. At the end of this round, the CML includes nodes 3, 4, 5 and 13. Nodes in the PML are nodes 1, 2, 7, 15 and 17. The CNL includes node 1,

2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 18 and 21. The related result is shown in Figure 10.

Figure 9 Here

Figure 10 Here

The selection will be executed until all nodes in the MANET environment are added to the CNL. Here, the results of each round of selection procedure are shown in Table 2, and the remaining results for the following rounds are shown in Figure 11 to Figure 14.

Table 2 Here

Figure 11 ~Figure 14 Here

After finishing the Relative Complement CDS Selection procedure, the CDS member can be effectively calculated and a smaller number of CDS members are selected. Following this, the system will execute the Gateway Remarking procedure.

4.2 The Gateway Remarking Procedure

As shown in Figure 15, we can find out that nodes 3, 4, 7, 8, 9 and 21 are neighbors of node 13, and these neighbor nodes can be covered by the CDS members including nodes 3 and 4 and 7. This situation meets the $N(v) \subseteq N(V_{ci}')$ condition, and the related result is shown in Figure 15. In this case, node 13 will be remarked as a non-gateway node. Finally, CDS members will be updated, and the new members include nodes 2, 3, 4, 5, 7 and 15. The PML is empty, and the nodes in the CNL are 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 and 23. The result is shown in Figure 16. Based on the two procedures in the IRCS-CDS algorithm, the system can get the smallest number of CDS members.

Figure 15 ~ Figure 16 Here

5 Results and Discussion

In this section, the related simulation results will be given. Also, the comparisons among the proposed IRCS-CDS algorithm and other famous algorithms including MaxD-CDS, RCDS and MaxS-CDS will be described and analyzed. The results are shown as follows.

5.1 Simulation Environment

In this research, we use the javascript programming language under the windows environment to compose the simulation environment. The related results and graphs will be recorded by Excel. Besides,

there are two main scenarios for simulation. First, the nodes of the simulation environment will be randomly scattered within 100*100 square meters, and the initial number of nodes is set to 50, with 50 nodes as the base is increased until there are 250 analog nodes in the region. All nodes have the same transmission range (radius R meters) ($R \in \{20,40,60\}$), and if the distance between the two nodes is less than R, the two nodes are connected directly. However, since the nodes are scattered randomly, this experiment will need to check whether all nodes in the network are connected. If the nodes in this graph cannot be connected, the system must reconfigure the node distribution until a connected graph is formed, and then the CDS algorithm can be executed. The simulation will be executed 50 times on the same number of nodes and transmission range, and then take their average as the result of the experiment.

Second, this research uses the degree g of nodes designed by previous scholars [31] to construct a loose and dense network environment. The range is set to (1~6) meters and the degree range is (7~16) and (17~30) meters in dense environments. This scenario will simulate the effect of node density in the current environment on the algorithm and identify algorithms that are more suitable for the specific environment.

Also, because the IRCS-CDS, RCDS and MaxS-CDS algorithms have Marking and Remark programs, we have documented these three algorithms separately. The number of CDS nodes generated after the Marking process is compared. This is mainly because if you can find a smaller number of CDS nodes during the Marking process, you can effectively reduce the burden of doing Remark later.

5.2 Simulation Results and Analysis

The first is the result of a simulated comparison of the number of CDS members resulting from variations in the number of nodes between 50 and 250 in transmission radius 20, 40, and 60. As shown in Figures 17, 18 and 19, IRCS-CDS and MaxS-CDS algorithms perform the smallest number of CDS. Since the program conditions for executing remark procedures are the same, the results among the proposed IRCS-CDS algorithm and the MaxS-CDS algorithm are equally the same.

Figure 17 ~ Figure 19 Here

In Figure 20, Figures 21 and 22, we compare the IRCS-CDS, MaxS-CDS and RCDS algorithms that executing the marking procedure in the same simulated environment. The number of CDS members is compared after the procedure. It can be found that the IRCS-CDS algorithm produces the smallest number of CDS when the marking procedures are completed. This can also mean that the proposed IRCS-

CDS algorithm is efficient and stable.

The next experiment will then simulate the number of CDS members in a fixed number of nodes among 50, 150, and 250 environments with a transmission range of 20 to 60. As shown in Figures 23, 24 and 25, the proposed IRCS-CDS algorithm also performs the smallest number of CDS across different transmission ranges.

Figure 20 ~ Figure 25 Here

The last experiment mainly simulates the impact of the network's tightness on each algorithm. First, in a looser network environment, we fixed the branch scale of each node (1~6), and the number of nodes varies from 50 to 250 to compare the number of CDS members. As shown in Figure 26, the MaxS-CDS algorithm and the proposed IRCS-CDS algorithm can build fewer CDS in a looser network environment. In the dense network environments, degree ranges are set to (7~16) and (17~30) as shown in Figures 27 and 28, we can find that the MaxD-CDS algorithm builds more numbers of CDS, because the concept of the MaxD-CDS algorithm is to select the number of nodes that are not covered. This factor promotes the nodes selected by the MaxD-CDS algorithm in a dense network environment to have a higher coverage rate, so that a smaller number of CDS members can be obtained.

However, CDS members that cover more non-gateway members will easily generate uneven workloads to reduce the lifetime of network. It is also not suitable for mobile arbitrary network architecture.

Also, under this network environment conditions, the number of CDS members after performing the Marking process will be compared among the IRCS-CDS, MaxS-CDS and RCDS algorithms. As shown in Figures 29, 30, and 31, those can be found that the IRCS-CDS and MaxS-CDS algorithms produce the minimum number of CDS when the Marking procedure is completed.

Figure 26~Figure 31 Here

5.3 The Conclusion of the Experiment

Based on the above simulation results, we can know that the proposed IRCS-CDS algorithm can construct a smaller number of CDS members after performing the marking procedure. The nodes in the remark procedure are then used to remove redundant CDS members to find the smaller set of CDS. At the same time, the proposed IRCS-CDS algorithm can achieve better results than other algorithms when the number of nodes is set to 50~250 and the transmission distance is set to 20~60. In the dense network

environments (degree 7~ 16), the MaxD-CDS algorithm can obtain a lower number of CDS members, that is because the MaxD-CDS algorithm only determines CDS members by the number of uncovered nodes. Here, the higher the coverage of the node, the lower the CDS members that can be found, causing the MaxD algorithm not suitable for looser network structures or low node degree topologies. In addition, for the highly mobile MANET, the gateway of MaxS-CDS will easy to form a bottleneck which will reduce the lifetime of network.. Besides, the results of the MaxS-CDS algorithm are similar to the proposed IRCS-CDS algorithm, but the distribution of nodes will affect the execution of the MaxS-CDS algorithm. This results will cause the MaxS-CDS often have different numbers of CDS members, which will be an unstable CDS algorithm with a high complexity of $O(3n^2 + 2n)$. It is much higher than the $O(2n^2 + 3n)$ of the proposed IRCS-CDS algorithm. Therefore, the proposed IRCS-CDS algorithm is more stable and more efficient in network environments, even in a network environments with loose nodes or small node coverage radius.

6 Conclusions and Future Works

Building a simple virtual backbone with limited bandwidth, communication capabilities and battery capacity is an important issue. In the past, the virtual backbone network that uses CDS under the MANET environment was considered the most suitable solution. However, the previous algorithms apply the marking procedure to form the CDS topologies with a larger number of CDS members because of the uneven distribution of nodes in the network. Therefore, this research proposes a more efficient and stable IRCS-CDS algorithm by applying the concept of relative complement theory and remark procedure to improve the problems encountered in previous studies. Then, through the experimental results, it can be found that the proposed IRCS-CDS algorithm can obtain better results when the number of nodes is set from 50 to 250 and the transmission distance is set from 20 to 60. Besides, the IRCS-CDS algorithm can reduce the complexity of the algorithm to $O(2n^2 + 3n)$. Therefore, the IRCS-CDS algorithm proposed in this research can efficiently and stably find a smaller CDS members for any environment with different node distribution, number of nodes and different transmission ranges.

Abbreviations

MANETs: Mobile Ad Hoc Networks; CDS: connected dominating set; AP: Access Point; DS: Dominating Set; WANET: Wireless Ad-Hoc Network; DSDV: Destination Sequenced Distance Vector;

AODV: Ad-Hoc On-Demand Distance Vector Routing; DSR: Dynamic Source Routing; CGL: CDS Gateway List; CNL: Covered-Node-List; PL: Priority List; SV: Suffrage Value

Declarations

Competing interests

The authors declare that they have no competing interests.

Funding

The manuscript “Relative Complement Set Based for Connected Dominating Set Algorithm for the Ad Hoc Network” has no funding received.

Authors Contribution

Mao-Lun Chiang: conceived of the presented idea, verified the analytical methods. Hui-Ching Hsieh: conceived of the presented idea, verified the analytical methods, wrote the manuscript. Chia-Fen Hsieh: verified the analytical methods, wrote the manuscript. Wei-Ling Lin: carried out the experiment. Xin-Zhi Lin: carried out the experiment.

References

- [1] Okoro Osahon and Edim Azom Emmanuel, “A Wireless Network Infrastructure Architecture for Rural Communities,” *International Journal of Computer Science & Information Technology*, Vol. 9, No. 3, 2017.
- [2] Piyush Yadav, Rajeev Agrawal, and Komal Kashish, “Performance Evaluation of ad hoc Wireless Local Area Network in Telemedicine Applications,” *Procedia Computer Science*, Vol. 125, pp. 267-274, 2018.
- [3] Filipe Araujo, André Gomes and Rui P. Rocha, “Towards Optimal Convergecast in Wireless Ad Hoc Networks,” *Ad Hoc Networks*, Vol.107, No.1, 2020.
- [4] Ahmet Akbulut, Gökhan Soysal, Aykut Kalaycıoğlu, Asım E. Yılmaz, and Miase Örümlü, “Modeling, simulation and optimization of a wireless ad hoc network's communication performance with regard to node deployment,” *Simulation Modelling Practice and Theory*, Vol. 96, 2019.
- [5] N. Meghanathan and M. Terrell, “An Algorithm to Determine Stable Connected Dominating Sets for Mobile Ad hoc Networks using Strong Neighborhoods,” *International Journal of Combinatorial Optimization Problems and Informatics*, May 2012, Vol. 3, No. 2, pp. 79-92.
- [6] M. G. Kang, Y. B. Kim, J. H. Lee, and W. Choi, “Secrecy Capacity Scaling by Jamming-Aided

- Hierarchical Cooperation in Ad Hoc Networks,” *IEEE Journal of Selected Topics in Signal Processing*, December 2016, pp. 1390-1403.
- [7] Santiago Felici-Castell, Miguel García-Pineda, Jaume Segura-Garcia, Rafael Fayos-Jordan, and Jesus Lopez-Ballester, “Adaptive live video streaming on low-cost wireless multihop networks for road traffic surveillance in smart cities,” *Future Generation Computer Systems*, Vol. 115, pp. 741-755, 2021.
- [8] Weidong Fang, Wuxiong Zhang, Wei Yang, Zhannan Li, Weiwei Gao, and Yinxuan Yang, “Trust management-based and energy efficient hierarchical routing protocol in wireless sensor networks,” *Digital Communications and Networks*, 2021.
- [9] R. Yamamoto, S. Ohzahata and T. Kato, “A hierarchical opportunistic routing with stability information for mobile ad hoc networks,” *Advanced Technologies for Communications (ATC)*, October 2016, pp. 43-47.
- [10] Sudeep Varshney, Chiranjeev Kumar, and Abhishek Swaroop, “Leach Based Hierarchical Routing Protocol for Monitoring of Over-ground Pipelines Using Linear Wireless Sensor Networks,” *Procedia Computer Science*, Vol. 125, pp. 208-214, 2018.
- [11] C. Deepa, and B. Latha, “HHSRP: a cluster based hybrid hierarchical secure routing protocol for wireless sensor networks,” *Cluster Computing*, Vol. 22, pp. 10449-10465, 2019.
- [12] A. Chinnasamy, B. Sivakumar, P. Selvakumar, and A. Suresh, “Minimum connected dominating set based RSU allocation for smartCloud vehicles in VANET,” *Cluster Computing*, Vol. 22, pp. 12795-12804, 2019.
- [13] P. Ramya, and V. Gopalakrishnan, “Proficient algorithms for enhancing topology control for dynamic clusters in MANET,” *Cluster Computing*, Vol. 22, pp. 9715-9726, 2019.
- [14] J. Wu and H. Li, “On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks,” *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, August 1999, pp. 7-14.
- [15] N. Meghanathan and A. Farago, “On the stability of paths, Steiner trees and connected dominating sets in mobile ad hoc networks,” *Ad Hoc Networks*, July 2008, Vol. 6, No. 5, pp. 744-769.
- [16] N. Meghanathan, “An Algorithm to Determine Minimum Velocity-based Stable Connected Dominating Sets for Ad hoc Networks,” *International Conference on Contemporary Computing (IC3)*, August 2010, pp. 206-217.
- [17] Y. C. Tseng, S. Y. Ni, Y. S. Chen, and J. P. Sheu, “The Broadcast Storm Problem in a Mobile Ad Hoc Network,” *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, August 1999, pp. 151-162.
- [18] J. McQuillan, I. Richer, and E. Rosen, “The New Routing Algorithm for ARPANET,” *IEEE Transactions on Communications*, May 1980, Vol. 28, No 5, pp. 711-719.
- [19] J. McQuillan and D. Walden, “The ARPA Network Design Decisions,” *Computer Networks*, Vol. 1, No. 5, pp. 243-289, August 1977.
- [20] K. Sakai, S. C. -H. Huang, W. S. Ku, M. T. Sun and X. Cheng, “Timer-Based CDS Construction

- in Wireless Ad Hoc Networks,” IEEE Transactions on Mobile Computing, 2010, pp. 1388-1402.
- [21] Yajing Li, Kan Yu, Mengling Liu, and Yubing Han, “Selection of Jammer and Analysis on Secrecy Outage Probability for WANETs,” *Procedia Computer Science*, Vol. 174, pp. 604-608, 2020.
- [22] N. Velumyylum and N. Meghanathan, “On the Utilization of ID-based Connected Dominating Sets for Mobile Ad hoc Networks,” *International Journal of Advanced Research in Computer Science*, September 2010, Vol. 1, No. 3, pp. 36-43.
- [23] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, “A Review of Routing Protocols for Mobile Ad hoc Networks,” *Ad hoc networks*, January 2004, Vol. 2, No. 1, pp. 1-22.
- [24] H. Lim and C. Kim, “Flooding in Wireless Ad Hoc Networks,” *Computer Communications*, February 2001, Vol. 24, No. 3-4, pp. 353-363.
- [25] D. Dash, “Restoring virtual backbone of Wireless Sensor Network on sensor failure,” 3rd International Conference on Recent Advances in Information Technology (RAIT), July 2016, pp.29-34.
- [26] S. Murthy and J. J. Garcia-Luna-Aceves, “A Routing Protocol for Packet Radio Networks,” *Proceedings of the 1st Annual ACM International Conference on Mobile Computing and Networking*, November 1995, pp. 86-95.
- [27] C. E. Perkins and P. Bhagwat, “Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers,” *Proceedings of ACM Conference on Communications Architectures*, October 1994, Vol. 24, No. 4, pp. 234-244.
- [28] D. B. Johnson and D. A. Maltz, “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks,” Internet-Draft, draft-ietf-manet-dsr-00.txt, February 1998.
- [29] C. E. Perkins and E. M. Royer, “Ad-hoc On-Demand Distance Vector Routing,” *Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications*, February 1999, pp. 90-100.
- [30] R. Hazarika and H. B. Russell, “A Load-Based Approach for Selecting the Backbone Terminals for a Hierarchical Ad Hoc Network,” *Military Communications Conference (MILCOM)*, November 2014, pp. 1139-1145.
- [31] P. Fly and N. Meghanathan, “Predicted Link Expiration Time Based Connected Dominating Sets for Mobile Ad hoc Networks,” *International Journal on Computer Science and Engineering*, November 2010, Vol. 2, No. 6, pp. 2096-2103.
- [32] K. F. Jea, M. L. Chiang and J. R. Chen, “A Localized Algorithm for Minimizing Connected Dominating Set in Ad Hoc Network,” *National Computer Symposium*, December 15-16, 2005, pp.1-10 °
- [33] C. B. Lin, K. W. Chen, and M. L. Chiang, “MaxSufferage Selection Based Connected Dominating Set for Ad hoc Wireless Networks,” *BAI2013 International Conference on Business and Information*, July 2013, pp. 73.
- [34] F. Dai and J. Wu, “Distributed Dominant Pruning in Ad Hoc Wireless Networks,” *Proceedings of IEEE International Conference on Communications*, June 2003, pp. 353-357.
- [35] F. Dai and J. Wu, “On constructing k-connected k-dominating set in wireless networks,”

Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium, April 2005, pp. 81a-81a.

- [36] J. Wu, "Extended Dominating-Set-Based Routing in Ad Hoc Wireless Networks with Unidirectional Links," IEEE Transactions on Parallel and Distributed Systems, November 2002, Vol. 13, No. 9, pp. 866-881.
- [37] J. Wu, D. Fai, M. Gao, and I. Stojmenovic, "On Calculating Power-Aware Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks," IEEE Journal of Communications and Networks, March 2002, Vol. 4, No. 1, pp. 59-70.
- [38] M. Maheswaran, S. Ali, H. J. Siegal, D. Hensgen, and R. F. Freund, "Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems," Proceedings of 8th IEEE Heterogeneous Computing Workshop, April 1999, pp. 30-44.
- [39] National Academy for Educational Research (NAER), <http://terms.naer.edu.tw/detail/1279653/>
- [40] F. Dai and J. Wu, "An extended localized algorithm for connected dominating set formation in ad hoc wireless networks," IEEE Transactions on Parallel and Distributed Systems, August 2004, Vol. 15, pp. 908-920.

Figure Title

- Figure 1.** The Network architecture of MANETs
- Figure 2.** Example of the Ad-Hoc Network Topology
- Figure 3.** An Example MANET structure
- Figure 4.** The Pseudo-code of the Relative Complement CDS Selection Procedure
- Figure 5.** The Initial Status of the MANET Architecture
- Figure 6.** The First Round of Relative Complement CDS Selection procedure
- Figure 7.** The Second Round of Relative Complement CDS Selection procedure
- Figure 8.** The Third Round of Relative Complement CDS Selection Procedure
- Figure 9.** The SV' Values of Node 5 and Node 7
- Figure 10.** The Fourth Round of Relative Complement CDS Selection Procedure
- Figure 11.** The Fifth Round of Relative Complement CDS Selection Procedure
- Figure 12.** The SV' Values of Node 1 and Node 2
- Figure 13.** The Sixth Round of Relative Complement CDS Selection Procedure
- Figure 14.** The Initial Number of CDS Members after the Marking Procedure with Various Algorithms (R=60)
- Figure 15.** Remove Node 13 by Gateway Remark Procedure
- Figure 16.** The Execution Result of IRCS-CDS in MANET
- Figure 17.** Numbers of CDS for Various Algorithms (R=20)
- Figure 18.** Numbers of CDS for Various Algorithms (R=40)
- Figure 19.** Numbers of CDS for Various Algorithms (R=60)
- Figure 20.** The Initial Number of CDS Members after the Marking Procedure with Various Algorithms

(R=20)

Figure 21. The Initial Number of CDS Members after the Marking Procedure with Various Algorithms

(R=40)

Figure 22. The Initial Number of CDS Members after the Marking Procedure with Various Algorithms

(R=60)

Figure 23. Number of CDS for Various Algorithms (Node size=50)

Figure 24. Number of CDS for Various Algorithms (Node size=150)

Figure 25. Number of CDS for Various Algorithms (Node size=250)

Figure 26. Number of CDS for various algorithms (Degree \in [1,6])

Figure 27. Number of CDS for various algorithms (Degree \in [7,16])

Figure 28. Number of CDS for various algorithms (Degree \in [17,30])

Figure 29. The number of CDS rendering graph Marking program for various algorithms (Degree \in [1,6])

Figure 30. The number of CDS rendering graph Marking program for various algorithms (Degree \in [7,16])

Figure 31. The number of CDS rendering graph Marking program for various algorithms (Degree \in [17,30])

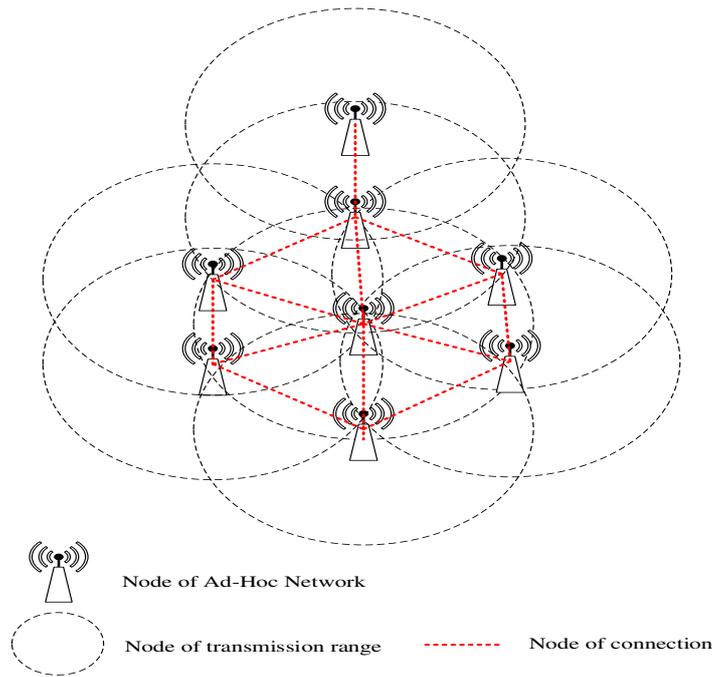


Figure 1. The Network architecture of MANETs

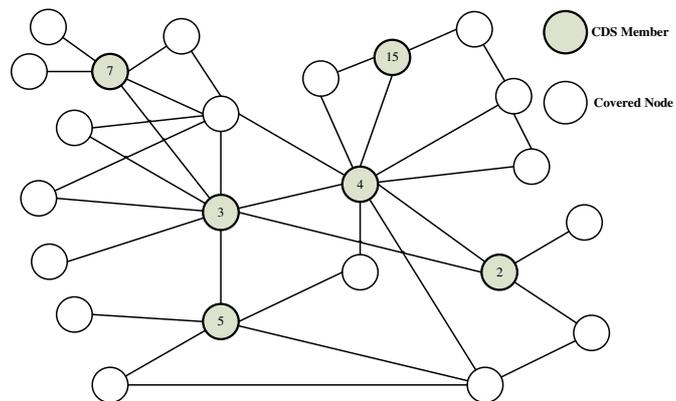


Figure 2. Example of the Ad-Hoc Network Topology

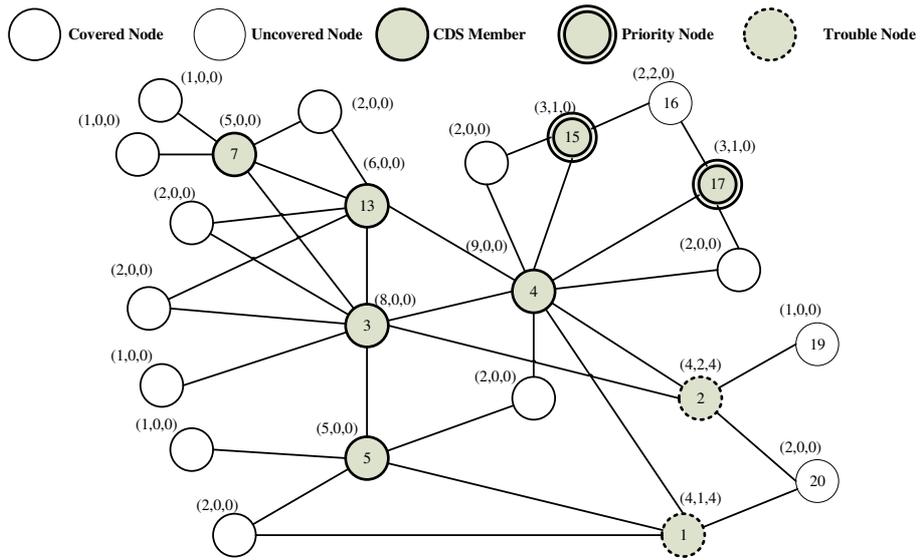


Figure 3. An Example MANET structure

<p>Symbol Definition:</p> <p>N_v: The neighbors of node v</p> <p>D_v: Degree of node v</p> <p>SV'_v: The Suffrage value of node v</p> <p>RC_v: The Relative Complement value of node v</p>
<p>Input: Graph $G = (V, E)$; V represents a set of nodes; E represents a set of edges.</p> <p>Output: CML</p> <p>Initialization: $CML = \emptyset$ $CNL = \emptyset$ $PML = \emptyset$</p> <p>Begin: Calculation of CDS</p> <p>Start vertex, s represents vertex with the largest of the degree in V</p> <p>$CML = CML \cup \{s\}$ $CNL = CNL \cup \{s\}$ $PML = PML \cup N_s$</p> <p>While ($CNL < V$ && $PML \neq \emptyset$) do</p> <p style="padding-left: 2em;">Select a node $v \in PML$ and v has the largest node degree and has at least one neighbor that is not in CNL</p> <p style="padding-left: 2em;">If (degree of node $v =$ degree of node u)</p> <p style="padding-left: 4em;">Calculate RC of node v and u</p> <p style="padding-left: 4em;">If($RC_v = 0$ && $RC_u = 0$)</p> <p style="padding-left: 6em;">Select node v with the least id for next CDS</p> <p style="padding-left: 4em;">end if</p> <p style="padding-left: 4em;">Calculate SV'_v of node v and u</p> <p style="padding-left: 4em;">If($SV'_v = SV'_u$)</p> <p style="padding-left: 6em;">Select node v and u for next CDS</p> <p style="padding-left: 4em;">end if</p> <p style="padding-left: 6em;">Select node v with the largest SV' for next CDS</p> <p style="padding-left: 2em;">End if</p> <p style="padding-left: 2em;">$CML = CML \cup \{v\}$</p> <p style="padding-left: 2em;">For all $x \in N_v$ and $x \notin CNL$</p> <p style="padding-left: 4em;">$CNL = CNL \cup \{x\}$</p> <p style="padding-left: 2em;">end for</p> <p style="padding-left: 2em;">$PML = CNL - CML$</p> <p style="padding-left: 2em;">return CML</p> <p>end Calculation of CDS</p>

Figure 4. The Pseudo-code of the Relative Complement CDS Selection Procedure

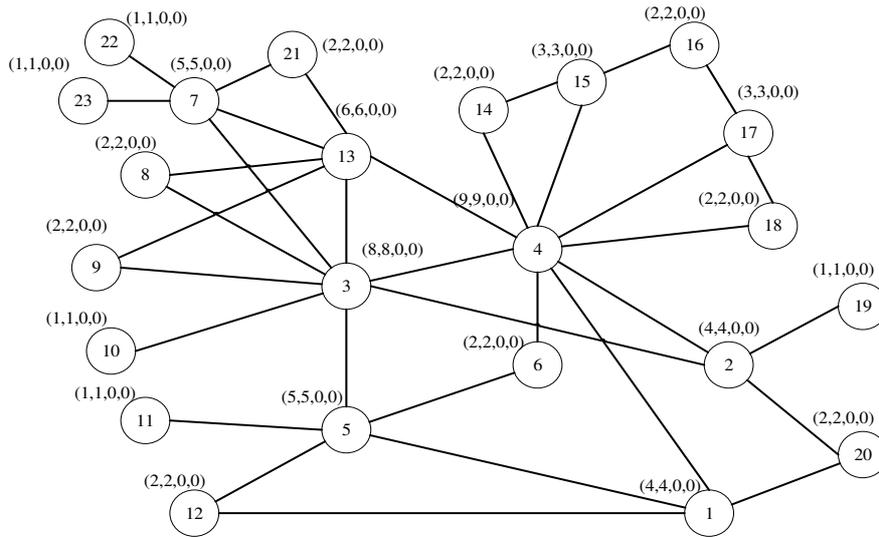


Figure 5. The Initial Status of the MANET Architecture

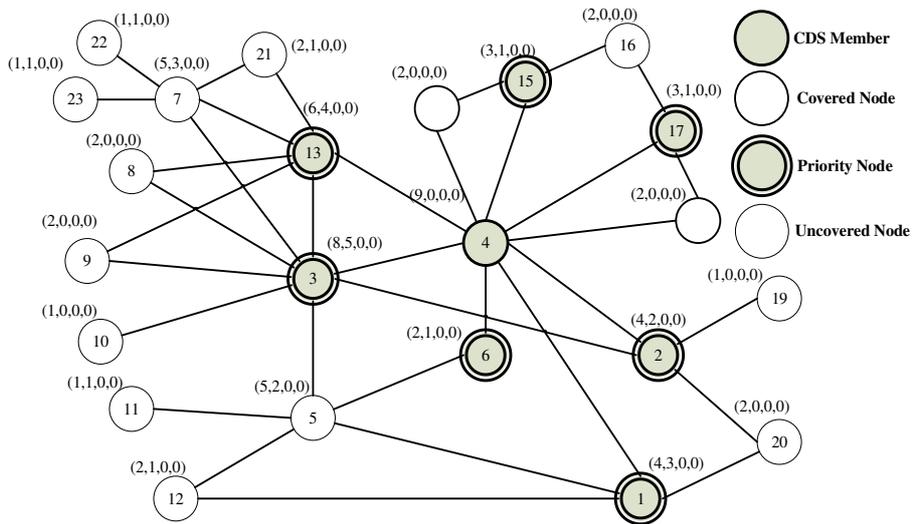


Figure 6. The First Round of Relative Complement CDS Selection procedure

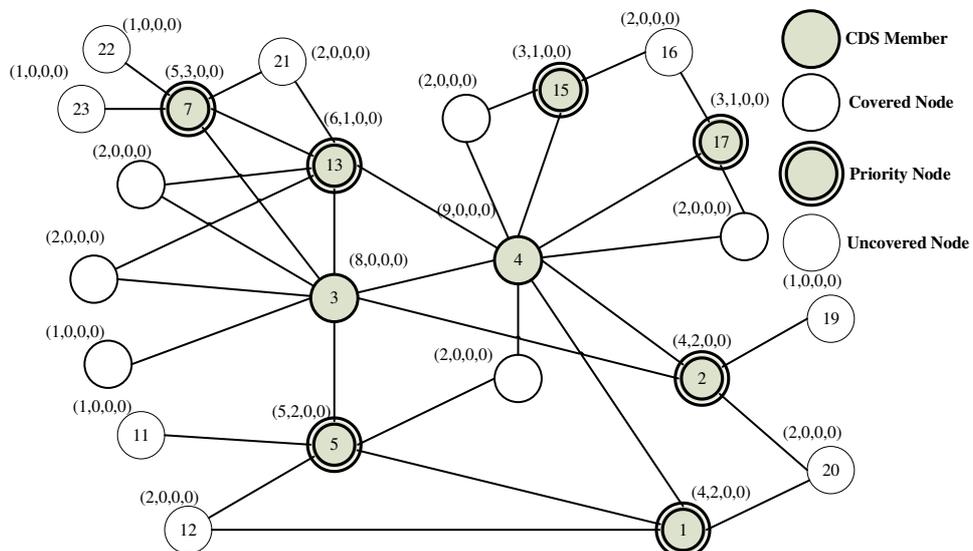


Figure 7. The Second Round of Relative Complement CDS Selection procedure

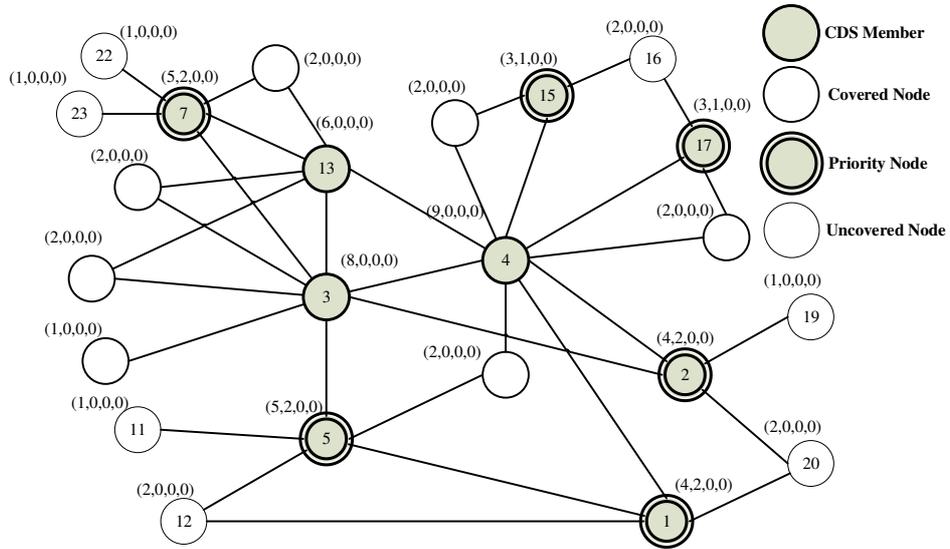


Figure 8. The Third Round of Relative Complement CDS Selection Procedure

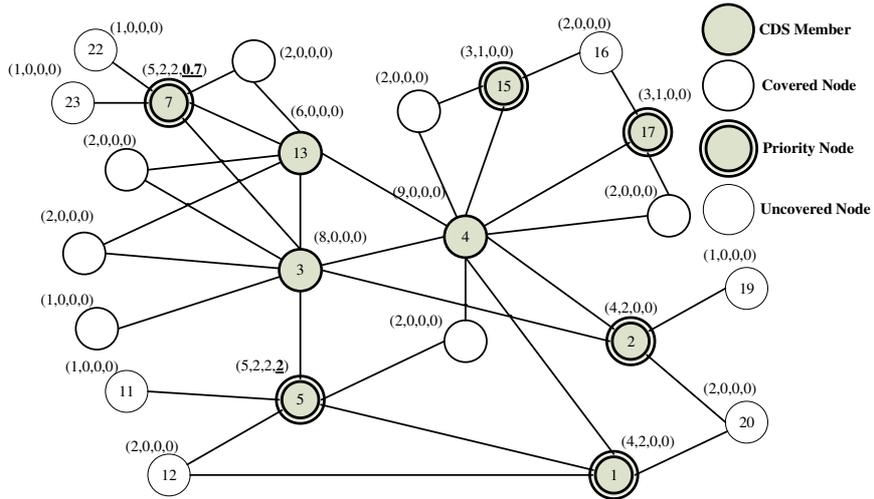
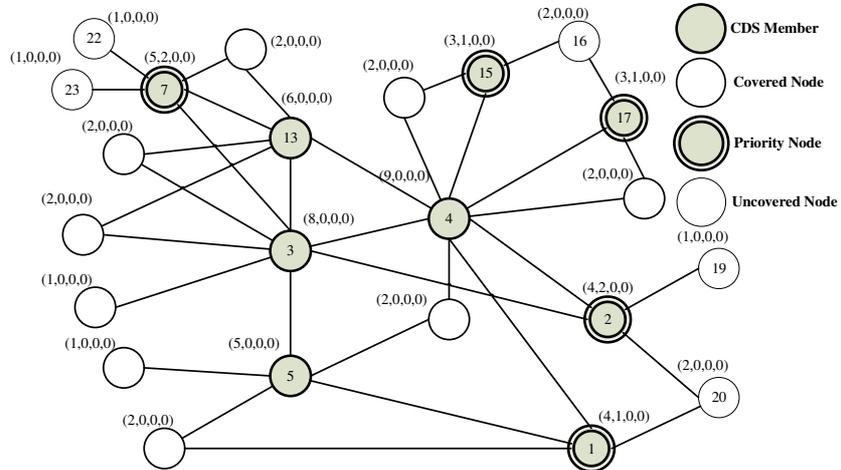


Figure 9. The SV' Values of Node 5 and Node 7



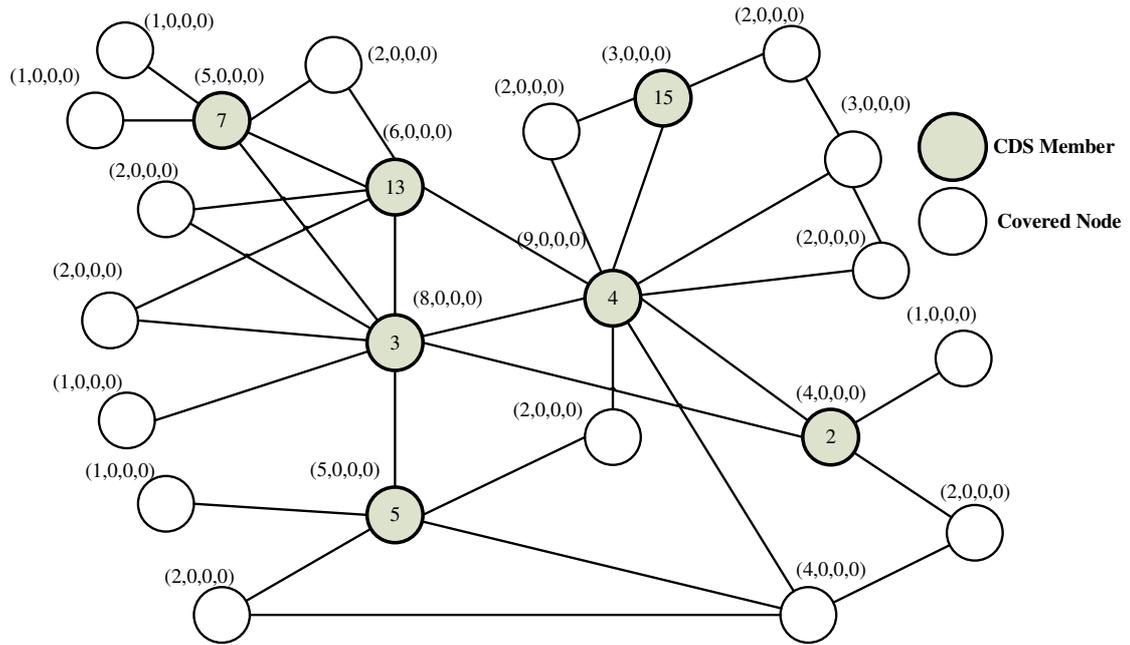


Figure 14. The Final Result of Relative Complement CDS Selection Procedure

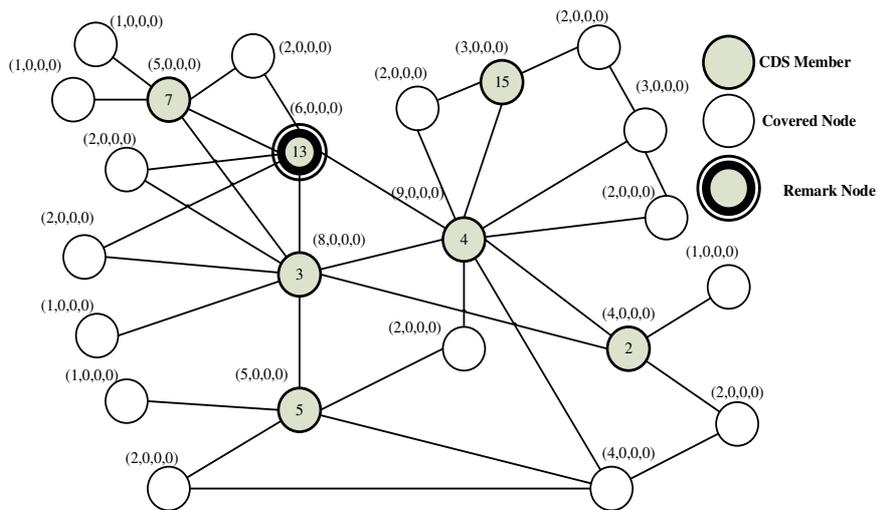


Figure 15. Remove Node 13 by Gateway Remark Procedure

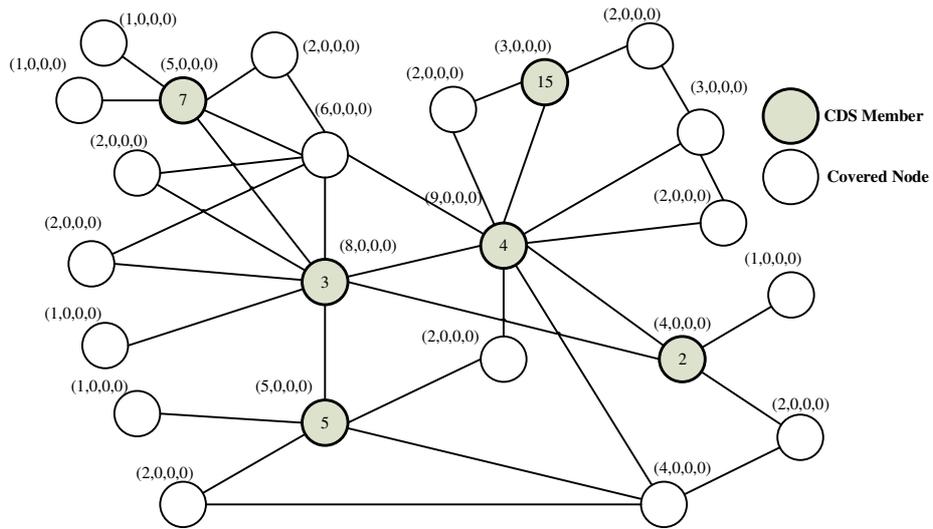


Figure 16. The Execution Result of IRCS-CDS in MANET

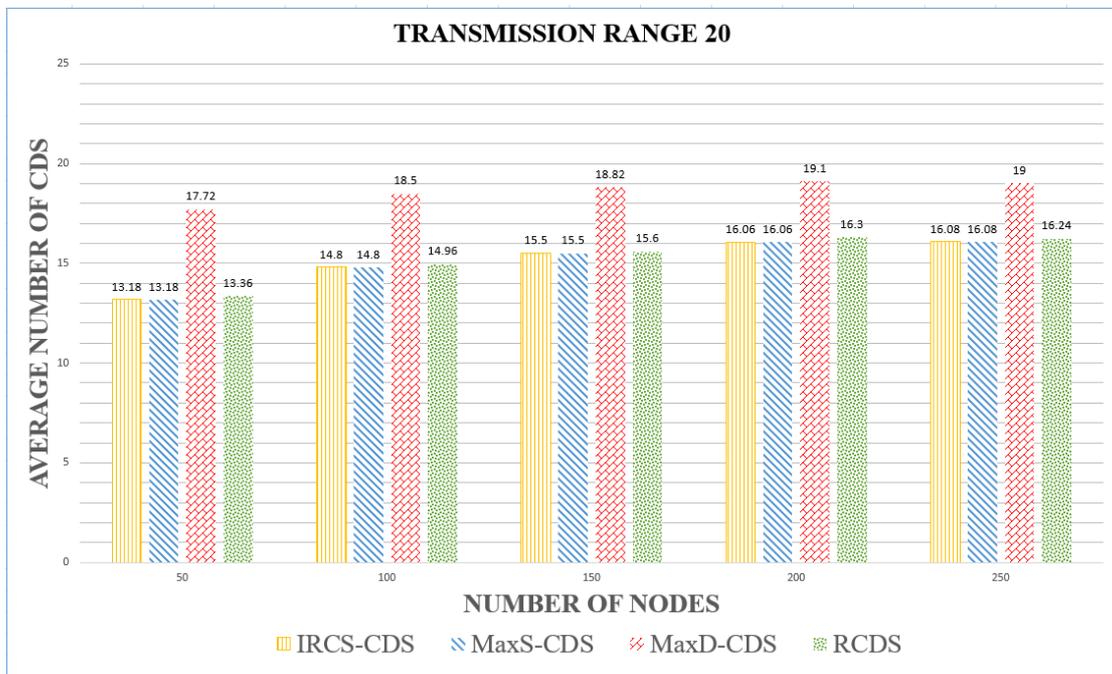


Figure 17. Numbers of CDS for Various Algorithms (R=20)

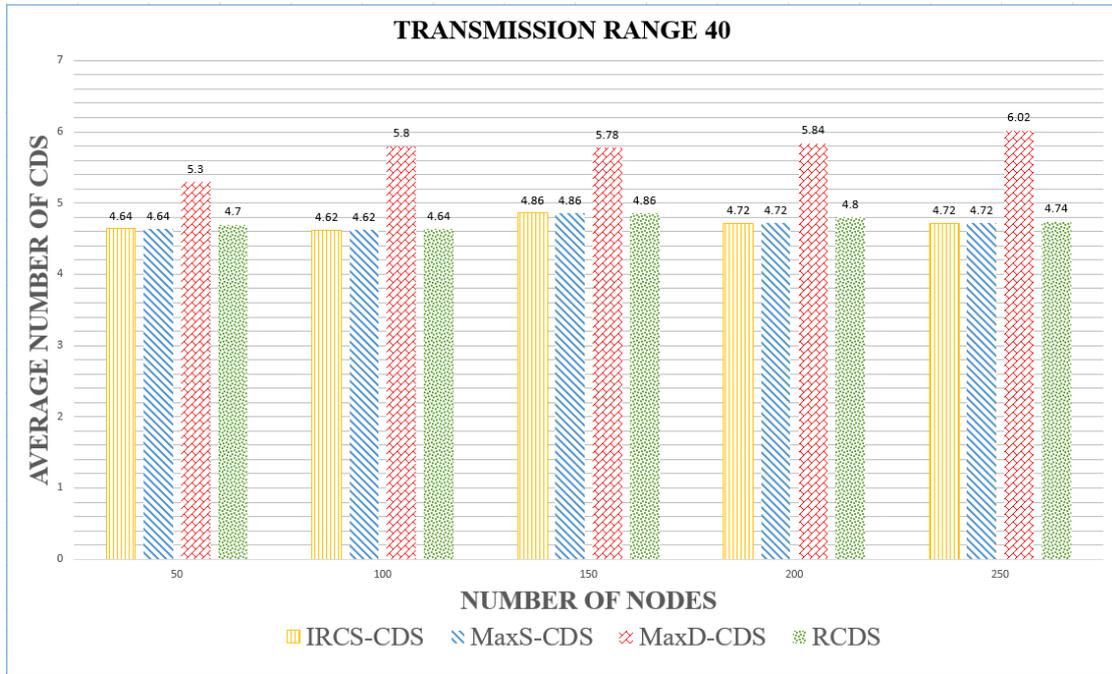


Figure 18. Numbers of CDS for Various Algorithms (R=40)

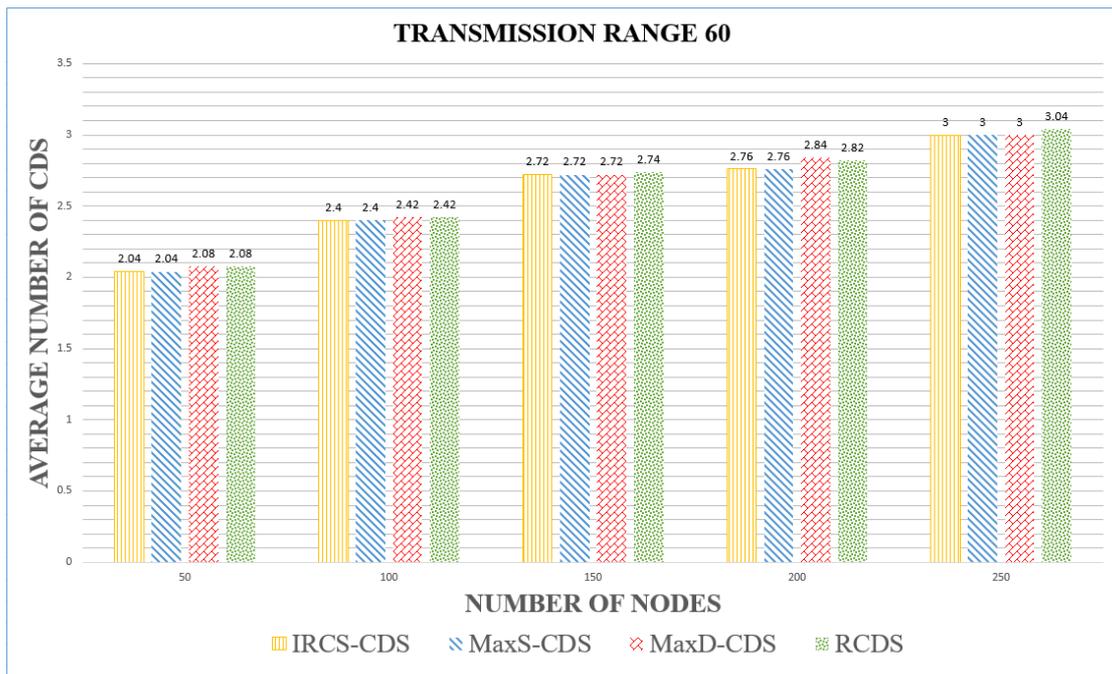


Figure 19. Numbers of CDS for Various Algorithms (R=60)

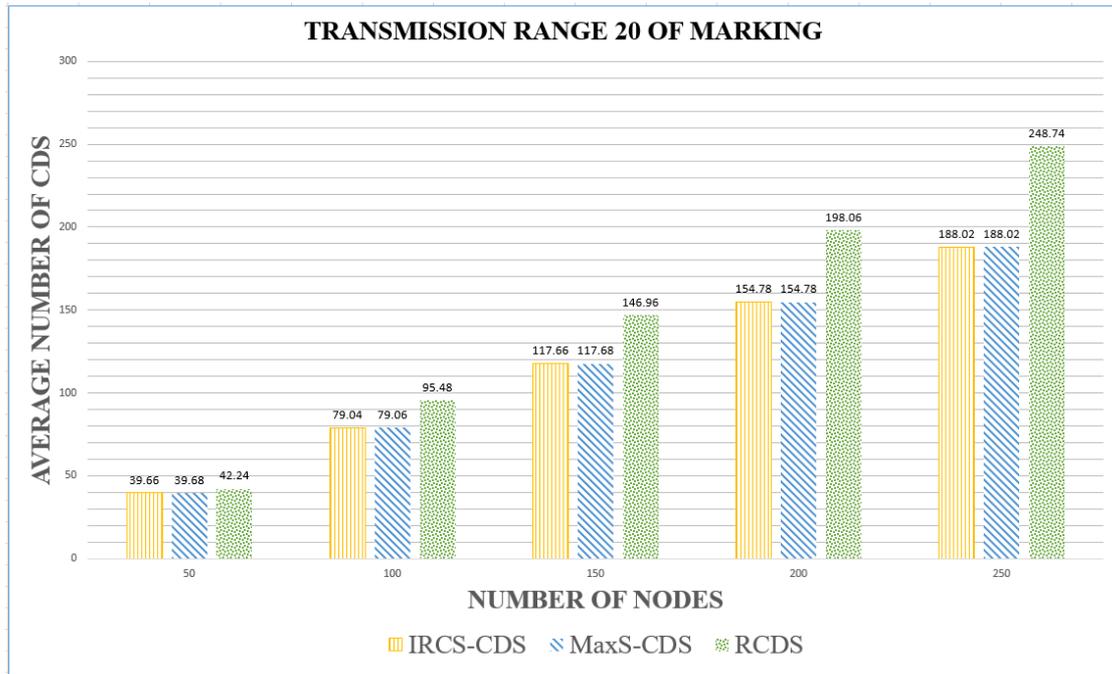


Figure 20. The Initial Number of CDS Members after the Marking Procedure with Various Algorithms (R=20)

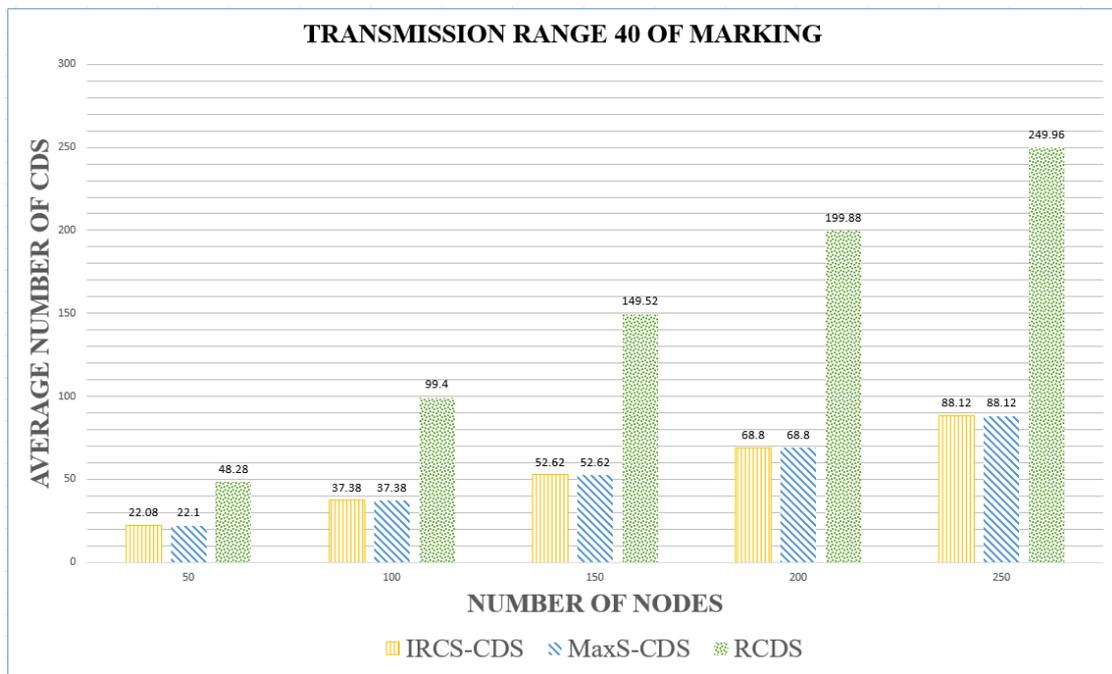


Figure 21. The Initial Number of CDS Members after the Marking Procedure with Various Algorithms (R=40)

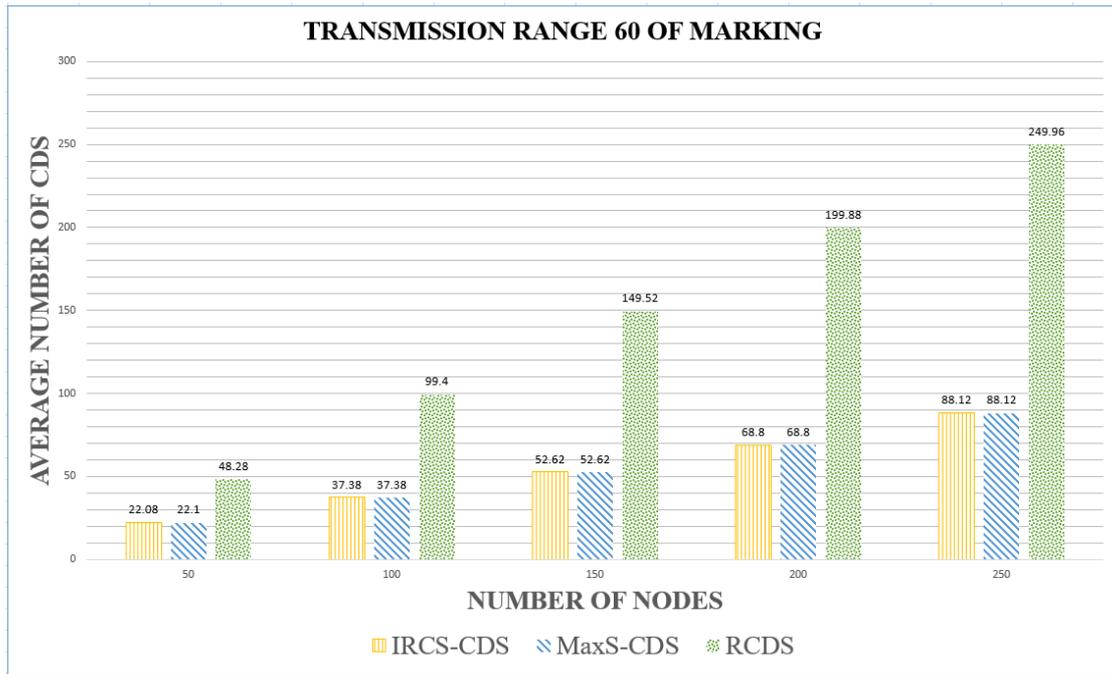


Figure 22. The Initial Number of CDS Members after the Marking Procedure with Various Algorithms (R=60)

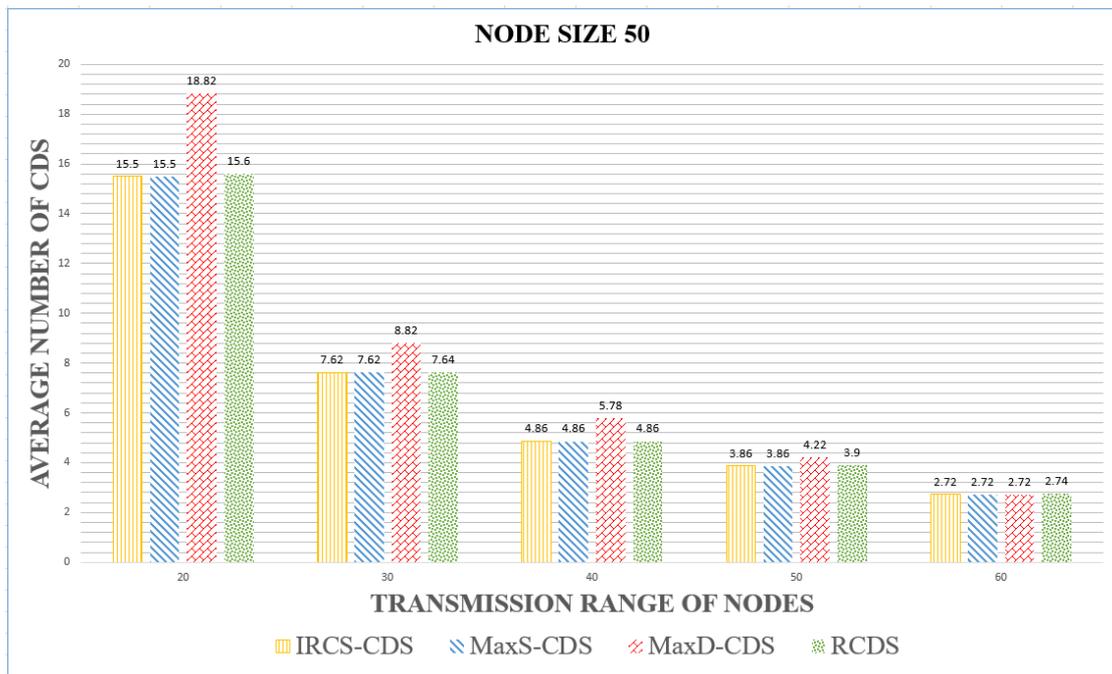


Figure 23. Number of CDS for Various Algorithms (Node size=50)

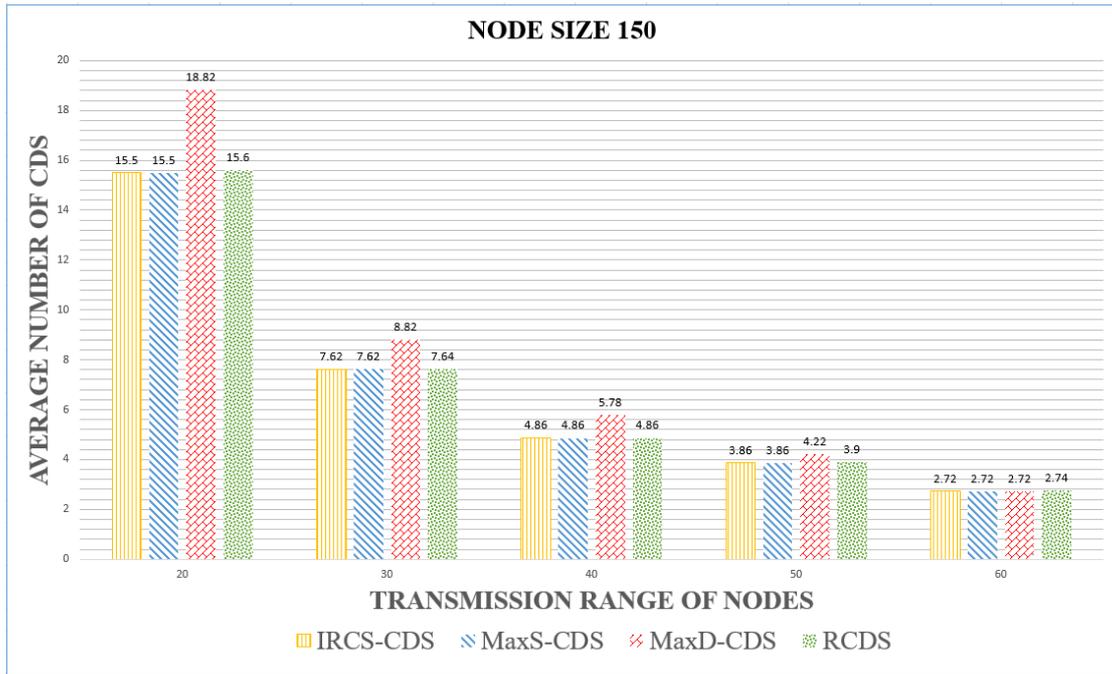


Figure 24. Number of CDS for Various Algorithms (Node size=150)

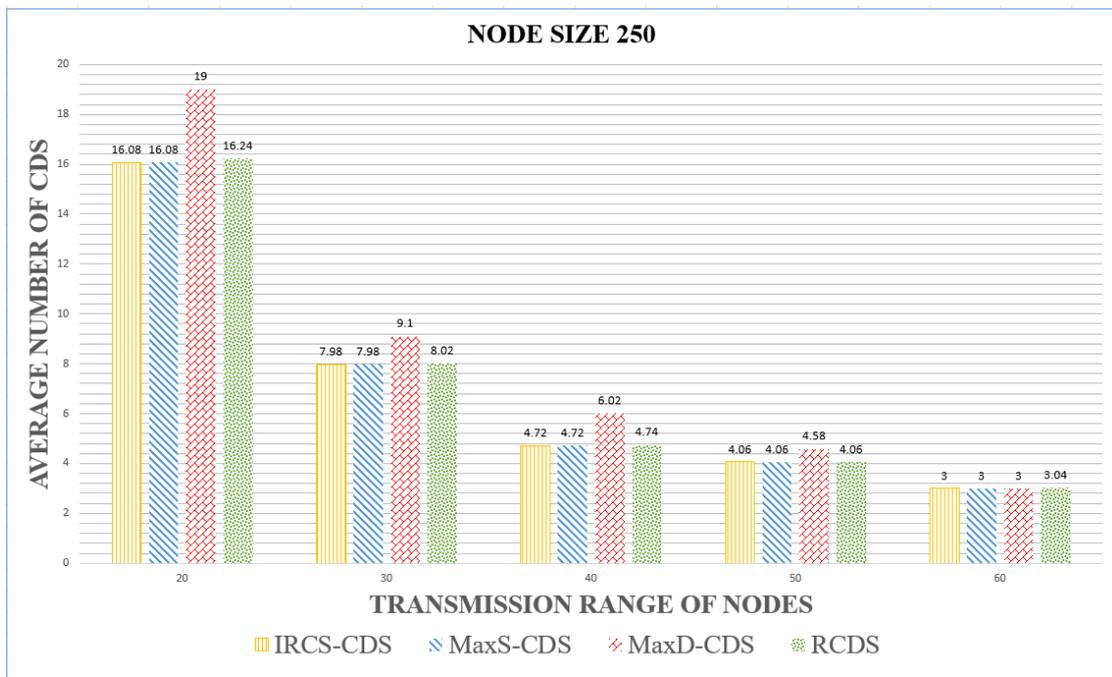


Figure 25. Number of CDS for Various Algorithms (Node size=250)

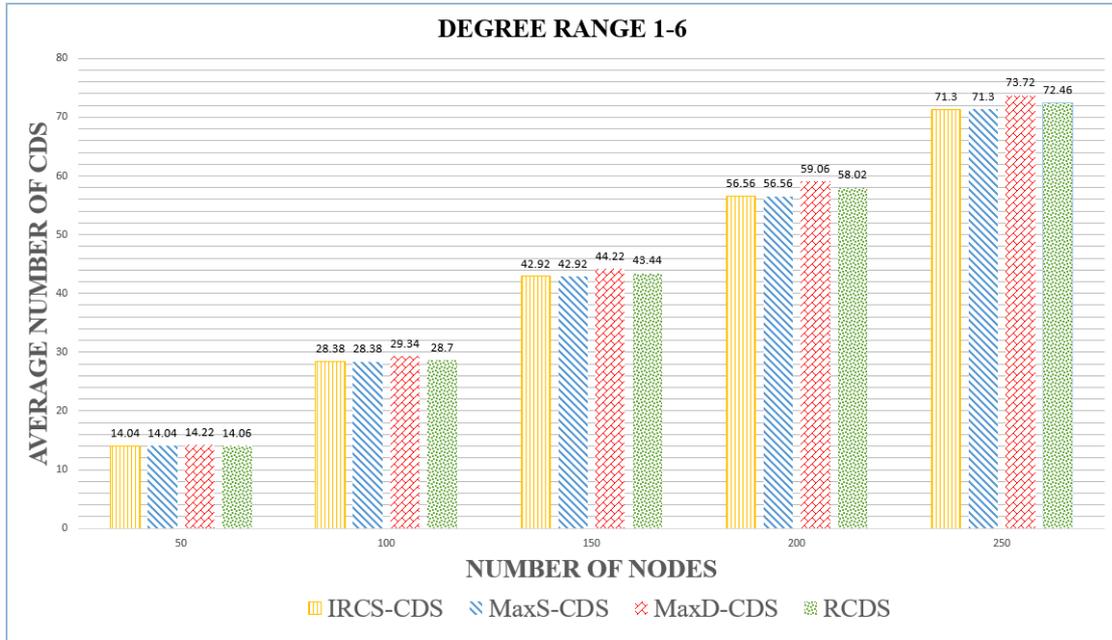


Figure 26. Number of CDS for various algorithms (Degree \in [1,6])

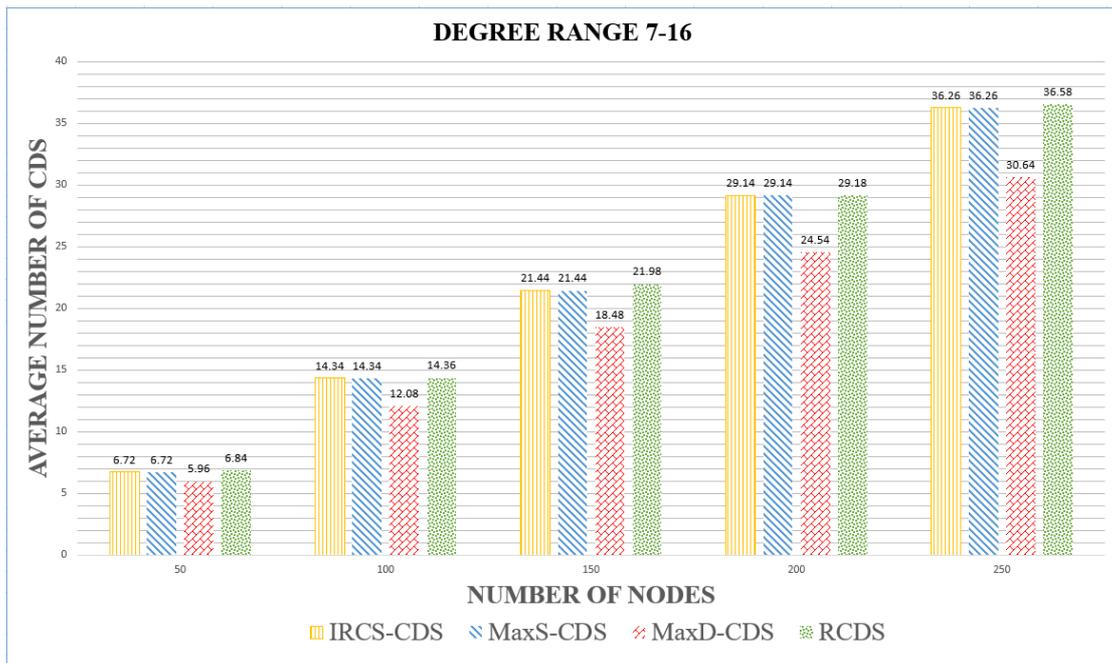


Figure 27. Number of CDS for various algorithms (Degree \in [7,16])

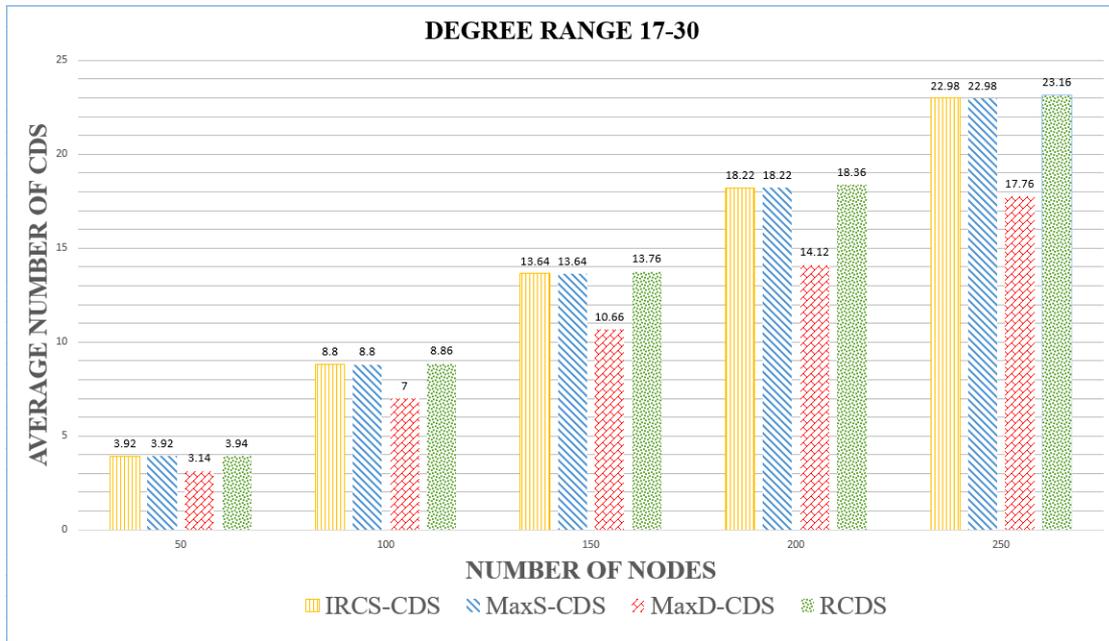


Figure 28. Number of CDS for various algorithms (Degree \in [17,30])

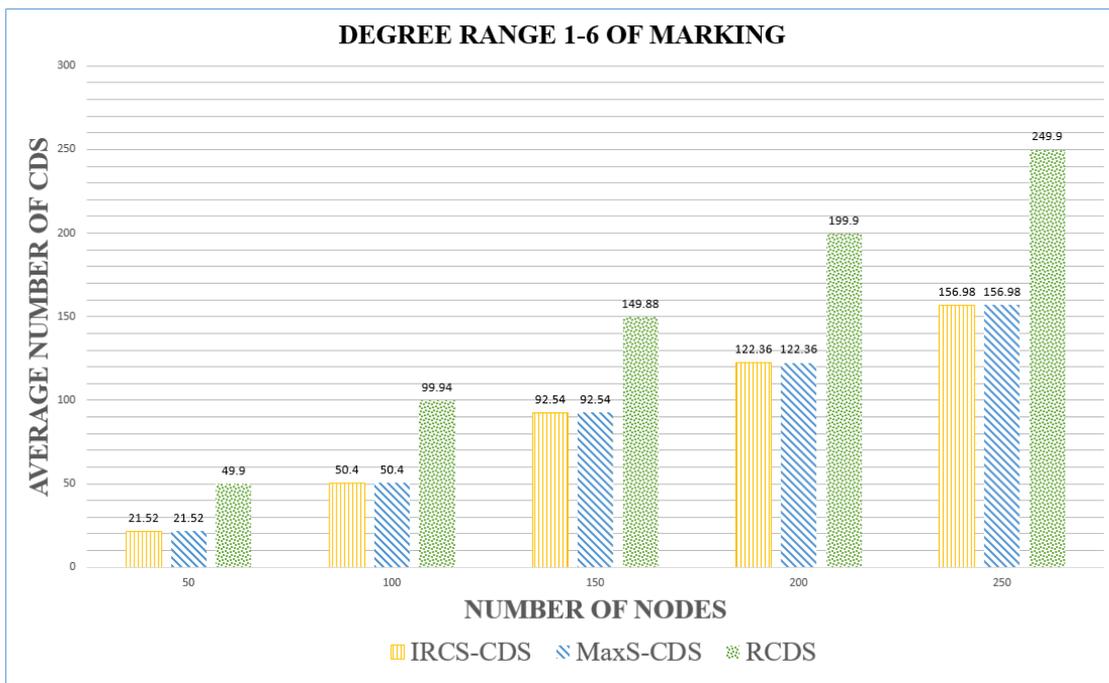


Figure 29. The number of CDS rendering graph Marking program for various algorithms (Degree \in [1,6])

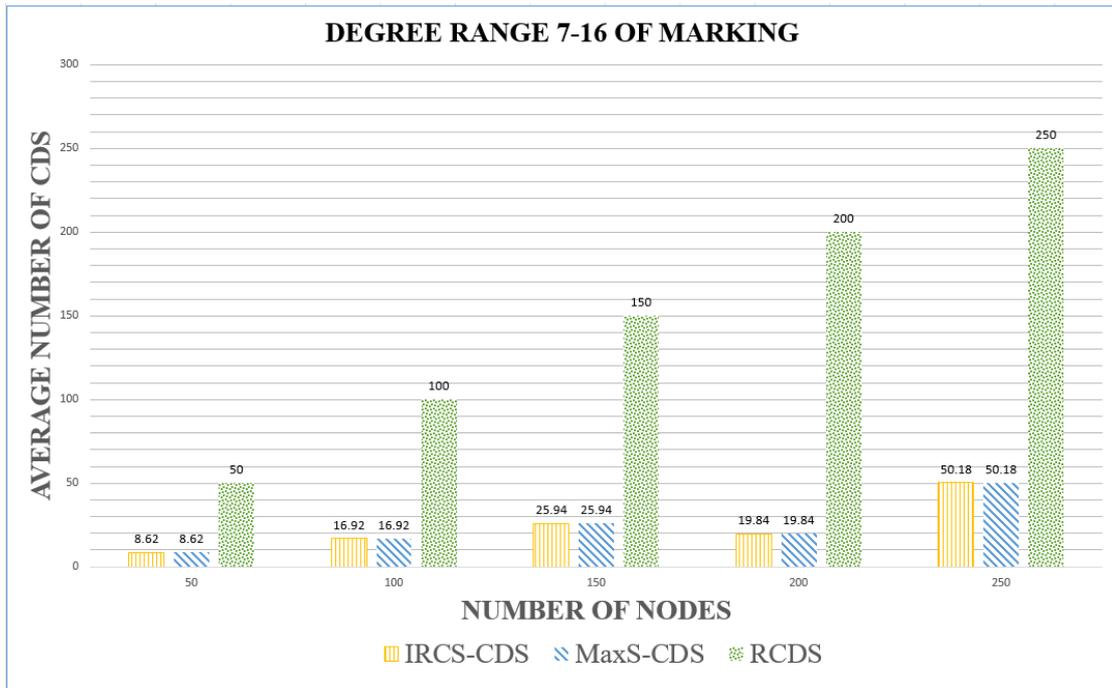


Figure 30. The number of CDS rendering graph Marking program for various algorithms (Degree \in [7,16])

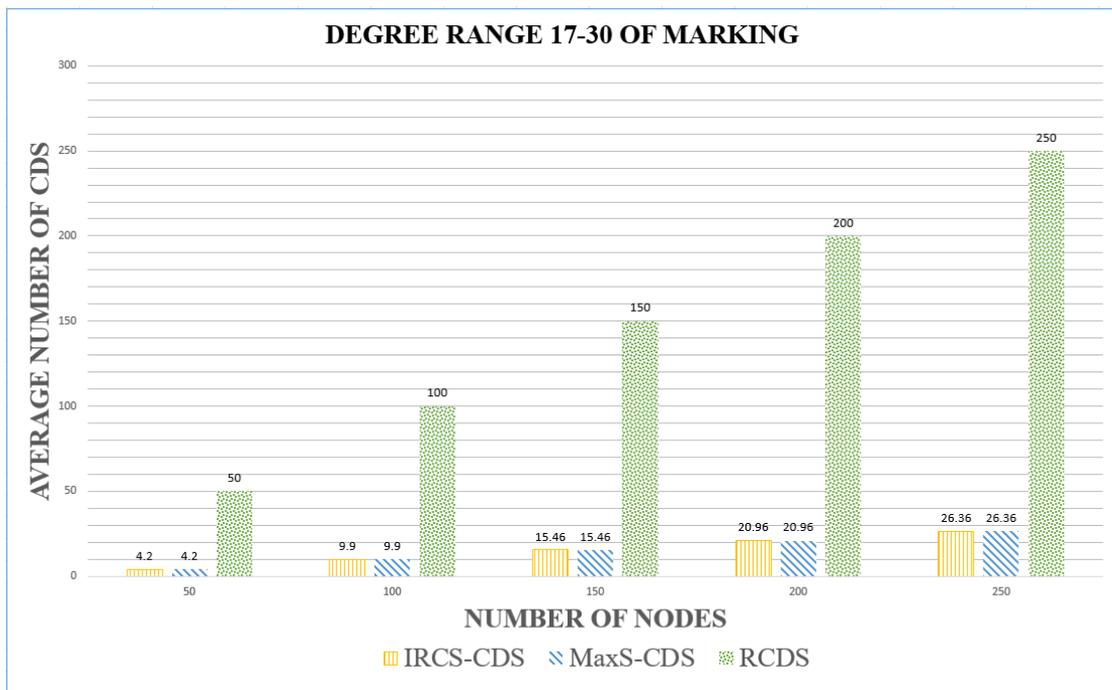


Figure 31. The number of CDS rendering graph Marking program for various algorithms (Degree \in [17,30])

Table 1. Routing Table of Node 2

CDS Member	Covered node	Through the node	Distance
4	{6,13,14,17,18}	2→4	1
15	{14,16}	2→4→15	2
3	{8,9,10,13}	2→3	1
5	{1,6,11,12}	2→3→5	2
7	{13,21,22,23}	2→3→7	2

Table 2. The Results of Each Round in Relative Complement CDS Selection Procedure

Round	CML	PML	CNL
1	4	1, 2, 3, 6, 13, 15, 17	1, 2, 3, 4, 6, 13, 14, 15, 17, 18
2	3, 4	1, 2, 5, 7, 13, 15, 17	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 14, 15, 17, 18
3	3, 4, 13	1, 2, 5, 7, 15, 17	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 14, 15, 17, 18, 21
4	3, 4, 5, 13	1, 2, 7, 15, 17	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 18, 21
5	3, 4, 5, 7, 13	1, 2, 15, 17	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 17, 18, 21, 22, 23
6	2, 3, 4, 5, 7, 13	15, 17	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 17, 18, 19, 20, 21, 22, 23
7	2, 3, 4, 5, 7, 13, 15		1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23