MULTI-INITIATOR CONNECTED DOMINATING SET CONSTRUCTION

FOR MOBILE AD HOC NETWORKS

Except where reference is made to the work of others, the work described in this thesis is my own or was done in collaboration with my advisory committee. This thesis does not include proprietary or classified information.

Kyoung Min Kim

Certificate of Approval:

Wei Shinn Ku
Assistant Professor
Computer Science and
Software Engineering

Min-Te Sun, Chair
Assistant Professor
Computer Science and
Software Engineering

Xiao Qin
Assistant Professor
Computer Science and
Software Engineering

Joe F. Pittman
Interim Dean
Graduate School

MULTI-INITIATOR CONNECTED DOMINATING SET CONSTRUCTION FOR MOBILE AD HOC NETWORKS

Kyoung Min Kim

A Thesis

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirement for the

Degree of

Master of Science

Auburn, Alabama
August 09, 2008

MULTI-INITIATOR CONNECTED DOMINATING SET CONSTRUCTION

FOR MOBILE AD HOC NETWORKS

Kyoung Min Kim

Signature of Author

Date of Graduation

VITA

Kyoung Min Kim, daughter of Nam Sung Kim and Soon Ja Park, was born on February 27, 1980. She was commissioned a lieutenant of infantry in 2003 after graduation from Korea Military Academy with a B.S. degree in Statistical Information Analysis. She served as a platoon leader in 6th infantry division. During serving as a computer engineering officer in Capital Artillery Brigade, she entered Auburn University and joined Dr. Sun's lab in fall 2006.

THESIS ABSTRACT

MULTI-INITIATOR CONNECTED DOMINATING SET CONSTRUCTION FOR MOBILE AD HOC NETWORKS

Kyoung Min Kim

Master of Science, August 09, 2008
(B.S. Statistical Information Analysis, Korea Military Academy, South Korea, 2003)

65 Typed Pages

Directed by Min-Te Sun

Connected dominating set (CDS) has been extensively used for routing and broadcast in mobile ad hoc networks. While existing CDS protocols are successful in constructing CDS of competitive size with localized information, they either lack the mechanism to properly handle nodal mobility or require lengthy period of time to recover when CDS becomes corrupted. In this thesis, a novel protocol, namely Multi-Initiator Connected Dominating Set protocol (MI-CDS), is proposed that constructs and maintains CDS of competitive size efficiently without introducing much communication overhead. The simulation results demonstrate that MI-CDS permits CDS to be available for the highest percentage of time in mobile network scenario compared with the other CDS protocols. Analytical model of the convergence time of MI-CDS as well as the number of messages it requires to construct CDS is built and validated by the simulation results.

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor, Dr. Min-Te Sun for his guide and support which was sincere and everlasting during the entire time of my research. I am deeply grateful for the achieving support of Dr. Wei- Shinn Ku and Dr. Xiao Qin being on my thesis committee. I also appreciate my research group members, Kazuya Sakai and Fangyang Shen, for their support. I want to thank to Mary Virginia Moore, Shuang Li, Lei Zhang, Qing Yang, Eliza Banu, Kyu Bum Hwang, Myung Ho Oh, Suk Bong Park, Hyun Ho Kim, Hwan Sik Lee, Tae Sin Kwak, Kyong Sun Kim, Bo Yeon Jang, Mi Kyong Lee, Chang Hoon Song, Hyung Dae Lee, Sang Hwan Lee, Sang Ho Rho, Tae Hyun Kim, Young Jae Lee, Hoon Suk Rho, and Yoon Sun Yang for their help, advice and friendship. Finally, I give my sincere appreciation to my father, mother and brother for their endless love and support.

Style manual or journal used Guide to Preparation and Submission of Theses and Dissertations

Computer software used Microsoft Word 2007, Excel 2007

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1.

INTRODUCTION

Mobile `ad hoc networks (MANETs) are recognized with the fact that they are suitable in an environment where a fixed infrastructure is insufficient such as disaster recoveries, rescue works, and battlefields. One of the problems regarding MANETs is to design routing protocols for efficient communication between nodes. Although the nature of MANETs [1] makes this problem challenging, it can be resolved through the availability of a virtual backbone. Connected dominating set (CDS) as a virtual backbone is widely used in wireless ad hoc networks. CDS is defined as a subset of nodes in a network such that every node is either in the subset or is neighboring with a node in the set, and the induced graph of the nodes in the set is connected.

In general, the smaller the CDS is, the less communication and storage overhead the protocols making use of it will incur. When it comes to constructing CDS for MANETs, it is desirable that the size of the CDS be as small as possible. On the other hand, it is known that the problem of finding the minimum CDS is NP-hard [3]. The problem becomes even more challenging when no node has the view of the complete network topology. As a result, the existing noticeable MANETs CDS protocols [4] – [6]

emphasize constructing a small CDS distributively with localized information. While they are successful in creating a small CDS, they either lack the mechanism for mobility handling or incur significant overhead when maintaining the CDS. For MANETs where nodes are roaming freely all the time, it is likely that the CDS will not be available for service for a significant amount of time due to either frequent CDS reconstruction or lengthy CDS maintenance should these protocols be adopted.

In this thesis, we propose the Multi-Initiator Connected Dominating Set protocol (MI-CDS). MI-CDS first elects a small number of initiators among nodes in the network, and then grows a dominator tree from each of the initiators distributively, and finally connects these trees via a small number of bridge nodes to form a CDS. Unlike the CDS protocols in [4], [5], MI-CDS is able to handle mobility without the need of reconstructing the CDS from scratch when network topology changes. Compared with the CDS protocol in [6], MI-CDS does not suffer from problems caused by a single point of failure and thus can maintain the CDS more efficiently. The simulations of both static and mobile network scenarios validate that MI-CDS constructs and maintains CDS of competitive size with low overhead. In the case of a mobile network scenario, MI-CDS permits CDS to be available for the highest percentage of time compared with the other CDS protocols. The analytical models of the time of convergence of MI-CDS and the number of messages required by MI-CDS to construct a CDS are also presented and validated by the simulation results.

The rest of this thesis is organized as follows. The existing MANETs CDS protocols are reviewed in Chapter 2. The MI-CDS protocol is presented in detail and how it handles nodal mobility is described in Chapter 3. The simulation results are shown in

Chapter 4. The analytical models are demonstrated and validated in Chapter 5. The conclusion and the future direction of our work are provided in Chapter 6.

CHAPTER 2.

LITERATURE REVIEW

2.1 MOBILE AD HOC NETWORKS

A mobile ad hoc network (or simply MANET) is a system that can be constructed as needed in wireless network, it does not need a centralized administration or fixed infrastructure [14]. MANET has a myriad of possibilities that can be used from disaster recoveries [15] to battlefields [16]. However, because there is no infrastructure, the role of virtual backbone is important [17]. Virtual backbone should be constructed satisfying several conflicting objectives : having small magnitude approximation ratio, reducing extra messages, converging fast, and adopting nodal mobility, etc [18], [19]. MANET can have a virtual backbone formed by mobile hosts in a Connected Dominating Set (or CDS).

2.2 CONNECTED DOMINATING SET PROTOCOL

As mentioned in chapter 1, CDS is defined as a subset of nodes in a network such that every node is either in the subset or is neighboring with a node in the set, and the induced graph of the nodes in the set is connected. Constructing the minimum CDS is known to be NP-hard. While there are some works that discuss how to approximate the minimum CDS under the assumption that the complete network topology is known [3], such an assumption is not practical when it comes to mobile ad hoc networks. More practical approaches satisfying numerous conflicting properties, such as mentioned above, have been proposed. These CDS protocols can be divided into two types. One is centralized CDS protocols and the other is distributed CDS protocols as shown in Figure 2.1.

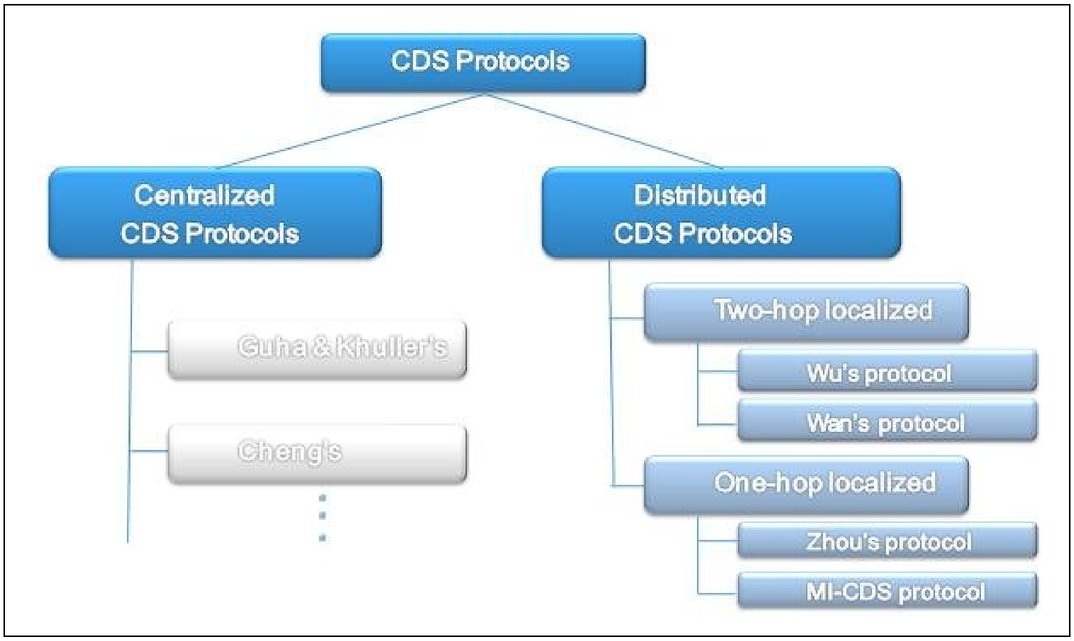

Figure 2.1 : Classification of CDS protocols

2.3 CENTRALIZED CDS PROTOCOLS

Centralized CDS protocols use information of a whole network. Guha and Khuller’s protocol proposed in 1998 [20] and Das et al protocol proposed in 1997 [21] are the well-known examples of the centralized CDS protocol. Nonetheless, applying centralized CDS protocols in MANET is impractical because the global network information has to be gathered to nodes, which can cause high message overhead and the centralized CDS protocols are not adaptive to nodal mobility. As a result, we focus on distributed CDS protocols in our studies.

2.4 DISTRIBUTED CDS PROTOCOLS

In MANET, a networking environment is dynamic and the nodal mobility increases. However, protocol should use a limited knowledge of network topology [13]. In this case, a distributed protocol is more applicable. Due to the lack of a centralized administration, distributed CDS protocols are preferred. Among several distributed CDS protocols, [4]–[6] are well-known protocols. They construct a small CDS based on localized information. Depending on how much localized information is used during CDS construction, these protocols can be classified as two-hop based and one-hop based.

2.4.1 TWO-HOP LOCALIZED CDS PROTOCOLS

The most well-known protocols in this category are Wu’s protocol [4] and Wan’s protocol [5]. We name these protocols two-hop localized CDS protocols since the information of the neighbors that are within two-hops from a node is used in the first step of Wu’s and in the second step of Wan’s.

2.4.1.1 WU’S PROTOCOL

Wu proposed an efficient distributed algorithm for calculating CDS in ad hoc wireless networks [4]. Wu’s CDS protocol is divided by two main phases, marking process for constructing CDS and extensions for removing redundant nodes from the CDS. First, every node collects two-hop neighboring information exchanging messages with its one-hop neighbors. If a node finds all its neighbors are connected to each other, it removes itself from the CDS. The basic marking process is:

1. Initially the marker of every node v, $m(v)$ is F.
2. Each node v exchanges its open neighbor set $N(v)$ with its neighbors.
3. If there are two unconnected neighbors, $m(v)$ is changed to T.

Next, two extensional rules are applied in order to reduce the size of the set. Extensional rules are defined as:

Rule 1) Consider two nodes u, v in dominating sets. If $N(v) \subseteq N(u)$ and $id(v) < id(u)$, $m(v)$ is changed to F.

Rule 2) Consider three nodes u, v, w in dominating sets, u and w are two marked neighbors of v. If $N(v) \subseteq N(u) \cup N(w)$, $id(v) < id(u)$ and $id(v) < id(w)$, $m(v)$ is changed to F.

Although these rules help reduce the size of CDS, as mentioned by Wan's paper [5], Wu's protocol introduces extra messages. Additionally, it does not provide the solution to maintain CDS when network changes. Since Wu's protocol was presented, several various protocols that extended Wu's protocol have been suggested. For example, Nanuvala's protocol [10] extends Wu's protocol to reduce the CDS size. Nanuvala's third extensional rule is as following:

Rule 3) In case of $N(v) \subseteq N(u) \cup N(w)$, condition (1) or (2) has to be satisfied.

(1) $id(v) < id(u) < id(w)$ and u has no neighbor z like

$id(v) < id(z)$ and $N(v) \subseteq N(u) \cup N(z)$.

(2) $id(v) < id(u)$ and $id(w) < id(u)$ and u has no neighbor z like

$$id(v) < id(z),\ N(v) \subseteq N(u) \cup N(z) \text{ and}$$

$$id(w) < id(z),\ N(w) \subseteq N(u) \cup N(z).$$

2.4.1.2 WAN'S PROTOCOL

Another remarkable two-hop localized CDS protocol is Wan's protocol [5]. It also use two-phase approach. The first phase is constructing a maximal independent set (MIS) of the given network topology distributively. The nodes in the MIS form the basis of the CDS. Throughout the MIS construction, a ranking process is induced by a rooted spanning tree by way of the leader-election algorithm. Hence, the second phase constructs a dominating tree from the MIS using local information and adds gateway nodes to create the CDS. In the first phase, whereas the nodes in the MIS are not connected, the distance between any pair of its complementary subsets is known to be exactly two hops away.

Wan's CDS protocol approximates minimum CDS by a constant factor. It takes a small time to construct the CDS. However, it also introduces extra messages during the process of exchanging neighbor list.

2.4.2 ONE-HOP LOCALIZED PROTOCOL

Contrary to the previous protocols, the approach of [6] is based on one-hop localized information when it constructs a CDS protocol. The node receiving a dominator message creates a timer and the node decides whether to join a CDS or not when the timer expires. First, a distributed leader election is used to select an unique initiator. The initiator is the root of a dominator tree and begins to grow dominator tree gradually. The initiator broadcasts to its neighbor about dominating status using a beacon. Thus, when its neighbor receives the dominating message, the status is changed from uncovered to covered. In other words, covered node means that it has an edge with the node in a dominator tree, yet is not in a dominator tree. Then, the covered node starts to set the defer timer based on the number of uncovered neighbors. As Zhou presented in [6], the covered node calculates the defer time, ΔT as follows.

$$\Delta T = T_{\max} \cdot \frac{1}{(numberofunmarkedneighbors)^{\alpha}}$$

, where the maximal defer timer is denoted by $T_{\max}$. When ΔT expires, if all its neighbors are covered by some dominators, the node is switched to the dominatee status. Otherwise, the node joins the CDS to cover its uncovered neighbors. This protocol does not require any extra messages during CDS construction.

Through previous studies, we can know that one-hop localized CDS protocols is better than two-hop ones in terms of the message overhead. Additionally, Zhou’s protocol makes use of periodic heartbeat signals from the initiator to detect mobility events that cause the CDS to corrupt and provides mechanisms to fix the corrupted CDS locally. However, when the initiator fails or moves away from the network, the CDS will have to be fully reconstructed. Another issue of Zhou’s protocol is the duration of the first stage. If it is set to be too long, the CDS construction will take too much time to complete. If it is set to be too short, it may end up having more than one initiator, which will result in a disconnected dominating set.

CHAPTER 3.

MULTI-INITIATOR CONNECTED DOMINATING SET PROTOCOL CONSTRUCTION

3.1 PROTOCOL SKELETON

In this Chapter, we optimize CDS construction based on Zhou's protocol in [6]. The advantage of Zhou's protocol is that it has a smaller amount of message overhead than two-hop localized CDS protocols when they construct CDS as mentioned in the previous Chapter. Although Zhou's makes a lower message overhead, it has a disadvantage that whole CDS has to be reconstructed when a single initiator fails. In order to make up for this weak point, we propose multiple initiators instead of a single one in the initiator election phase. Each initiator generates a tree in exactly the same manner the single initiator does in [6]. This will produce several small disjoint trees and the union of these trees will form a dominating set. In the remainder of this thesis, we refer to the tree generated from an initiator as the dominator tree. To obtain a CDS, we can simply include a few more nodes to connect these dominating trees. If the CDS is constructed this way, the failure of an initiator will only affect the associated dominator tree. The other part of the CDS will remain intact. Table 3.1 illustrates the skeleton of our proposed Multi-Initiator Connected Dominating Set protocol [22]. For convenience, Multi-Initiator Connected Dominating Set is abbreviated as MI-CDS in the following.

Table 3.1 : Multi-Initiator Connected Dominating Set protocol

1. Initiator Election

   /* Initiators are localized elected without introducing extra messages */

2. Tree Construction

   /* Each initiator grows a dominator tree independently */

3. Tree Connection

   /* Additional nodes are included to connect disjoint dominator trees */

In the following subsections, we explain how each of the three phases of our MI-CDS can be accomplished effectively and distributively in detail.

3.1.1 INITIATOR ELECTION PHASE

For the multiple initiators election, we provide an intuitive rule. The rule is to let the local minimum become an initiator. According to IEEE 802.11b specification [7], a node periodically broadcasts the beacon with its unique identifier (id) such as MAC address. Through the beacon, a node can obtain the ids of its one-hop neighbors. If a node's id is smaller compared with its one-hop neighbors' id, it sets itself as an initiator.

The above process does not need any extra messages because the result is acquired by using the beacons. However, it may produce too many initiators. Given that n nodes uniformly distributed in a region of size S and the transmission radius of each node to be r, and the average number of neighbors to be Δ, the average number of initiators n_{init} can be computed by Equation 3.1.

$$n_{init} = n \cdot \left(\frac{1}{\Delta + 1}\right) \approx \frac{S}{\pi r^2} \qquad (3.1)$$

For example, a network of 50 nodes with 150m transmission radius deployed in $1000\text{m} \times 1000\text{m}$ area will have an average number of 14 initiators. Too many initiators will consequently lead to a large CDS due to two reasons. First, all initiators will eventually be included as part of the CDS. Second, after each initiator generates a dominator tree, additional nodes will be added to connect these trees. More initiators mean more trees, and thus require more nodes to connect them.

Since it generates too many initiators to elect local minimum among one-hop neighbors, we suggest the following. To reduce the number of initiators, the local minimum among two-hop neighbors are elected as the initiators. Each node encodes the minimal id among its one-hop neighbors in the beacon. A node becomes an initiator only when its id equals to the minimal id of the one-hop neighbors encoded in the beacon from all of its one-hop neighbors. With few rounds of beacon exchanges, the local minimum among its two-hop neighbors will become the initiator without the extra messages. When we use same denotation with above, the average number of initiators n_{init} is expressed by Equation 3.2.

$$n_{init2} = n \cdot \left(\frac{1}{\frac{n\pi\left(r + \bar{d}\right)^2}{S} + 1} \right) \approx \frac{9}{25} \cdot \frac{S}{\pi r^2} \qquad (3.2)$$

Here, the average distance of each node, $\bar{d}$ is added together with r because an initiator is elected among two-hop neighbors and it is defined by Equation 3.3.

$$\bar{d} = \int_0^r \frac{2\pi x \cdot x}{\pi r^2} dx = \frac{2}{3} r \qquad (3.3)$$

For instance, the network of 50 nodes with 150m transmission radius deployed in $1000m \times 1000m$ area now has an average number of 5 initiators. This significantly reduces the possibility of a large CDS.

3.1.2 TREE CONSTRUCTION PHASE

In this phase, each initiator generates a dominator tree independently and distributively. The process is similar to Zhou's CDS protocol in [6].

Table 3.2 : Tree Construction Algorithm

```
Initiation : $status(x) \leftarrow unmarked$
For all x
    if $status(x) = unmarked \wedge localminimum(x)$ then
        $status(x) \leftarrow initiator$

    if $(x \leftarrow beacon(y)) \wedge (status(y) = initiator \vee status(y) = dominator)$ then
        $y$ becomes dominator of $x$, $root(x) = root(y)$, start $timer(x)$

    if $(timer(x) = 0) \wedge (\exists\ z, (z \in neighbor(x)) \wedge status(z) = unmarked)$ then
        $status(x) \leftarrow dominator$
    else $status(x) \leftarrow dominatee$

    if $(\exists\ a,b \in neighbor(x) \wedge (root(a) \neq root(b))$ then
        $status(x) \leftarrow border$

    if $\exists\ y \in neighbor(x), (status(x) \neq unmarked) \wedge (root(x) \neq root(y))$ then
        $x$ sends message to its initiator
```

Table 3.2 shows the pseudo code of the tree construction phase. For dominator tree generation, a node further includes its status, a dominator’s id if it is covered by a dominator, and an initiator’s id in the extended beacon. The status of a node can be *uncovered* (equals to unmarked in the pseudo code), *covered*, *dominator*, or *dominatee*. At the beginning, every node is in uncovered status. After an initiator is elected, it immediately switches its status to dominator and sets its initiator to itself. When an uncovered node receives the first beacon from a dominator neighbor, it sets its dominator to that neighbor, its initiator to that neighbor’s initiator, switches its status to covered, and sets a timer inversely proportional to the number of uncovered neighbors. When the timer expires, if the node still has uncovered neighbors, it switches its status to dominator to cover these neighbors. Otherwise, it switches its status to dominatee.

At the end of the process, each node is either a dominator or a dominatee. The dominators belonging to an initiator form a tree rooted at the initiator, which is referred to as the dominator tree. If two dominator trees are close to each other i.e., separated by either one or two covered neighbors, they are referred to as neighboring dominator trees.

3.1.3 TREE CONNECTION PHASE

In this phase, additional nodes are included to connect neighboring dominator trees. Table 3.3 shows the pseudo code of tree connection phase. If a node has a neighbor belonging to a different initiator, it is referred to as a border node. To connect the dominator trees distributively, the most intuitive idea is to have all the border nodes turn to dominators. Although this approach does not introduce extra message, it will create a very large CDS, as there are possibly many border nodes between each pair of neighboring dominator trees. To limit the size of CDS, it is better that the root of a tree (i.e., initiator) determines what border nodes to use to connect to the neighboring trees.

Since an initiator does not know what neighboring trees it has with only the localized information, extra messages have to be introduced so that the initiator can collect such information from its border nodes. After the defer timer expires, if a node finds that it is a border node, it sends its initiator the initiator's id of the neighboring tree. For instance, in Figure 3.1, a border node P belonging to initiator A finds that one of its neighbors X belongs to initiator B from B's beacon, P then sends a message to A containing the id of the initiator B. Similarly, X will send a message to B containing the id of the initiator A.

Table 3.3 : Tree Connection Algorithm

```
For $\exists\ x$
    if $status(x) = initiator$
        if($x \leftarrow message(y) \wedge (x \in p(y))$ then
            update $x$'s state
        if $timer(x) = 0$ then
            $x$ sends message to connect tree

    if $status(x) = dominator$
        if $timer(x) = 0$ then
            $p(x) : \leftarrow message(x)$
        if ($message(y) \rightarrow x) \wedge (x \in p(y))$ then
            update $x$'s state
        if $message(y) \rightarrow x) \wedge (x = p(y))$ then
             $x$ send message to connect tree

    if  $status(x) = border$
        if ($message(y) \rightarrow\ x) \wedge (y \in p(x))$ $then$
            $x$ connects tree, $status(x) = dominator$
```

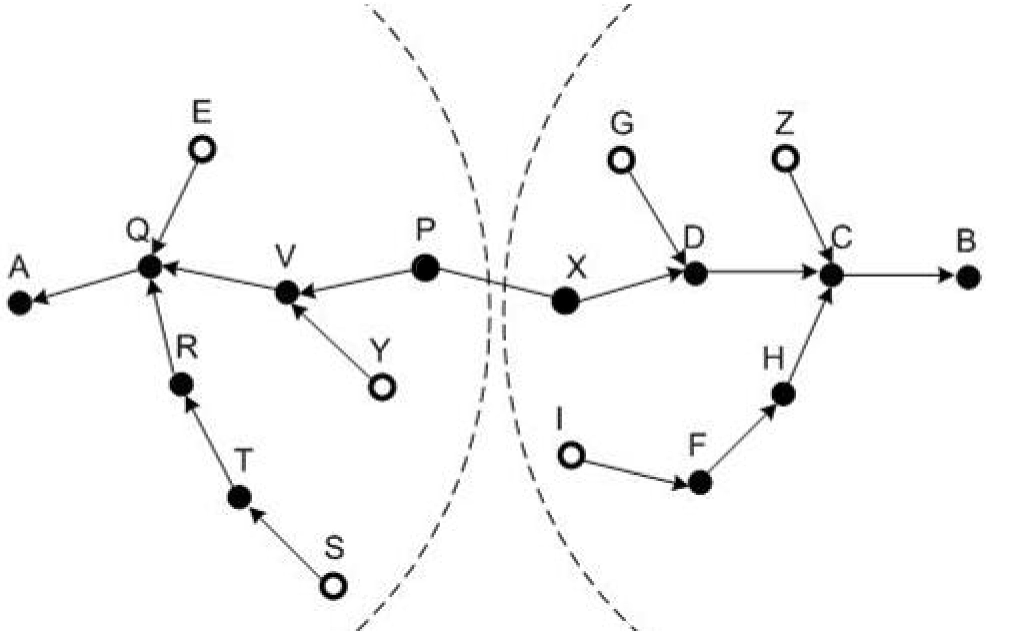

Figure 3.1 : Example of Tree Connection

At the first glance, this process seems to introduce many messages. However, when a dominator receives messages from the neighbors it covers about the neighboring trees, it only forwards one copy of the messages if the initiator id contained in the messages are the same. For instance, in Figure 3.1, when node V receives messages from P and Y about the same neighboring dominator tree, it only forwards one copy of the message to ids dominator Q. Similarly, Q only forwards one copy of the message it receives from E, V and R to its dominator A. By doing this, the number of messages can be controlled to $O(n)$, where n is the number of nodes in the network.

When an initiator learns about its neighboring trees, it can then instruct only border nodes on a particular path to each of its neighboring trees to switch the status to dominator and connect to its neighboring trees. The border nodes that are used to connect the trees are referred as the bridge nodes. Our CDS consists of the dominator nodes in the dominator trees and the bridge nodes that connect the trees.

If each initiator tries to connect to each of its neighboring dominator trees, it is likely that there will be two paths between each pair of neighboring dominating trees. In the worst case, at most four border nodes (two for each path) will become dominators. While having two paths between neighboring dominator trees may improve the degree of fault tolerance and system throughput, it will create a larger CDS. To limit the size of CDS, an initiator makes a connection to a neighboring dominator tree only when the id of the initiator of the neighboring tree is smaller than its own. This roughly reduces the number of the bridge nodes by half.

3.1.4 THE EXAMPLE OF MI-CDS PROTOCOL EXECUTION

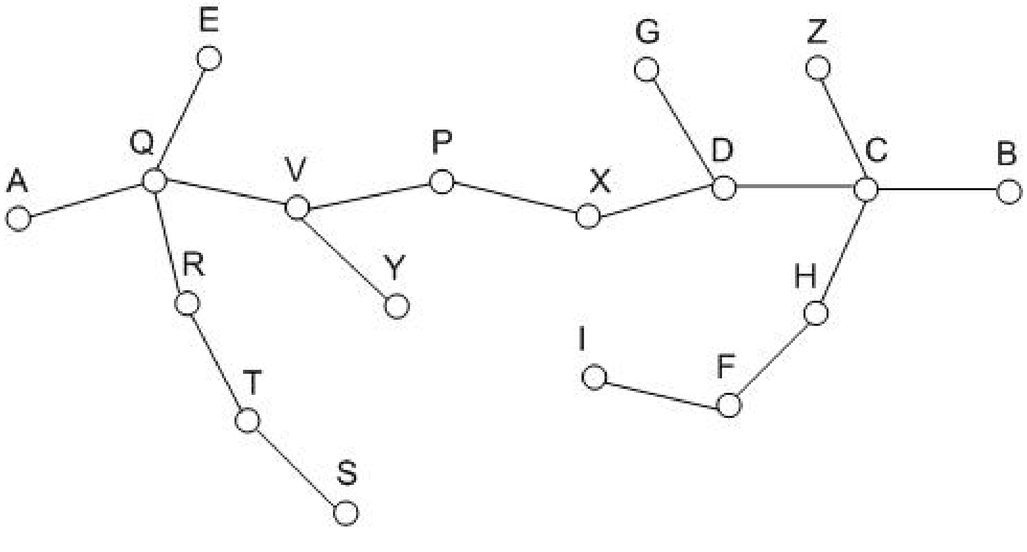

Figure 3.2 : Initial State

This section explains how to execute MI-CDS protocol step by step by using a sample graph. Figure 3.2 shows 18 nodes in a two dimensional area. We assume that each alphabet of the nodes represents each node’s id.

Figure 3.3 presents the result when we select the local minimum among one-hop neighbors in the initiator election phase. Node A, B, E, F, P, and S become the initiators because they have the smallest id. However, there are too many initiators for CDS. Thus, we apply the second rule.

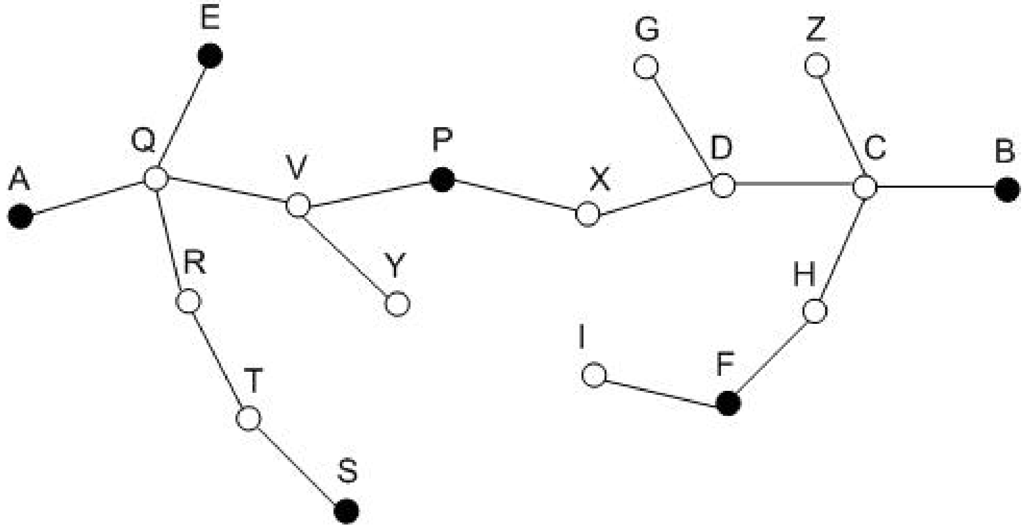

Figure 3.3 : Initiator election among one-hop neighbors

The two-hop neighbors of the 6 initiators except remaining nodes are as follows:

$N(A) = \{E, Q, R, V\}$, $N(B) = \{C, H, Z\}$, $N(E) = \{A, Q, R, V\}$,

$N(F) = \{C, H, I\}$, $N(P) = \{D, Q, V, X, Y\}$, $N(S) = \{R, T\}$

Because node E, F, P, and S have at least one neighbor with an id smaller than their own, they are excluded from the initiators and only node A and B become the initiators as shown in Figure 3.4.

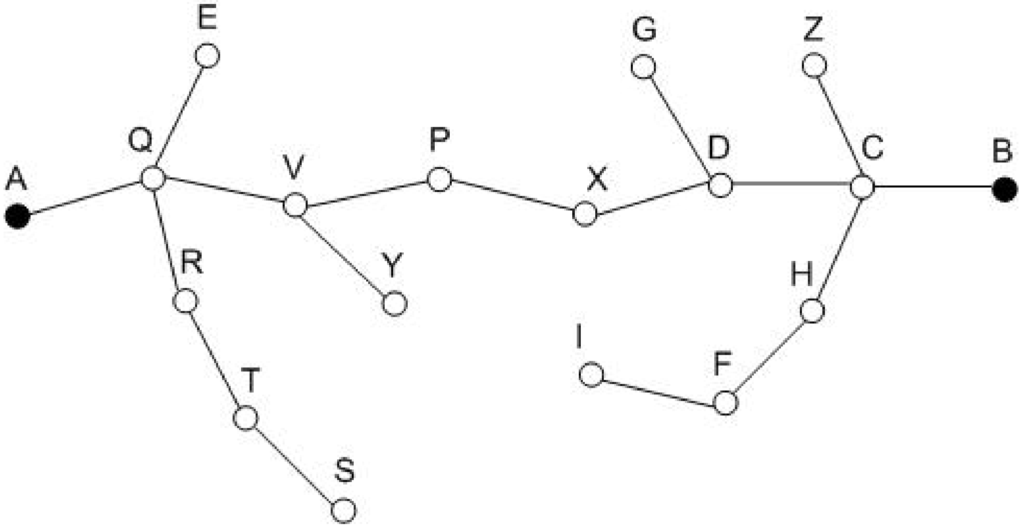

Figure 3.4 : Initiator election among two-hop neighbors

Now, two trees can be generated by two initiators of the former phase. We assume that a black node stands for a dominator node, a gray node denotes a covered node, a bold lined node denotes a dominatee node, and a white node is an uncovered node to easily observe each node’s state change. Following the MI-CDS tree construction algorithm, the initiators, A and B become dominators. And node C, Q switch their state from uncovered to covered after receiving a beacon from them as described in Figure 3.5.

Node C, Q set a timer inversely proportional to the number of uncovered neighbors. When the timer expires, they set their state to dominator as shown in Figure 3.6.

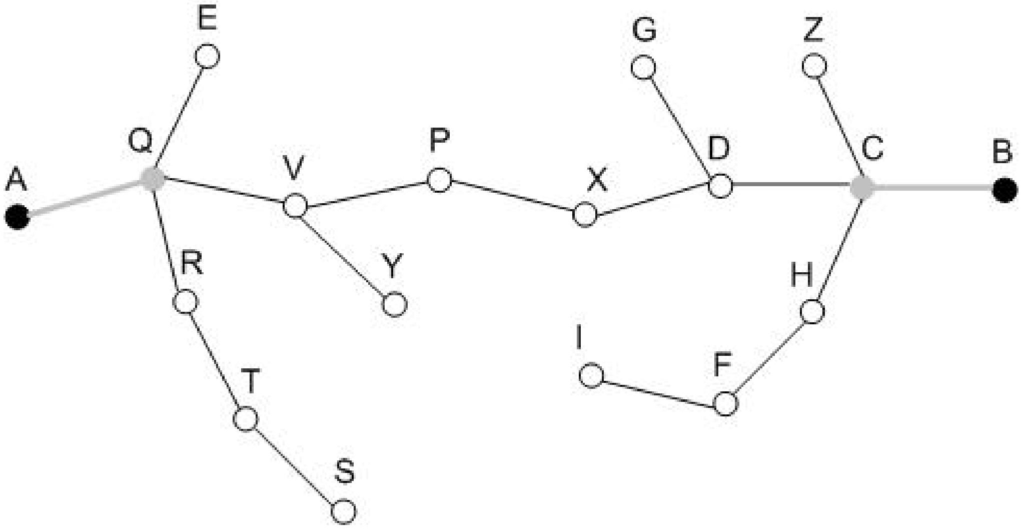

Figure 3.5 : Node A, B become dominator, node C, Q are covered

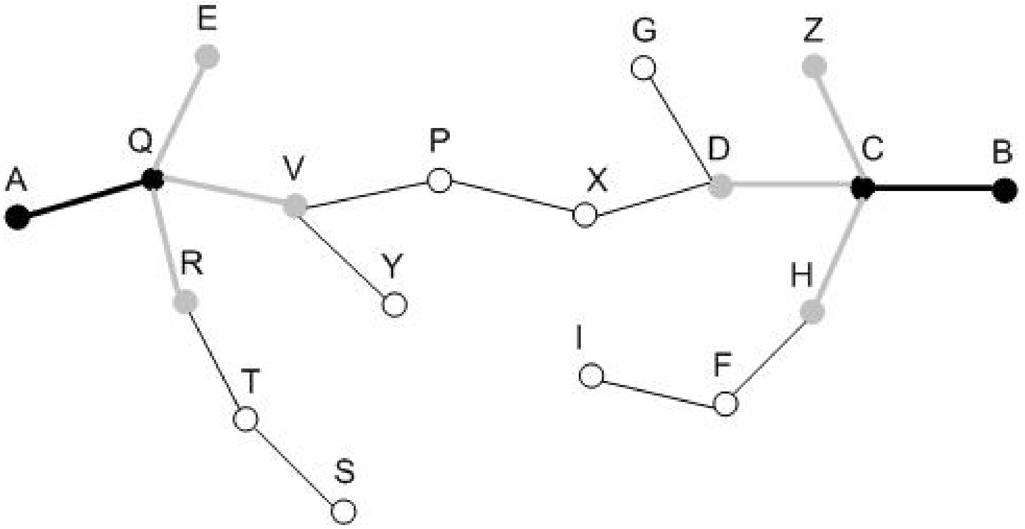

Figure 3.6 : Node C, Q become dominator, node E, R, V, D, H, and Z are covered

Immediately after that, node C's neighbors, E, R, V and node Q's neighbors, D, H, Z switch to covered state. According to the same rule, node R, V, D, and H become dominators. But node E, Z change their state to dominatee because they do not have uncovered neighbors as illustrated in Figure 3.7.

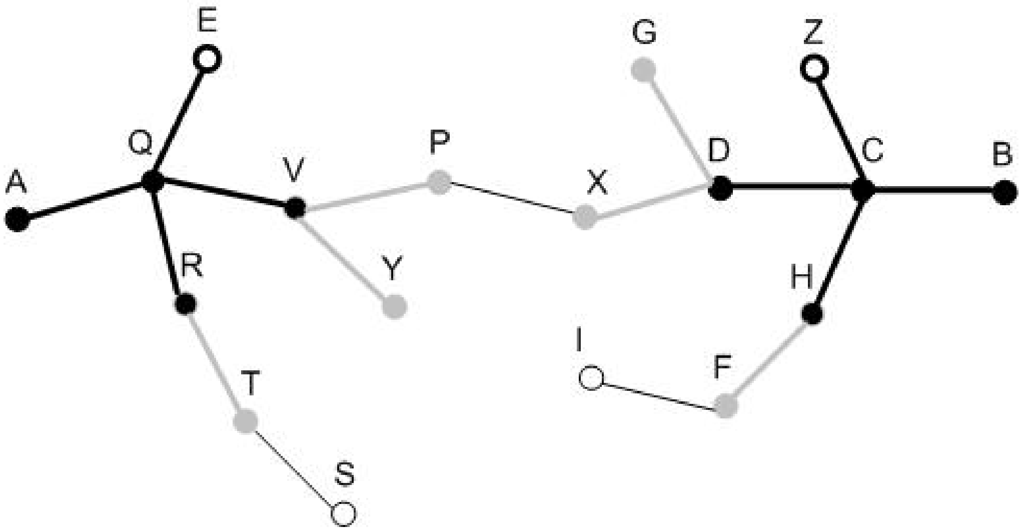

Figure 3.7 : Node R, V, D, and H become dominator and node E, Z become dominatee

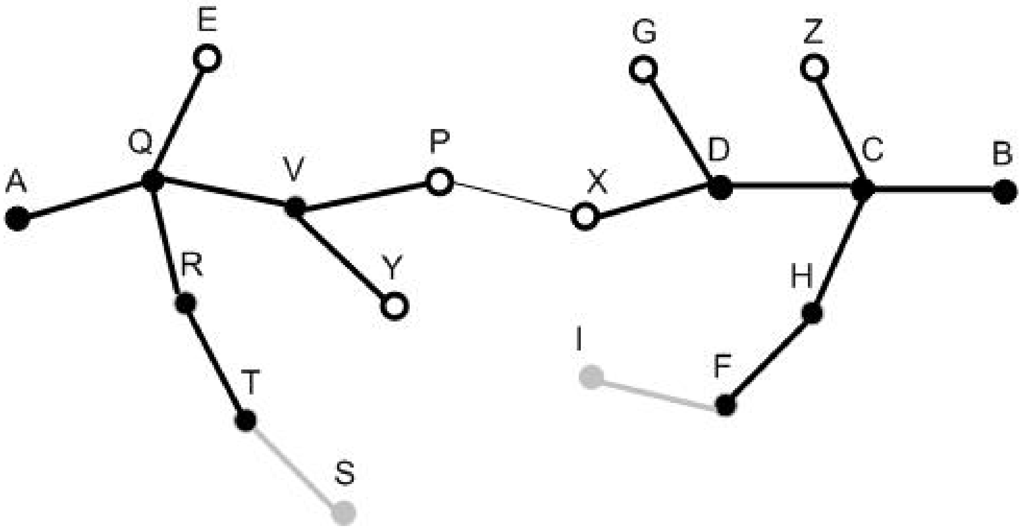

Figure 3.8 : Node T, F become dominator, node S, I become covered node, node P, Y, X and G become dominatee

Figure 3.8 describes next step node’s state change that is applied to the former unchanged rule. Node T, and F set their timer and they switch their state to dominator after the timer expires. Their neighbors, node S and I become covered nodes and node P, Y, X, and Z become dominatees. Node S and I become dominatees too because they could not find an uncovered neighbor when their timer expires. Finally we can obtain two dominator trees as we see in Figure 3.9.

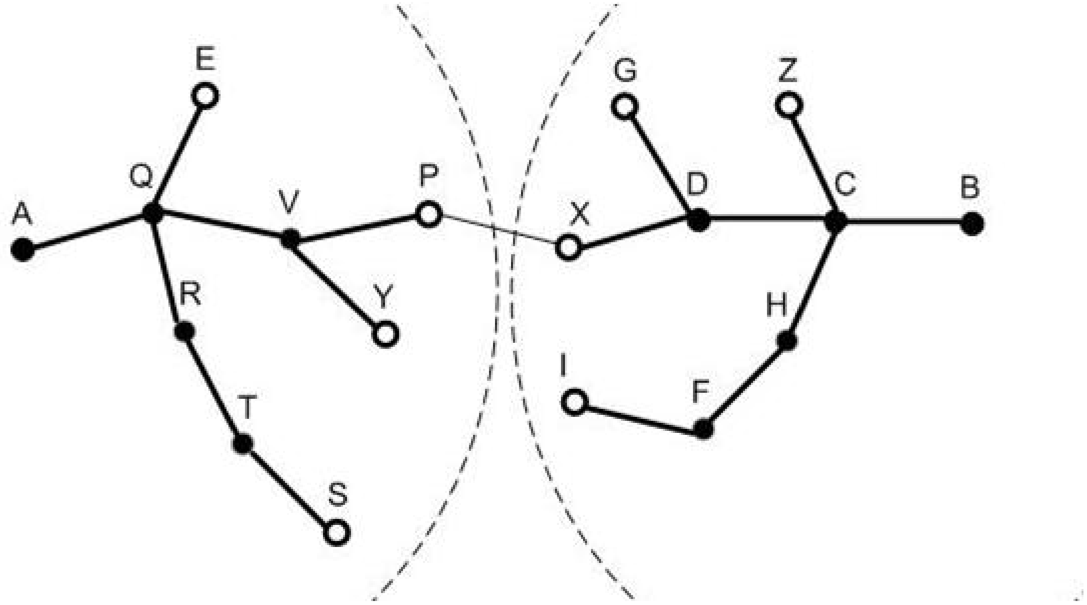

Figure 3.9 : Two dominating tree construction

3.2 MOBILITY HANDLING

In MI-CDS, each dominator tree is maintained the same way Zhou’s protocol [6] does to its CDS. This allows our protocol to take advantage of the mobility handling capability of Zhou’s protocol inside each of the dominator trees. The root of a dominator tree periodically broadcasts a heartbeat signal consisting of the initiator’s id and a sequence number to nodes under the tree so that any topology changes due to mobility can be captured and handled in a timely fashion. As described in [6], this mechanism is able to handle the following four different mobility cases:

1) The initiator leaves its dominator tree.
2) A dominator leaves its dominator tree.
3) A new node joins a dominator tree after the tree is constructed.
4) A redundant dominator switches to dominatee status without disconnecting the dominator tree.

MI-CDS will create multiple dominator trees, so there are additional mobility cases that involve more than one tree. Notice that most of these new cases can be considered as the combinations of the above four cases. For instance, if a dominator moves from one tree to another, it can be seen as the second case for the first tree plus the third case for the second tree. Consequently, most of these new cases can be handled and resolved properly and locally with no change to the protocol. The only new case that needs to be addressed is when a bridge node leaves its dominator tree.

In Figure 3.10, given a bridge node X, assume that it belongs to a dominator tree with root A, and X is used by A to connect to a neighboring dominator tree with the root B. Clearly, both A and B are initiators. If X leaves the tree, the dominator that covers X, say node P, will find it out after a couple of beacon periods. In this case, P will first try to fix the problem locally by querying its neighbors if any of them has a neighbor belonging to the initiator B. If P receives a positive response from some of its border neighbors, it instructs one of them to switch to dominator status i.e., act as the new bridge between two trees. If P does not hear back from its neighbors for a period of time, it sends a message to the initiator A, which will send a tree-wide query down to all its border nodes looking for a possible connection to the neighboring dominator tree rooted at B. The responses sent back from the border nodes are handled similarly to the messages in the tree connection phase. Afterwards, if A receives some responses from the border nodes with neighbors belonging to initiator B, it instructs the border nodes on one of the paths to turn to dominators (i.e., become bridge nodes) to connect two trees. If A does not receive any response, that implies that the dominator tree rooted at B is no longer a neighboring dominator tree for A. In this case, nothing needs to be done by node A.

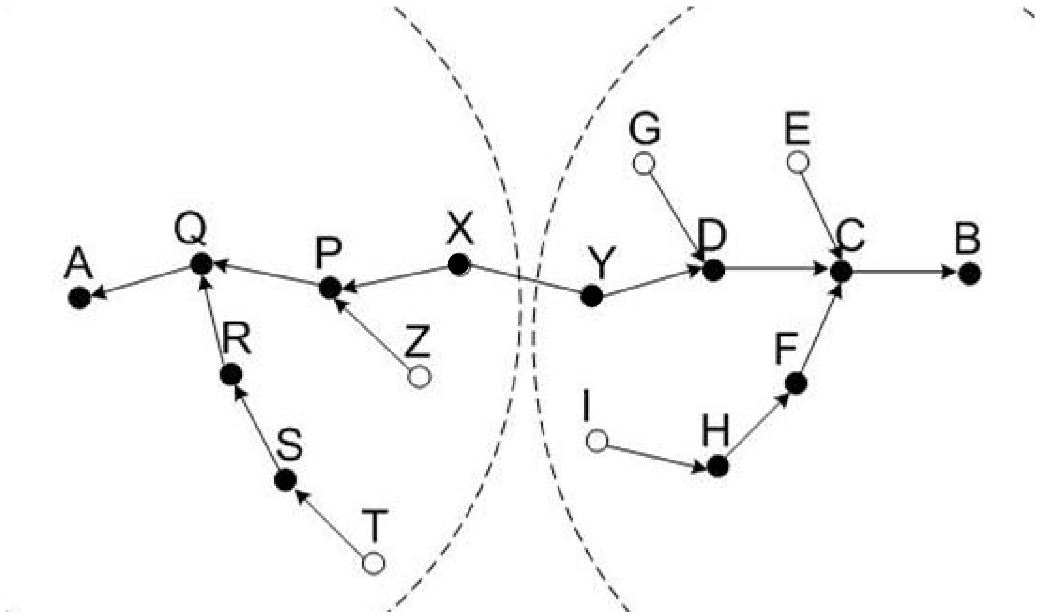

Figure 3.10 : Example of Mobility Handling

It may seem odd that the dominating set will remain connected if nothing is done after the departure of a bridge node makes two neighboring dominator trees no longer neighbors to each other. If we regard each dominator tree as a big cluster, what the tree connection phase is doing is to create an edge between each neighboring clusters. If two clusters (trees) are no longer neighbors to each other, as long as the whole network remains connected, they should be connected via the other clusters (trees). If the departure of a bridge node partitions the network into components, it will be impossible to create CDS for the network. However, MI-CDS can still maintain CDS inside each of the connected components and recover back to a single CDS when the components reconnect to each other. This feature is especially important for mobile ad hoc networks.

CHAPTER 4.

SIMULATION RESULTS

To evaluate the performance of MI-CDS, we implement our protocol in C++ along with the other CDS protocols, including Wu’s [4], Wan’s [5], and Zhou’s [6] protocols. Two different network scenarios are considered and simulated. In static scenario, the results including CDS size, number of messages to construct CDS, average traffic and convergence time are compared. In the mobile scenario, percentage of time CDS is alive, average CDS size, number of messages to maintain CDS and average traffic to maintain CDS are compared.

4.1 SIMULATION CONFIGURATIONS

Table 4.1 lists the constants used in the simulations. The network topology is randomly generated by placing nodes on a 1000 by 1000 square units of a two dimensional simulation area. The simulation area is wrapped around vertically and horizontally to avoid the edge effect. If the generated network is disconnected, it is simply discarded and a new network topology is generated with the same parameters to ensure the connectivity of the whole network. For the given simulation configurations, 100 different network topologies are generated.

Table 4.1 : Simulation Parameters

Parameters	Value	Description
l_x	1000	Width of simulation region
l_y	1000	Height of simulation region
R	150	Wireless transmission range

4.2 SIMULATION RESULTS FOR THE STATIC NETWORKS SCENARIO

In the simulation for static networks, the total number of nodes varies from 100 to 450 in a fixed network region, which corresponds to the range of the neighbors per each nodes from 5 to 30 roughly. For the evaluation of MI-CDS and comparison with other protocols, total CDS size, number of messages, average traffic, convergence time i.e. values that can judge the performance of protocol are analyzed. For Zhou's and MI-CDS, the configurations in the tree construction phase are set to be the same in [6].

Figure 4.1 shows the CDS size of different CDS protocols as a function of the number of neighbors. Among the proposed CDS protocols, Zhou's does the best while Wu's performs the worst. MI-CDS performance is very close to Zhou's. While it is proven that the size of Wan's CDS is suboptimal [5], the CDS size of MI-CDS is smaller than Wan's except when the number of neighbors is less than 15. In Wu's case, as the number of neighbors increases, the size of CDS increases. Hence, this protocol is not appropriate for dense networks. On the other hand, the other protocols are rarely affected by the variation of number of neighbors. This suggests that MI-CDS, Zhou's, and Wan's protocols are stable and scalable.

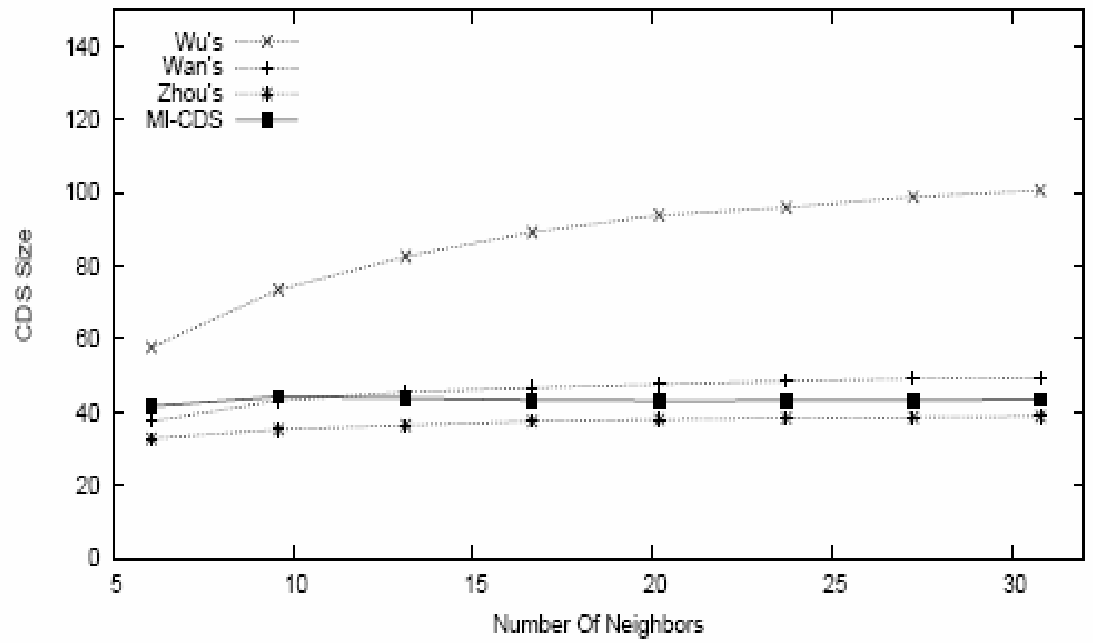

Figure 4.1 : CDS Size with variable number of neighbors

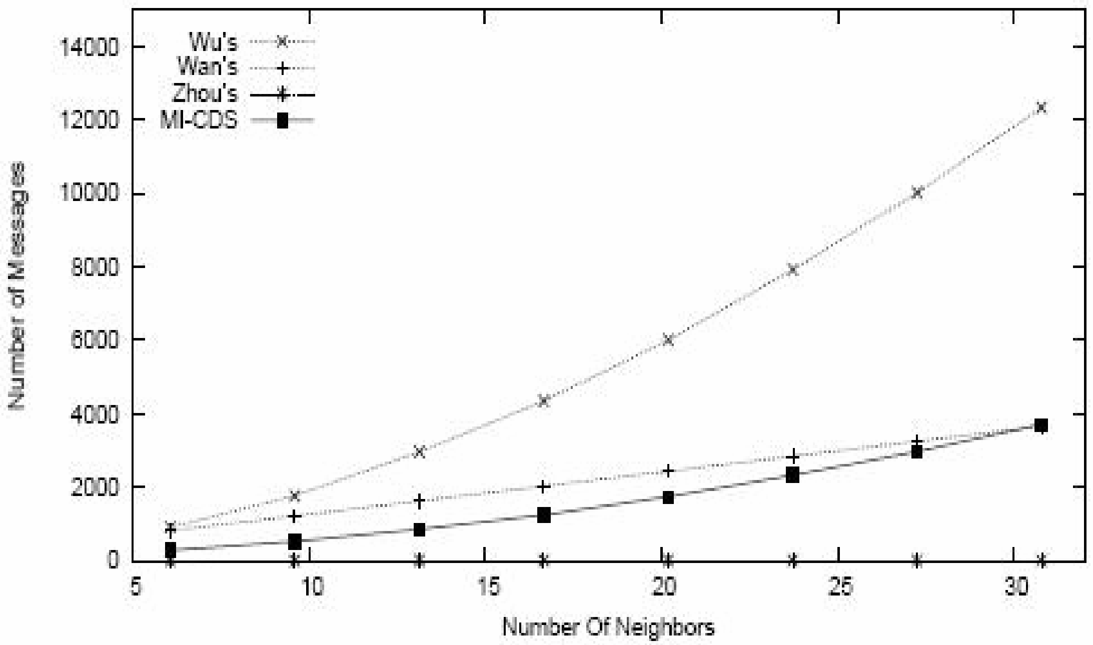

Figure 4.2 : Number of Messages with variable number of neighbors

Figure 4.2 illustrates the number of extra messages with respect to the network density. As shown in Figure 4.2 MI-CDS, Wu’s, Wan’s protocol except Zhou’s protocol generally have more extra messages according to the increase of network density. From this, it can be easily found that the higher network density is, the more number of messages they need. The reason why Zhou’s result does not introduce any extra message is that Zhou’s protocol only beacon to exchange information, as discussed in [6]. MI-CDS’s performance is somewhat better than Wan’s performance in network density ranging from 5 to 30. Particularly, when the network density ranges from 5 to 15 neighbors per node, MI-CDS reduces the number of extra messages up to 60%. In addition, MI-CDS introduces merely 30% of the number of extra messages introduced by Wu’s.

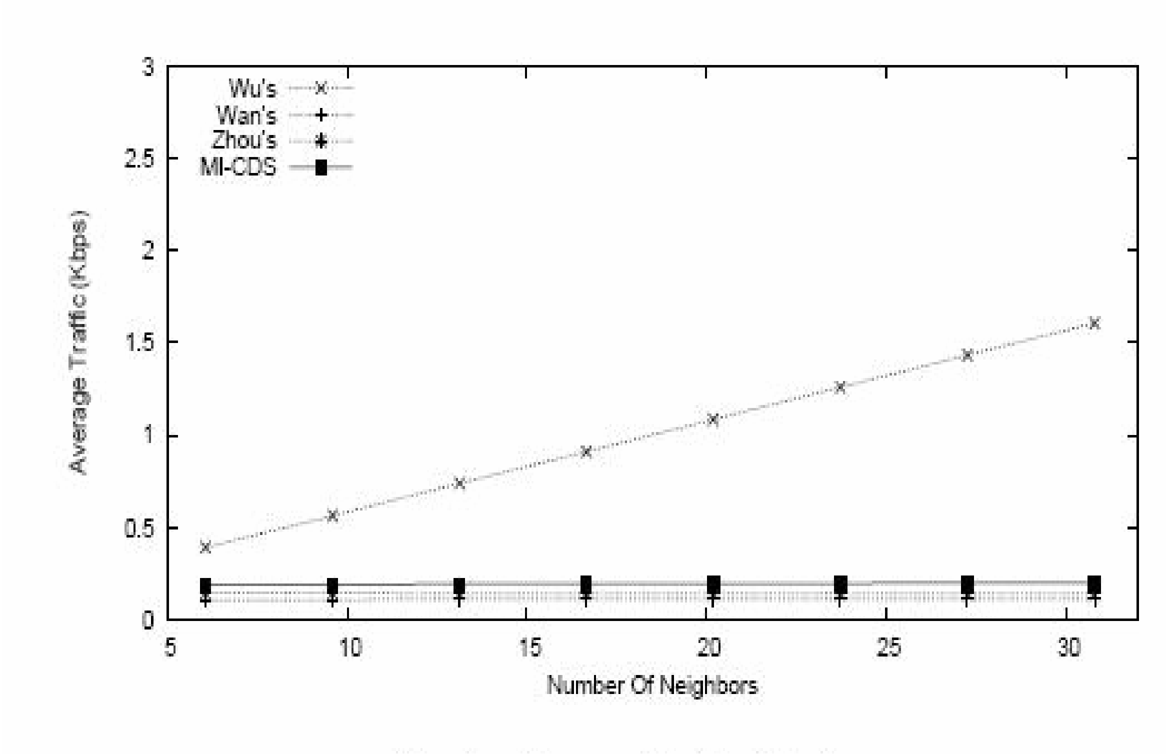

Figure 4.3 : Average Traffic (kbps) with variable number of neighbors

Figure 4.3 presents the average traffic introduced for CDS construction with a variable number of neighbors. Obviously, Wu's protocol introduces the largest traffic, and the traffic increases in proportion to the network density. This is because of the inclusion of the neighboring list in the beacon frame. For the other three protocols, the average traffic is almost the same. It is interesting that even as the number of messages increases with respect to the number of neighbors as shown in Figure 4.2, the traffic remains relatively constant to the network density. This implies the traffic is mostly dominated by the extra bits in the beacon.

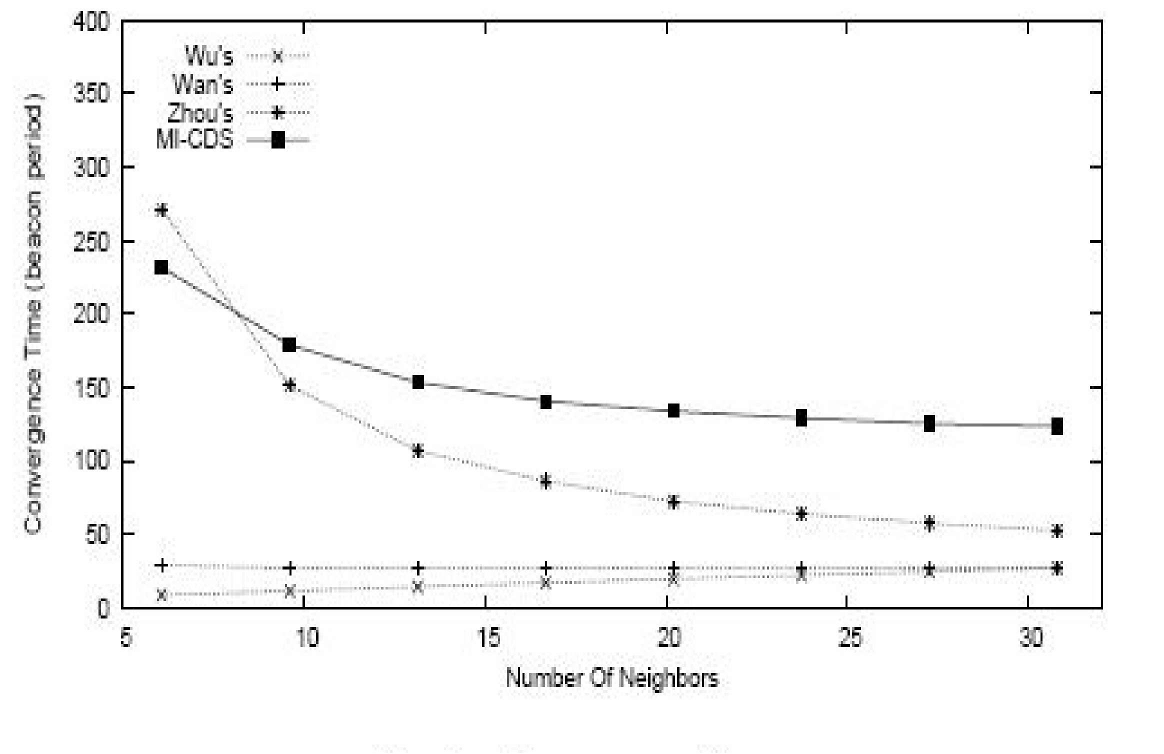

Figure 4.4 : Convergence Time with variable number of neighbors

Figure 4.4 shows the convergence time with respect to the network density. As mentioned in [6], convergence time means the period of time required for CDS construction. As shown in Figure 4.4, the convergence time of MI-CDS and Zhou's decreases quickly as the network density increases. This indicates that the convergence time is dominated by the time spent for the tree construction, in which a node in dense networks is likely to have more uncovered neighbors and will have a smaller defer timer. MI-CDS takes longer time to form CDS than Zhou's. This is due to the additional time spent in the tree connection phase.

4.3 SIMULATION RESULTS FOR THE MOBILE NETWORKS SCENARIO

In this scenario, the average network density is set to be 10 neighbors per node, and some nodes are assumed to be mobile. The percentage of the mobile nodes ranges from 20% to 50% with speed up to 5m/s. The Weighted Way Point (WWP) [8] is adopted as our mobility model. In WWP, the weight of selecting next destination and pause time for a node depend on both current location and time. The value of weights is based on empirical data carried out on the University of Southern California's campus [9]. Since Wu's and Wan's protocol do not have mobility handling mechanisms, the whole CDS is reconstructed if the CDS is found corrupted. For Zhou's and MI-CDS, the corrupted CDS is recovered according to their mobility handling procedures when the topology changes. The configurations of the heartbeat period are set to be the same as in [6]. Each simulation lasts 3600 rounds of a beacon period. In the simulation, if the network topology is partitioned into disjointed connected components, MI-CDS maintains separate CDS within each component.

To assess the performance of different protocols, five metrics are used, including the size of CDS, the number of extra messages, average traffic, the convergence time, and the percentage of time CDS is alive. For MI-CDS, the messages in the tree connection phase and query / response messages in the mobility handling are counted as the extra messages. For Zhou's protocol, all the information exchanges between nodes are done by beacons. For Wu's protocol, beacons are considered as extra messages since the size of its beacon increases in proportion to the network density and are too large compared with the standard beacon frame. For Wan's protocol, the messages exchanged in both stages are counted as extra messages. Each protocol changes the beacon frame format to include additional information. For Zhou's, node id, status, color, and dominator id are included in the beacon. MI-CDS enlarges the beacon of Zhou's to include the initiator id and the minimal id of one-hop neighbors. The beacon of Wu's protocol includes node id, status, marker, and the list of ids for one-hop neighbors. For Wan's protocol, dominator id and color are added to the beacon. The extra bit in the beacon and the size of extra messages for each protocol are counted toward the traffic (in kbps) for CDS construction. The period of time for CDS protocols to complete is defined as the convergence time. For the mobile networks scenario, the total amount time CDS is valid divided by the total simulation period is defined as the percentage of time CDS is alive.

Figure 4.5 depicts the percentage of time CDS is alive with respect to the percentage of the mobile nodes. As can be seen in Figure 4.5, MI-CDS always has the highest percentage of CDS alive time compared to the other CDS protocols. The more mobile nodes the network has, the better performance MI-CDS achieves in terms of the CDS alive time percentage compared with the other protocols. For instance, in the case of 50% of mobile nodes, CDS built by MI-CDS is alive eight times longer than Wu and Wan's protocol, and more than twice as long than Zhou's. Although smaller CDS size is generally more vulnerable to topology changes, MI-CDS shows excellent mobility adaptation compared with the other CDS protocols.

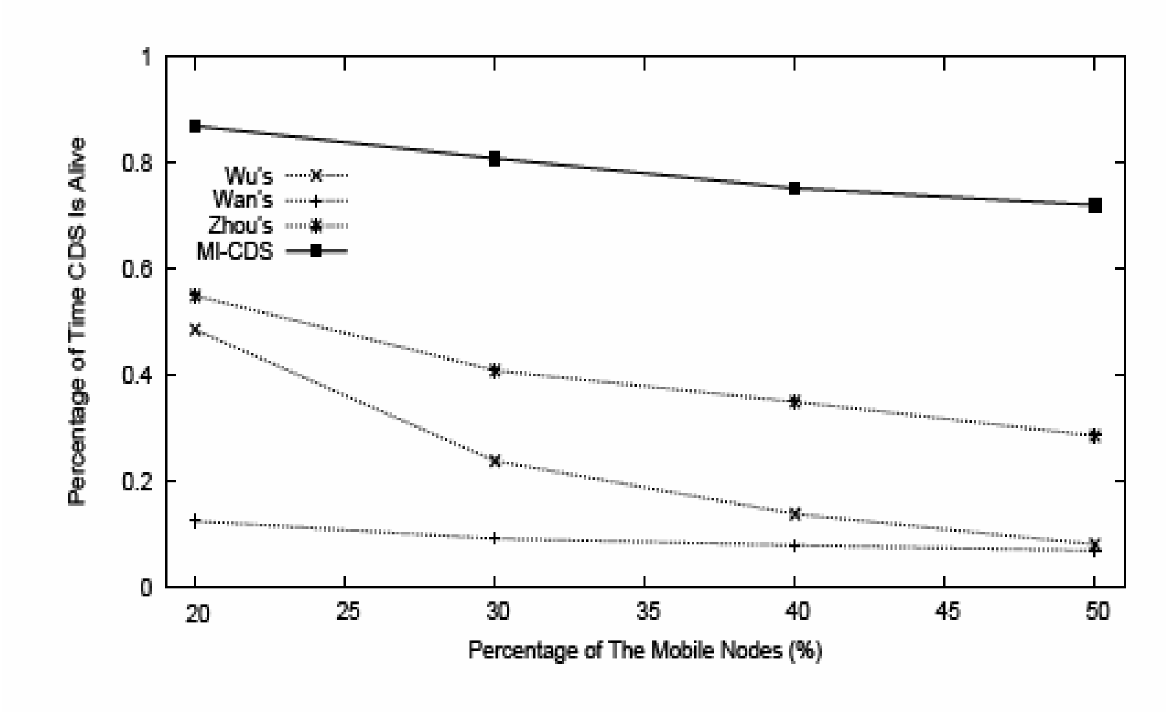

Figure 4.5 : Percentage of Time CDS is Alive

Figure 4.6 shows the average CDS size with respect to the percentage of the mobile nodes. As illustrated in Figure 4.6, Zhou’s consistently produces the smallest CDS and Wu’s consistently produces the largest CDS. The average CDS size of MI-CDS is only few nodes higher than Wan’s.

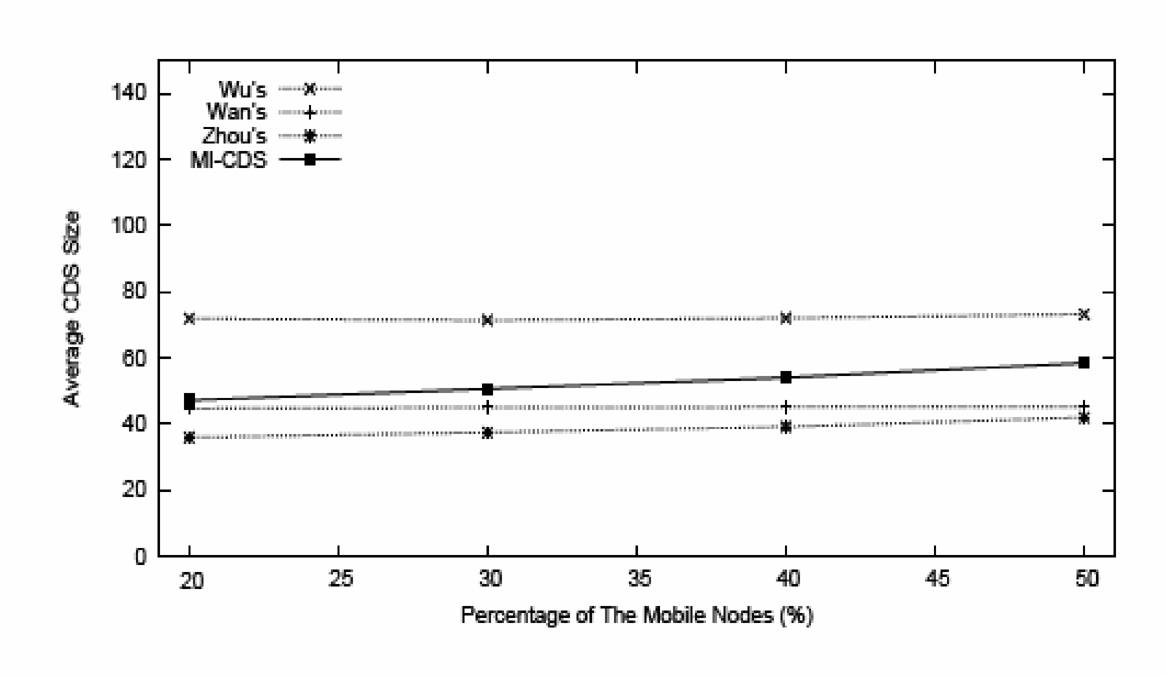

Figure 4.6 : Average CDS Size

Figure 4.7 presents the number of extra messages to maintain CDS with respect to the percentage of the mobile nodes. As illustrated in Figure 4.7, MI-CDS requires a low number of extra messages to maintain CDS. The number of messages is only 20% of that of Wan’s and 8% of Wu’s. This demonstrates that MI-CDS can handle topology changes efficiently with only a small number of messages.

Figure 4.8 shows the average traffic required at each node to maintain CDS with respect to the percentage of the mobile nodes. As can be seen in Figure 4.8, the average traffic of MI-CDS is at least 50% lower than that of Wu’s. Compared with Wan’s and Zhou’s, MI-CDS has slightly higher traffic, but the difference is not significant. This is because the beacon in MI-CDS is slightly larger than that of Wan’s and Zhou’s. As we have pointed out in simulation results for the static network scenarios, the traffic is mostly dominated by the extra bits in the beacon frame rather than the extra messages. This again is validated by Figure 4.7 and Figure 4.8.

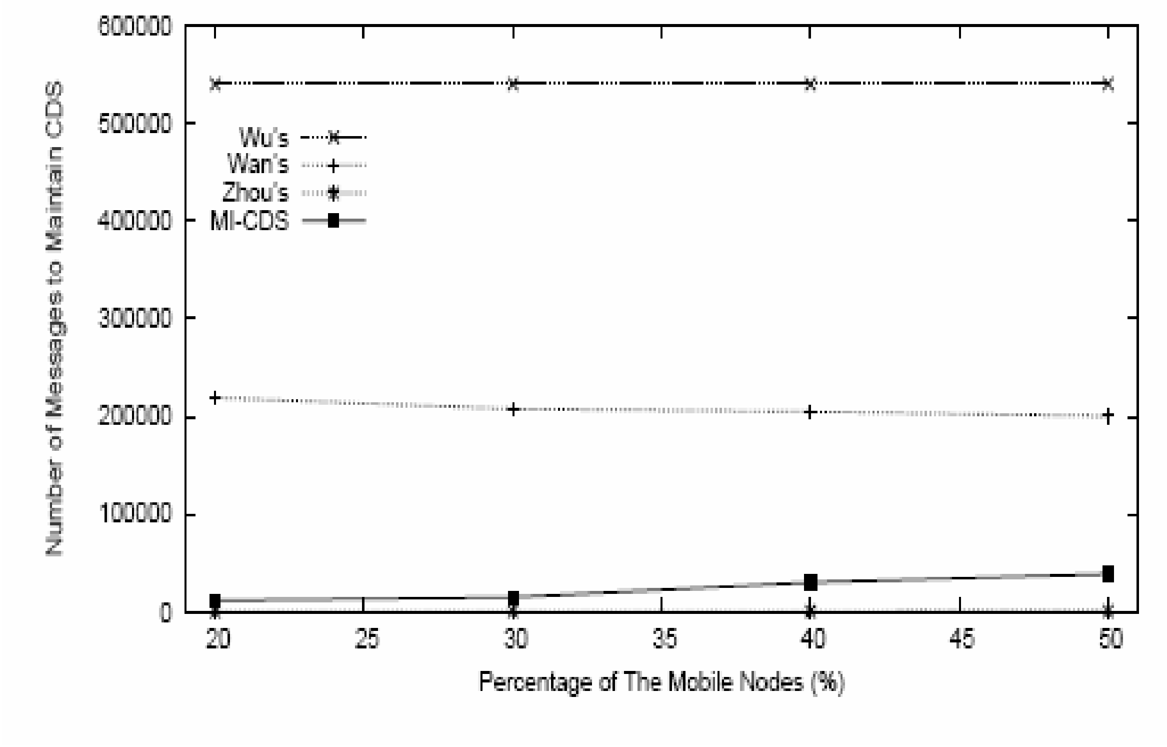

Figure 4.7 : Number of Messages to Maintain CDS

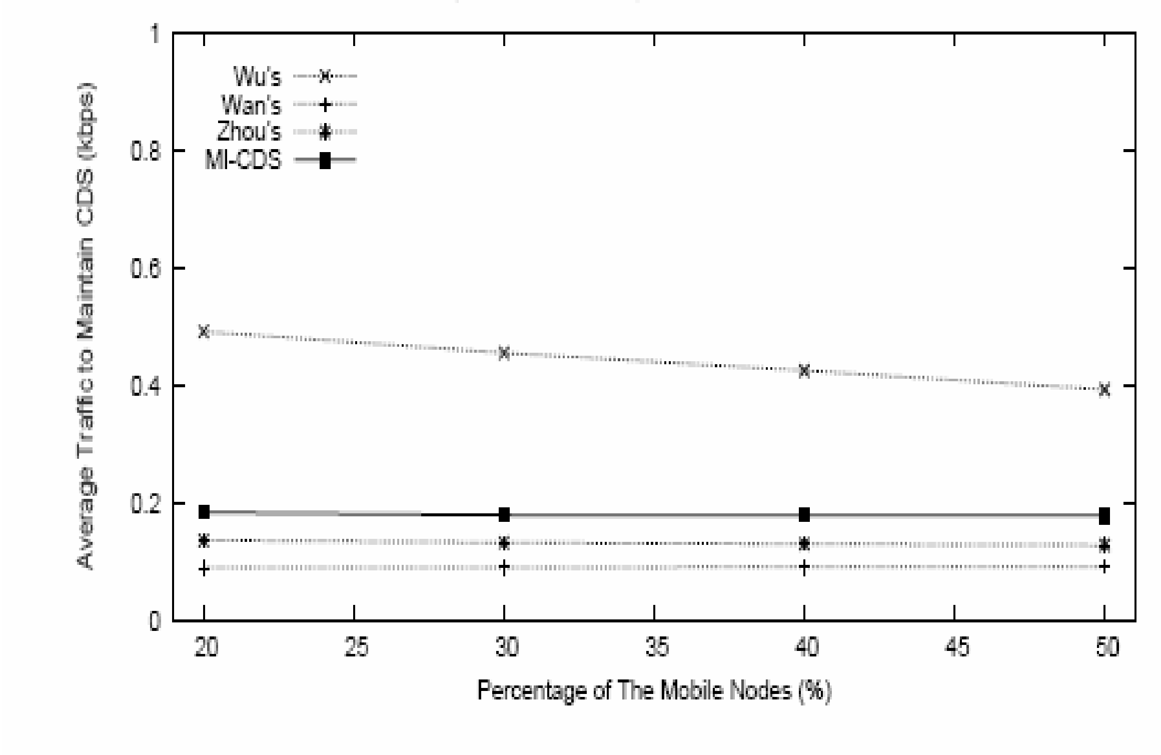

Figure 4.8 : Average Traffic to Maintain CDS

CHAPTER 5.
ANALYSIS

In this chapter, the analysis of the convergence time and the number of messages MI-CDS requires during CDS construction is presented with our analytical model.

5.1 CONVERGENCE TIME

In the tree construction phase of MI-CDS, the defer timer, ΔT is set to be $\Delta T = T_{\max} \cdot \frac{1}{(numberofunmarkedneighbors)^{\alpha}}$, where $T_{\max}$ is the maximal defer time period and *numberofunmarkedneighbors* is the number of uncovered neighbors. Since Zhou's protocol is used in the tree construction phase, before we discuss the convergence time of MI-CDS, let us first consider the convergence time of Zhou's protocol. As Figure 4.4 shows in the previous chapter, the convergence time of Zhou's decreases in proportion to the increase of the number of neighbors inversely. This phenomenon can be explained by the fact that the convergence time of Zhou's is the product of the number of hops from the initiator to the edge of the tree and the average defer time. To formulate this, let us denote the width and height of the simulation region as l_x and l_y respectively. The number of hops, denoted by *h*, is the distance from the initiator to the edge of tree divided by the average distance between two nodes. Assuming that length of

l_x and l_y are the same and the area of the region $S = l_x \times l_y$, then the average value of h can be computed by Equation 5.1.

$$h = \frac{\sqrt{l_x^2 + l_y^2}}{\frac{2}{3}r} = \frac{3\sqrt{2S}}{2r} \qquad (5.1)$$

The average defer time can be the maximal defer time divided by the average number of uncovered nodes in that the covered node calculates the ΔT when it has uncovered neighbors. Assuming that the maximal defer timer is denoted by T_{max}, ΔT is defined by Equation 5.2.

$$\Delta T = T_{max} \cdot \frac{1}{(number\ of\ unmarked\ neighbors)} \qquad (5.2)$$

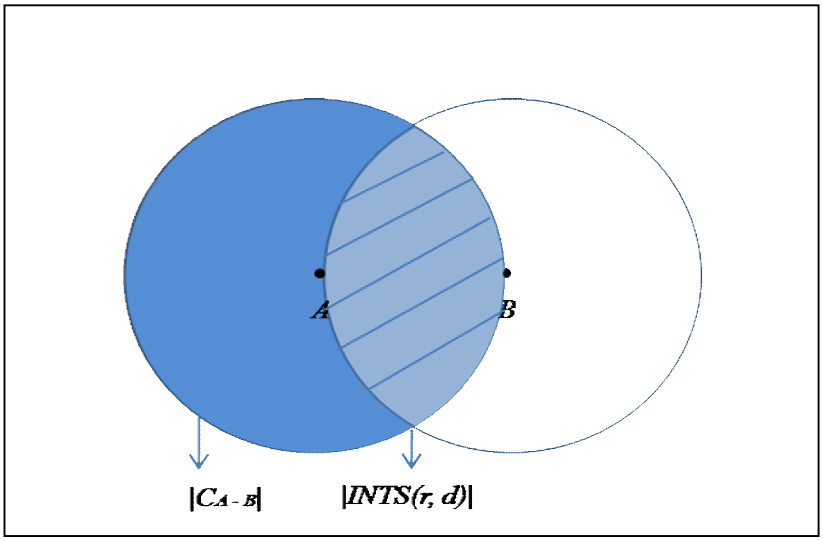

Figure 9 : Coverage area of node *A* and *B*

To formulate the average defer time, let C_A and C_B be the coverage area of two neighboring nodes *A* and *B*, and *d* be the distance between them as we see in Figure 5.1.

The additional area that node *B* forwards a message from node *A* can be $|\, C_{A-B}|$, it can be rewritten by the following formula.

$$|\, C_{A-B}| = \pi r^2 - |INTS(r,d)| \tag{5.3}$$

, where $INTS(r,d)$ is the intersection area of the two circle with radius *r* and their centers separated by *d*.

$$INTS(r,d) = 4\int_{d/2}^{r} \sqrt{r^2 - x^2}\,\mathrm{d}x \tag{5.4}$$

Then, the average additional coverage area is

$$\int_0^r \frac{2\pi x\left(\pi r^2 - INTS(r,x)\right)}{\pi r^2}\mathrm{d}x \approx 0.41\pi r^2 \tag{5.5}$$

Therefore, the average number of uncovered nodes is *0.41*$\boldsymbol{\Delta}$. Lastly, the convergence time of Zhou's, $CONVT_{Zhou's}$ is approximately adjusted by Equation 5.6.

$$CONVT_{Zhou's} = h \cdot \frac{T_{max}}{0.41\Delta} \tag{5.6}$$

Now let us turn to MI-CDS. The duration of the tree construction phase in MI-CDS can be calculated in a similar fashion. In the case of multi-initiator, the number of hops from an initiator to the edge of its tree, h_{mi}, is computed by Equation 5.7.

$$h_{mi} = \frac{r/\bar{d}}{n_{init}\pi r^2/S}$$

$$= \frac{3}{2} \cdot \frac{S}{n_{init} \pi r^2} \qquad (5.7)$$

For the tree connection phase in MI-CDS, a border node without any uncovered neighbor will wait T_{max} before it sends a message to its initiator. Hence, the time required for the tree connection phase is bounded by $2h_{mi} + 1$. The convergence time of MI-CDS is the total time spent on the tree construction phase and the tree connection phase. Finally, the convergence time for MI-CDS, $CONVT_{MI-CDS}$, is computed by Equation 5.8.

$$CONVT_{MI-CDS} = (h_{mi} - 1) \cdot \frac{T_{max}}{0.41\Delta} + 2h_{mi} + T_{max} + 1 \qquad (5.8)$$

Figure 5.1 depicts the values calculated from our analytical model and the results of our computer simulation. As seen in this figure, our analytical model provides very accurate estimations in terms of the convergence time (without counting the time spent for initiator election).

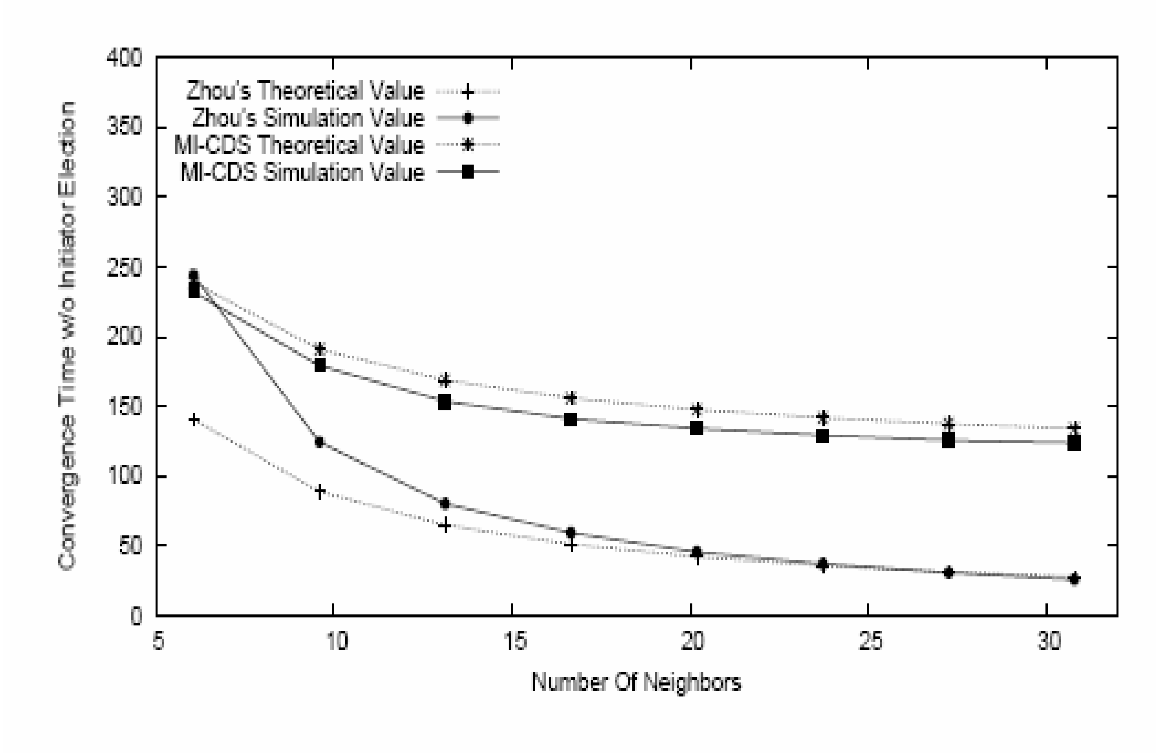

Figure 5.10 : The Convergence Time w/o Initiator Election

5.2 NUMBER OF MESSAGES

MI-CDS does not introduce extra messages except in the tree connection phase. In the tree connection phase, all nodes except initiators forward messages from their children as many times as the number of neighboring trees to their initiator. Thus, the number of message required for MI-CDS to construct CDS is computed by Equation 5.9.

$$0.41\Delta \cdot \frac{n_{init} - 1}{n_{init}} \cdot n \qquad (5.9)$$

Figure 5.2 depicts the values calculated from our analytical model and the results of our computer simulation.

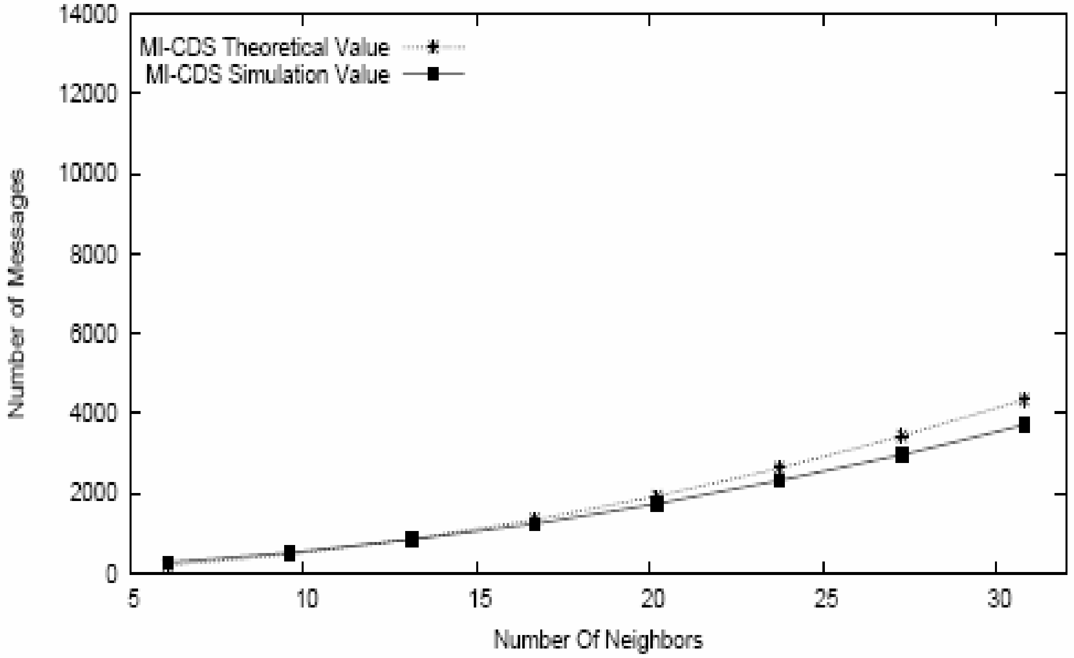

Figure 5.11 : The Number of Messages

CHAPTER 6

CONCLUSION

The CDS protocols proposed in the past either lack the ability to handle nodal mobility, or recover slowly when the constructed CDS becomes corrupted. In this thesis, the Multi-Initiator Connected Dominating Set protocol (MI-CDS) is proposed that can construct and maintain CDS efficiently. Instead of relying on one single initiator, MI-CDS elects a small number of initiators to generate CDS. This allows MI-CDS to fix the corrupted CDS quickly caused by different types of nodal mobility. The simulation results demonstrate that MI-CDS consistently generates CDS of competitive size with very low communication overhead. In case of mobile networks, the simulation results confirm that MI-CDS allows CDS to be available for the highest percentage of the time compared with the other CDS protocols. Analytical model is built to estimate the time of convergence and the number of messages for MI-CDS.

The existing mobile ad hoc networks CDS protocols are reviewed in Chapter 2. The MI-CDS protocol is presented in detail and how it handles nodal mobility is described in Chapter 3. The simulation results are shown in Chapter 4. The analytical models are demonstrated and validated in Chapter 5.

In the future, we would like to consider virtual backbone construction for multi-hop wireless mesh network, where each node may be equipped with multiple radio interfaces. The definition of the connected dominating set may need to be extended so that it can be applied to multi-radio wireless mesh networks. In addition, the performance metrics of a virtual backbone for multi-radio wireless mesh network should include throughput, which makes the CDS protocol performance more difficult to assess since the throughput is also dependent to the routing protocol.

BIBLIOGRAPHY

[1] S. Corson, and J. Macker, "Mobile Ad Hoc Networking (MANET) : Routing Protocol Performance Issues and Evaluation Consideration," RFC 2501, Jan. 1999.

[2] J. Cartigny, D. Simplot, and I. Stojemenovic, “Localized Minimum-energy Broadcasting in Ad Hoc Networks,” Proc. of IEEE Conference on Computer Communications (INFOCOM), pp. 2210 - 2217, Mar. 2003.

[3] D. S. Hochbaum, “Approximation Algorithms for NP-Hard Problems,” PWA Publishing Company, Boston, MA, 1995.

[4] J. Wu and H. Li, “A dominating-set-based routing scheme in ad hoc wireless networks,” Telecommunication Systems Journal, vol.18, pp. 13 - 36, 2001.

[5] P. J. Wan, K. M. Alzoubi, and O. Frieder, “Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks,” Proc. of IEEE Conference on Computer Communications (INFOCOM), pp. 141 - 149, Apr. 2004.

[6] D. Zhou, M. –T. Sun, and T. Lai, “A Timer-based Protocol for Connected Dominating Set Construction in IEEE 802.11 Multihop Mobile Ad Hoc Networks,” Proc.

of International Symposium on Applications and the Internet (SAINT), pp. 2 - 8, Jan. 2005.

[7] IEEE 802.11b Standard, http://standards.ieee.org/getieee802/download/802.11b-1999.pdf

[8] W. Hsu, K. Merchant, H. Shu, C. Hsu, and A. Helmy, “Weighted Waypoint Mobility Model and its Impact on Ad Hoc Networks,” ACM SIGMOBILE Mobile Computing and Communications Review (MC2R), pp. 59 - 63, Jan. 2005.

[9] Mobilab:Community-wide Library of Mobility and Wireless Networks Measurements nile.usc.edu/MobiLib/

[10] N. Nanuvala, “An Enhanced Algorithm to Find Dominating Set Nodes in Ad Hoc Wireless Networks,” Master of Science thesis in the College of Arts and Science, Georgia State University, 2006.

[11] D. Zhou, “Clock Synchronization and Dominating Set Construction in Ad Hoc Wireless Network,” PhD thesis, Ohio State University, 2005.

[12] J. Manner and M. Kojo, “Mobile Related Terminology,” RFC 3753, Jun. 2004.

[13] J. Blum, M. Ding, A. Thaeler, and X. Cheng, “Connected dominating set in sensor networks and MANETs,” in *Handbook of Combinatorial Optimization*, D.-Z. Du and P. Pardalos, Eds. Norwell, MA: Kluwer, 2004.

[14] D. P. Agrawal and Q-A. Zeng, “Introduction to Wireless and Mobile Systems,” Brooks/Cole, 2003.

[15] N. Aschenbruck, M. Frank, P. Martini, and J. Tolle, “Human mobility in MANET disaster area simulation a realistic approach,” Proc. of 29th Annual IEEE International Conference on Local Computer Networks (LCN’04), 2004.

[16] Pullin, A.J. and Pattinson, C, “A Realistic Battlefield Model for the Evaluation of MANET,” Proc. of IEEE Conference on Computer Communications, 2008.

[17] S. Butenko, X. Cheng, D.-Z. Du and P. Pardalos, “On the construction of virtual backbone for ad hoc wireless networks,” in 2nd Conference on Cooperative Control and Optimization, 2001.

[18] J.W.A.F. Dai, “A Distributed Formation of a Virtual Backbone in MANETs using Adjustable Transmission Ranges,” International Conference on Distributed Computing Systems, Washington, DC, USA. pp. 372 - 379. 2004.

[19] B. Kim, J. Yang, D. Zhou, and M.-T. Sun, “Energy-aware connected dominating set construction in mobile ad hoc networks,” ICCCN, 2005.

[20] S. Guha and S. Khuller. “Approximation algorithms for connected dominating sets. Algorithmica,” 20(4) pp. 374 – 387, Apr. 1998.

[21] B. Das, R. Sivakumar, and V. Bhargavan. “Routing in ad hoc networks using a spine,” Proc. of 6th IEEE International Conference on Computers Communications and Networks (IC3N ’97), pp. 1 – 20, Sep. 1997.

[22] K. Sakai, F. Shen, K. M. Kim, M. –T. Sun, and H. Okada, “Multi-Initiator Connected Dominating Set Construction for Mobile Ad Hoc Networks”, ICC, 2008.

APPENDIX

1. OUTPUT DATA FOR THE STATIC NETWORKS SCENARIO

Protocol	Number of Neighbors	CDS size	Number of messages	Average traffic	Convergence time
MI-CDS protocol	6.07	41.59	279.20	0.19	232.02
	9.60	44.23	526.09	0.19	179.19
	13.13	43.87	851.30	0.19	153.65
	16.66	43.28	1244.23	0.20	140.96
	20.20	43.09	1741.30	0.20	134.13
	23.73	43.15	2326.50	0.20	129.33
	27.26	43.23	2968.52	0.20	125.73
	30.79	43.43	3716.37	0.20	123.83
Zhou's protocol	6.07	32.74	0.00	0.14	244.08
	9.60	35.19	0.00	0.14	124.49
	13.13	36.39	0.00	0.14	80.33
	16.66	37.51	0.00	0.14	59.42
	20.20	37.95	0.00	0.14	45.73
	23.73	38.37	0.00	0.14	37.32
	27.26	38.59	0.00	0.14	30.87
	30.79	38.81	0.00	0.14	25.92

Wan's protocol	6.07	37.47	815.26	0.11	29.36
	9.60	43.18	1219.25	0.11	27.44
	13.13	45.49	1621.35	0.11	27.09
	16.66	46.68	2028.83	0.11	27.00
	20.20	47.61	2435.28	0.11	27.02
	23.73	48.47	2840.65	0.11	27.10
	27.26	49.23	3252.33	0.11	27.18
	30.79	49.47	3663.30	0.11	27.07
Wu's protocol	6.07	57.73	910.00	0.39	9.10
	9.60	73.54	1767.00	0.56	11.78
	13.13	82.61	2960.00	0.74	14.80
	16.66	89.25	4347.50	0.91	17.39
	20.20	93.90	6006.00	1.09	20.02
	23.73	95.99	7924.00	1.26	22.64
	27.26	98.95	10024.00	1.43	25.06
	30.79	100.66	12343.50	1.61	27.43

2. OUTPUT DATA FOR THE MOBILE NETWORKS SCENARIO

Protocol	Percentage of the mobile nodes	Percentage of time CDS is alive	Average CDS size	Number of messages to maintain CDS	Average traffic to maintain CDS
MI-CDS protocol	20	0.87	47.21	11929.67	0.18
	30	0.81	50.68	15181.89	0.18
	40	0.75	54.10	30279.82	0.18
	50	0.72	58.47	38984.90	0.18
Zhou's protocol	20	0.55	35.96	205.36	0.14
	30	0.41	37.45	669.85	0.13
	40	0.35	39.19	550.05	0.13
	50	0.29	42.01	804.08	0.13
Wan's protocol	20	0.12	44.77	219227.53	0.09
	30	0.09	45.05	207656.63	0.09
	40	0.08	45.13	204978.96	0.09
	50	0.07	45.21	200163.84	0.09
Wu's protocol	20	0.49	71.92	540004.50	0.49
	30	0.24	71.43	540012.00	0.45
	40	0.14	72.12	540012.00	0.43
	50	0.08	73.24	540019.50	0.39

3. OUTPUT DATA FOR THE CONVERGENCE TIME W/O INITIATOR ELECTION OF THEORETICAL AND SIMULATION VALUE

Number of neighbors	Zhou's protocol		MI-CDS protocol	
	Theoretical value	Simulation value	Theoretical value	Simulation value
6.07	140.75	244.08	238.78	232.02
9.60	88.95	124.49	190.82	179.19
13.13	65.02	80.33	168.67	153.65
16.66	51.23	59.42	155.91	140.96
20.20	42.27	45.73	147.61	134.13
23.73	35.98	37.32	141.78	129.33
27.26	31.32	30.87	137.47	125.73
30.79	27.72	25.92	134.14	123.83

4. OUTPUT DATA FOR THE NUMBER OF MESSAGES OF MI-CDS THEORETICAL AND SIMULATION VALUE

Number of neighbors	MI-CDS Theoretical value	MI-CDS Simulation value
6.07	206.66	279.2
9.60	479.3	526.09
13.13	860.9	851.3
16.66	1346.06	1244.23
20.20	1934.56	1741.3
23.73	2634.56	2326.5
27.26	3435.77	2968.52
30.79	4347.8	3716.37