Connected Dominating Sets in Wireless Networks with Different Transmission Ranges

My T. Thai Feng Wang Dan Liu Shiwei Zhu Ding-Zhu Du
Dept. of Computer Science & Enginering
University of Minnesota
Minneapolis, MN 55455
{mythai, fwang, danliu, zhu, dzd}@cs.umn.edu

***Abstract*— Since there is no fixed infrastructure or centralized management in wireless ad hoc networks, a Connected Dominating Set (CDS) has been proposed as the virtual backbone. The CDS of a graph representing a network has a significant impact on an efficient design of routing protocols in wireless networks. This problem has been studied extensively in Unit Disk Graphs (UDG), in which each node has the same transmission range. However, in practice, the transmission ranges of all nodes are not necessary equal. In this paper, we model a network as a disk graph and introduce the CDS problem in disk graphs. We present three constant approximation algorithms to obtain a minimum CDS of a given network. These algorithms can be implemented as distributed algorithms. Furthermore, we show the size relationship between a maximal independent set and a CDS as well as the bound of the maximum number of independent neighbors of a node in disk graphs. The theoretical analysis and simulation results are also presented to verify our approaches.**

Keywords: Connected Dominating Set, Independent Set, Disk Graph, Wireless Network, Virtual Backbone

I. Introduction

In wireless ad hoc networks, there is no fixed or pre-defined infrastructure. Nodes in wireless networks communicate via a shared medium, either through a single hop or multihops. Although there is no physical backbone infrastructure, a virtual backbone can be formed by constructing a Connected Dominating Set (CDS). Given an *undirected* graph $G = (V, E)$, a subset $V' \subseteq V$ is a CDS of G if for each node $u \in V$, u is either in V' or there exists a node $v \in V'$ such that $uv \in E$ and the subgraph induced by V', i.e., $G(V')$, is connected. The nodes in the CDS are called *dominators*, other nodes are called *dominatees*. With the help of the CDS, routing is easier and can adapt quickly to network topology changes. To reduce the traffic during communication and simplify the connectivity management, it is desirable to construct a Minimum CDS (MCDS).

The CDS problem has been studied intensively in Unit Disk Graph (UDG), in which each node has the same transmission range. The MCDS problem in UDG has been shown to be NP-hard [1]. To build a CDS, most of current algorithms first find a Maximal Independent Set (MIS) I of G and then connect all nodes in I to have a CDS. The independent set I is a subset of V such that for any two nodes $u, v \in I$, $uv \notin E$. In other words, the nodes in I are pairwise nonadjacent. A maximal independent set is an independent set such that no more nodes can be added to remain the non-adjacency property. The most relevant related work using this scheme are in [2], [3]. In [2], Wan *et al.* proposed the first distributed algorithm with the performance ratio of 8. Later, Li *et al.* proposed a better algorithm with the performance ratio of $(4.8 + \ln 5)$ by constructing a Steiner tree when connecting all nodes in I [3].

However, in practice, the transmission ranges of all nodes are not necessary equal. In this case, a wireless ad hoc network can be modeled using a directed graph $G = (V, E)$. The nodes in V are located in a Euclidean plane and each node $v_i \in V$ has a transmission range $r_i \in [r_{min}, r_{max}]$. A directed edge $(v_i, v_j) \in E$ if and only if $d(v_i, v_j) \leq r_i$ where $d(v_i, v_j)$ denotes the Euclidean distance between v_i and v_j. Such graphs are called *disk graphs*. An edge (v_i, v_j) is bidirectional if both (v_i, v_j) and (v_j, v_i) are in E, i.e., $d(v_i, v_j) \leq min\{r_i, r_j\}$. In other words, the edge (v_i, v_j) is bidirectional if v_i is in the disk D_j centered at v_j with radius r_j and v_j is in the disk D_i centered at v_i with radius r_i. In this paper, we only study the CDS problem in disk graphs where all the edges in the network are bidirectional, called Disk Graphs with Bidirectional links (DGB). In this case, G is undirected. Figure 1 gives an example of DGB representing a network. In Figure 1, the dotted circles represent the transmission ranges and the black nodes represent a CDS.

The MCDS problem in DGB is NP-hard [14] since the MCDS problem in UDG is NP-hard and UDG is

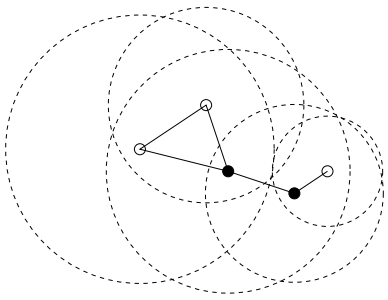


Fig. 1. A Disk Graph with Bidirectional Links

a special case of DGB. In this paper, we present three constant approximation algorithms for computing a minimum CDS in DGB. We first introduce their centralized versions and later show how to implement them as distributed algorithms. We also analyze the size relationship between an MIS and a CDS in DGB. Furthermore, we show the upper bound of the number of independent neighbors of any node in DGB. These analysis can help us to study the CDS problem in a general disk graph, where both unidirectional and bidirectional links are considered.

The remainder of this paper is structured as follows. Section II describes the related research work on the CDS problem, mainly focuses on UDG. The size relationship between an MIS and a CDS in DGB is shown in section III. The three algorithms and their performance analysis are discussed in section IV. Section V presents the performance comparisons of our three algorithms through simulation results. The distributed implementations are illustrated in section VI and section VII ends the paper with conclusions and some future work.

II. Related Work

The CDS problem in wireless ad hoc networks has been studied extensively. Algorithms that construct a CDS can be divided into two categories: *centralized* algorithms and *decentralized* algorithms.

The *centralized* algorithms in general yield a smaller CDS with a better performance ratio than that of decentralized algorithms. In [6], Guha and Khuller first proposed two polynomial time algorithms to construct a CDS in a general graph G. These algorithm are greedy and centralized. The first one has the approximation ratio of $2(H(\Delta)+1)$ where Δ is the maximum degree of G and H is a harmonic function. The idea of this algorithm is to build a spanning tree T rooted at the node with maximum degree and grow T until all nodes are added to T. The non-leaf nodes in T form a CDS. In particular, all nodes in a given network are white initially. The greedy function that the algorithm uses to add nodes into T is the number of the white neighbors of each node or a pair of nodes. The one with the largest such number is marked black and its neighbors are marked grey. These nodes (black and grey nodes) are then added into T. The algorithm stops when no white node exists in G. The second algorithm is an improvement of the first one. This algorithm consists of two phases. The first phase is to construct a dominating set and the second phase is to connect the dominating set using a Steinter tree algorithm. With such improvement, the second algorithm has the performance factor of $H(\Delta)+2$. These algorithms later were studied and implemented by Das *et al.* [11]-[13]. In [7], Ruan *et al.* introduced another centralized and greedy algorithm of which the approximation ratio is $(2+\ln\Delta)$.

The decentralized algorithms can be further divided into two categories: *distributed* algorithms and *localized* algorithms. In distributed algorithms, the decision process is decentralized. In the localized algorithms, the decision process is not only distributed but also requires only a constant number of communication rounds. Most of the distributed algorithms find a Maximal Independent Set (MIS) and connect this set. Note that in an undirected graph, an MIS is also a Dominating Set (DS). In [2], [15], [16], the authors proposed a distributed algorithm for a CDS problem in UDG. This algorithm consists two phases and has the constant approximation ratio of 8. The algorithm first constructs a spanning tree. Then each node in a tree is examined to find an MIS for the first phase. All nodes in an MIS are colored black. At the second phase, more nodes are added (color blue) to connect the black nodes. Later, Cardei *et al.* presented another 2-phase distributed algorithm for a CDS in UDG. This algorithm has the same performance ratio as the previous one. However, the improvement over [2] is the message complexity. The root does not need to wait for the COMPLETE message from the furthest nodes. Recently, Li *et al.* proposed another distributed algorithm with a better approximation ratio, which is $(4.8+\ln 5)$ [3]. This algorithm also has two phases. At the first phase, an MIS is found. At the second phase, a Steiner tree algorithm is used to connect the MIS.

For the localized algorithms, Wu and Li [8] proposed a simple algorithm that can quickly determine a CDS based on the connectivity information within the 2-hops neighbors. This approach uses a marking process. In particular, each node is marked true if it has two unconnected neighbors. All the marked nodes form a CDS. The authors also introduced some dominant pruning rules to reduce the size of the CDS. In [2], the authors showed that the performance ratio of [8] is within a factor of

$O(n)$ where n is the number of nodes in a network.

In [10], Alzoubi *et al.* proposed another localized 2-phase algorithms with the performance ratio of 192. At the first phase, an MIS is constructed using the 2-hops neighbors information. Specifically, once a node knows that it has the smallest ID within its neighbors, it becomes a dominator. At the second phase, the dominators are responsible for identifying a path to connect the MIS. In [9], Li *et al.* proposed another localized algorithm with the performance ratio of 172. This localized algorithm has only 1 phase. A node marks itself as a dominator if it can cover the most white nodes compared to its 2-hops neighbors.

Most of the constant approximation algorithms are for the CDS problem in UDG. However, in practice, the communication ranges of nodes in a network are not necessary equal. Such a network can be modeled as a disk graph. In this paper, we present three constant approximation algorithms for the CDS problem in DGB. The main approach is to construct an MIS and then connect them. Hence we first need to analyze the size relationship between a CDS and an MIS, which is shown in next section.

III. The Size Relationship Between a CDS and a Maximal Independent Set

In this section, we show the size relationship between any maximal independent set and a CDS of a given DGB. Denote OPT as an optimal CDS and opt as the size of OPT, we have:

Fact 1: Given 3 nodes x, y, and z such that $d(x,y) \leq d(x,z)$ and $d(y,z) \leq d(x,y)$, then y and z are adjacent.

Proof: The disk at y has radius at least $d(x,y)$ and the disk at z has radius at least $d(x,z)$. Therefore, both disks have radius at least $d(y,z)$. Hence, y and z are adjacent.

□

Lemma 1: Let $N_{ID}(u)$ denote the independent neighbors of node u. In a DGB, the size of $N_{ID}(u)$ is bounded by:

$$|N_{ID}(u)| \leq \begin{cases} 5 & \text{if } k = 1 \\ 10\lfloor \frac{\ln k}{\ln(2\cos(\pi/5))} \rfloor & \text{otherwise} \end{cases}$$

where $k = \dfrac{r_{max}}{r_{min}}$

Proof: When $k = 1$, a DGB is a UDG. Thus the lemma holds. When $k > 1$, consider a node x and all nodes that are adjacent to x. Without loss of generality, assume that the disk at x has radius 1. Then a node y that is adjacent to x has radius at least $d(x,y)$ and at most k. Thus, all nodes that are adjacent to x lie in the inside of circle at center x with radius k.

We first evenly divide this area into several small ones A_i with rays (half lines) at x. Two adjacent rays form an angle α. Suppose xy and xz are two rays with angle α between them. Suppose $d(x,y) \leq d(x,z)$ and $d(y,z) = d(x,y)$. Then from Fact 1, we know that y and z are adjacent. Since $d(y,z) = d(x,y)$, we have $\angle xzy = \alpha$. Hence $d(x,z) = 2d(x,y)cos\alpha$, i.e., $d(x,z)/d(x,y) = 2\cos\alpha$. Hence, each area A_i can be divided into subareas by circles at x with radius $1, 2\cos\alpha, (2\cos\alpha)^2, ..., (2\cos\alpha)^j$. Note that $(2\cos\alpha)^j \leq k$. Hence, $j = \lfloor \frac{\ln k}{\ln(2\cos\alpha)} \rfloor$. Now we need to show that all nodes that are adjacent to x and lie in each subarea are adjacent. Indeed, let y and z be such nodes. Then x, y, and z satisfy the condition in Fact 1.

Therefore, there are at most $\lfloor \frac{\ln R}{\ln(2cos\alpha)} \rfloor (2\pi/\alpha)$ subareas. In other words, x can be adjacent to at most $\lfloor \frac{\ln R}{\ln(2cos\alpha)} \rfloor (2\pi/\alpha)$ independent nodes.

Let $f(\alpha) = \lfloor \frac{\ln R}{\ln(2cos\alpha)} \rfloor (2\pi/\alpha)$. Note that in our proof, $\alpha = 2\pi/m$ where m is an integer and $m > 6$. Hence, we need to find a local minima of $f(\alpha)$ where $0 < \alpha < 2\pi/6$. With some algebraic steps, we have $\alpha = 2\pi/10$. Hence, when $k > 1$, x can be adjacent to at most $10\lfloor \frac{\ln k}{\ln(2cos(\pi/5))} \rfloor$ independent nodes.

□

Theorem 1: In a DGB $G = (V, E)$, the size of any maximal independent set is upper bounded by $Kopt$ where $k = \dfrac{r_{max}}{r_{min}}$ and $K = 5$ if $k = 1$, otherwise, $K = 10\lfloor \frac{\ln k}{\ln(2cos(\pi/5))} \rfloor$

Proof: Let I be an MIS. By Lemma 1, no node in OPT can dominate more than K nodes in I. Thus the theorem follows: $|I| \leq Kopt$.

□

From now on, we will refer k as r_{max}/r_{min} and $K = 5$ if $k = 1$, otherwise, $K = 10\lfloor \frac{\ln k}{\ln(2cos(\pi/5))} \rfloor$.

IV. Approximation Algorithms and Analysis

In this section, we present three constant approximation algorithms for the CDS problem in DGB and analyze their performance ratio.

A. First Algorithm

1) Algorithm Description: The First Algorithm (TFA) has two phases. First, we construct the maximal independent set I, then connect them by finding a set of connector nodes B. This algorithm is similar to [2]. Through out the paper, sometimes, we call a black node as a node in I, a blue node as a connector node in B, and a grey node as a non-CDS node. Note that the set I is also a dominating set of G.

To construct an MIS, we first randomly choose a vertex $u \in V$ and construct the Breath-First Search tree

Algorithm 1 The First Algorithm

INPUT: A DGB $G = (V, E)$, all nodes are white
OUTPUT: A CDS of G
Randomly pick a vertex $u \in V$ and color u black
$I = \{u\}$, $B = \emptyset$, $GREY = \emptyset$
Construct a BFS tree T of G, rooted at u
$d = depth(T)$
for $j = 1$ to d **do**
  $TMP_i = \{\ v_i \mid level(i) = j\}$
  if v_i is dominated by some black nodes in TMP_{i-1} **then**
    Color v_i grey; $GREY = GREY \cup \{v_i\}$
  end if
  Choose a set A of nodes in TMP_{i-1} that is not grey and a maximal independent set in $G(TMP_{i-1} - GREY)$
  Color A black; $I = I \cup A$
  Choose a set C of nodes in TMP_{i-1} that are parent of black nodes in TMP_i
  Color C blue; $B = B \cup C$
end for
Return $I \cup B$

(BFS) T of G rooted at u. Next, we mark u black. Then mark all the neighbor nodes of u grey. For each level $level(i)$ of T, find a maximal independent set of a set of nodes that are neither black nor grey. In other words, we need to find an MIS of the nodes that are not dominated yet. After finding an MIS, the process of connecting them is easy with the help of T. For each black node u, we just need to find a grey node that is a parent of u and be dominated by another black nodes in the previous level. The detail of TFA is shown in Algorithm 1.

2) Theoretical Analysis: The maximal independent set I obtained from TFA satisfies this property:

Lemma 2: Any pair of complementary subsets of the MIS has a distance of exactly two hops.

Proof: This is trivial from the construction of I in the algorithm. If a node $u \in I$, then all $N(u) \in GREY$. If $u \in GREY$, then there exist a node v such that $uv \in E$ and u is black.

□

Theorem 2: TFA produces a CDS with the size bounded by $2Kopt$ where $K = 5$ if $k = \frac{r_{max}}{r_{min}} = 1$, otherwise $K = 10\lfloor\frac{\ln k}{\ln(2cos(\pi/5))}\rfloor$

Proof: From Lemma 2 and Theorem 1, we have:

$$|CDS| \leq 2|I| \leq 2Kopt$$

□

In practice, we expect that k is very small since the transmission ranges of all nodes in a network should be slightly different.

Corollary 1: If the maximum and minimum transmission ranges are bounded, then our algorithm has an approximation factor of $O(1)$.

B. Second Algorithm

In the first algorithm, we connect two black nodes u and v (assume that $level(u) < level(v)$) by finding a grey node w that is a parent of v in T and neighbor of u in G. However, we can connect I by using the Steiner tree, which is a tree interconnecting all nodes in I. The nodes in the Steiner tree but not in I are called Steiner nodes. To reduce the size of an obtained CDS, we need to find a Steiner tree with the Minimum number of Steiner Nodes (MSN). We can define this problem as follows:

Definition 1: **Steiner Tree with MSN**: Given a graph $G = (V, E)$ and a set of nodes $V' \subseteq V$ called *terminals*, construct a Steiner tree T that connects all the terminals such that the number of Steiner nodes is minimum.

1) Algorithm Description: The Second Algorithm (TSA) also has two phases. The first phase is to find an MIS I that satisfies Lemma 2. Note that this condition is vital for the second phase to work. Since the obtained MIS I from TFA satisfies this condition, we can use the procedure in TFA to find the MIS I. At the second phase, we construct a Steiner tree with the minimum number of Steinter nodes to interconnect all nodes in I as follows. Define a *black-blue component* as a connected component of the subgraph induced only by black and blue nodes, *ignoring connections between blue nodes*. Initially, we have $|I|$ black-blue components. Let B be a set of Steiner nodes, called blue nodes. Initially, B is empty. From Lemma 1, we know that each node is adjacent to at most K independent nodes. In other words, a blue node is adjacent to at most K black nodes. Color all nodes in $V - I$ grey. At each iteration, we can find a grey node that is adjacent to most black-blue components and color it blue. Formally, for j from K to 2, at each iteration j, find a grey node v such that v is adjacent to at least j black nodes in *different* black-blue components. Color v blue and re-compute the black-blue components as described in Algorithm 2

2) Theoretical Analysis: The CDS in this algorithm is a union of set I and set B. To analyze the performance ratio of our algorithm, we first compare the size of set B to opt. Recall that B is a set of all the Steiner nodes. Let T^* be an optimal tree when connecting a given set I and $C(T^*)$ is the number of the Steiner nodes in T^*, we have this following lemma:

Algorithm 2 The Second Algorithm

1: INPUT: A DGB $G = (V, E)$, all nodes are white
2: OUTPUT: A CDS of G
3: $I = \emptyset$; $B = \emptyset$
4: Use the procedure in Algorithm 1 to compute I, other nodes are grey at this stage
5: **for** $j = K$ to 2 **do**
6: **while** There exists a grey node v adjacent to at least j black nodes in different black-blue components **do**
7: $B = B \cup \{v\}$
8: **end while**
9: **end for**
10: Return $I \cup B$

Lemma 3: The size of B obtained from TSA is at most $(2 + \ln K)C(T^*)$

Proof: Let $n = |I|$ and $p = |B|$. If $n = 1$, then the lemma is trivial. Assume that $n \geq 2$, thus $C(T^*) \geq 1$. Let $v_j, j = 1...p$ be the blue nodes in the order of appearance in the second phase. Let a_i be the number of the black-blue components after $v_1, ..., v_i$ turns blue. Since every black-blue component contains a black node which is adjacent to a Steiner node of T^*, there exists v_i which is adjacent to at least $\frac{a_i}{C(T^*)}$. Thus we have:

$$a_{i+1} \leq a_i - \frac{a_i}{C(T^*)} + 1$$

Hence we have this following recurrence:

$$\begin{aligned} a_i &\leq a_{i-1} - \frac{a_{i-1}}{C(T^*)} + 1 \\ &\leq a_{i-1}\left(1 - \frac{1}{C(T^*)}\right) + 1 \\ &\leq a_{i-2}\left(1 - \frac{1}{C(T^*)}\right)^2 + \left(1 - \frac{1}{C(T^*)}\right) + 1 \\ &\leq ... \\ &\leq a_0\left(1 - \frac{1}{C(T^*)}\right)^i + \sum_{j=0}^{i-1}\left(1 - \frac{1}{C(T^*)}\right)^j \\ &\leq a_0\left(1 - \frac{1}{C(T^*)}\right)^i + C(T^*) \end{aligned}$$

For the last step in the above recurrence, we note that the second term $\sum_{j=0}^{i-1}\left(1 - \frac{1}{C(T^*)}\right)^j$ is the geometric series and it will converge to $C(T^*)$. After $i = C(T^*)\ln\frac{a_0}{C(T^*)}$ iterations, the number of black-blue components will be:

$$\begin{aligned} a_i &\leq a_0\left(1 - \tfrac{1}{C(T^*)}\right)^i + C(T^*) \\ &\leq e^{-\frac{i}{C(T^*)}} + C(T^*) \\ &\leq 2C(T^*) \end{aligned}$$

Therefore, the total number of blue nodes is bounded as follows:

$$\begin{aligned} |B| &\leq i + 2C(T^*) \leq C(T^*)(\ln\tfrac{a_0}{C(T^*)} + 2) \\ &\leq C(T^*)(\ln\tfrac{n}{C(T^*)} + 2) \leq (2 + \ln K)C(T^*) \end{aligned}$$

□

Theorem 3: The Second Algorithm produces a CDS with size bounded by $(K + 2 + \ln K)opt$ where $K = 5$ if $k = \frac{r_{max}}{r_{min}} = 1$, otherwise $K = 10\lfloor\frac{\ln k}{\ln(2cos(\pi/5))}\rfloor$

Proof: From Theorem 1 and Lemma 3, we have:

$$\begin{aligned} |CDS| &= |I| + |B| \\ &\leq (K + 2 + \ln K)opt \end{aligned}$$

□

Corollary 2: If the maximum and minimum transmission ranges are bounded, then TSA has an approximation factor of $O(1)$.

C. The Third Algorithm

In the two previous proposed algorithms, we find an MIS based on the Breath First Search tree T which is constructed based on the connectivity information of a given network. In this section, we show the effect of the size of the disks on the size of an MIS. We first introduce the following lemma:

Lemma 4: In a DGB G, there exists a node that is adjacent to at most five independent nodes.

Proof: Let D be a disk with radius r_{min} centered at node u. Note that D is the smallest disk in G. We prove that u has at most 5 independent neighbors by contradiction. Suppose that u has more than 5 independent neighbors. Let v_j, $1 \leq j \leq 6$ be the independent neighbors of u. Then there exist two nodes that lie in a sector with the angle less than or equal to 60 degree. Without loss of generality, assume that v_1 and v_2 are such nodes as shown in Figure 2. Then $d(v_1, v_2) \leq r_{min}$. Hence v_1 and v_2 are connected, contradicting to our assumption.

□

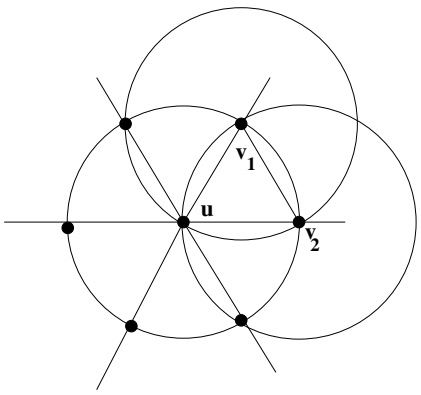


Fig. 2. On the Proof of 5 Independent Neighbors

Note that the subgraph of a DGB is still a DGB. Hence, let us consider the algorithm to find an MIS as shown in Algorithm 3.

Algorithm 3 Choose Smallest Disks

1: INPUT: A DGB $G = (V, E)$
2: OUTPUT: A Maximal Independent Set I
3: $I = \emptyset$
4: **while** $V \neq \emptyset$ **do**
5: Find a node $u \in V$ with the smallest radius, color u black
6: $I = I \cup \{u\}$
7: $V = V - \{u\} - N(u)$
8: **end while**
9: Return I

In this algorithm, at each iteration, we find a node with the smallest radius in V and color it black, then remove this node and its neighbors from V. This step runs iteratively until V is empty. The black nodes form a maximal independent set I. Let I^* be the optimal MIS of G, i.e., $|I^*| \geq |I|$ for any MIS I, we have:

Lemma 5: The size of I is at least $\frac{|I^*|}{5}$

Proof: Every node $v \in V$ is either in I or adjacent to some nodes in I. Since $I^* \subset V$, every node $v \in I^*$ is either in I or adjacent to some nodes in I. Let define $N[u]$ as the closed neighbors of u when adding u into I, i.e., $N[u] = N(u) \cup \{u\}$. Then every node $v \in I^*$ is in $N[u]$ for some $u \in I$. Because at each step, we choose a node u with the smallest disk, each u has at most 5 independent nodes (Lemma 4). Thus each $N[u]$ contains at most 5 vertices from I^*. This results to $|I| \geq \frac{|I^*|}{5}$ □

Now, let us color the biggest disks instead of the smallest disks black. Specifically, as shown in Algorithm 4, at each iteration, we find a node with the largest transmission range in V and color it black. Remove this node and its neighbors from V. The set of black nodes forms a maximal independent set I.

Again, let I^* be the optimal MIS of G, we have the following lemma:

Lemma 6: The size of I is at least $\frac{|I^*|}{K}$ where $K = 5$ if $k = \frac{r_{max}}{r_{min}} = 1$, otherwise $K = 10\lfloor \frac{\ln k}{\ln(2cos(\pi/5))} \rfloor$.

Proof: Using the same approach in the previous proof, by Lemma 1, each $N[v]$ contains at most K independent nodes in I^*. This follows that $|I| \geq \frac{|I^*|}{K}$ □

We believe that the size of I obtained from Algorithm 4 is less than that obtained from the First or Second Algorithm due to the above lemma. Thus we introduce

Algorithm 4 Choose Biggest Disks

1: INPUT: A DGB $G = (V, E)$
2: OUTPUT: A Maximal Independent Set I
3: $I = \emptyset$
4: **while** $V \neq \emptyset$ **do**
5: Find a node $u \in V$ with the biggest radius, color u black
6: $I = I \cup \{u\}$
7: $V = V - \{u\} - N(u)$
8: **end while**
9: Return I

The Third Algorithm (TTA) and its performance is evaluated by simulations. In this algorithm, we first find a set of dominating set I using the Algorithm 4. Then connect I by choosing a node that is adjacent to most black-blue components and color it blue. Recall that the black-blue component is defined as a connected component of the subgraph induced only by black and blue nodes, ignoring connections between blue nodes. The detail of TTA is shown in Algorithm 5.

Algorithm 5 The Third Algorithm

1: INPUT: A DGB $G = (V, E)$, all nodes are white
2: OUTPUT: A CDS
3: $I = \emptyset$; $B = \emptyset$
4: $I =$ **Choose Biggest Disks**(G)
5: **while** I is disconnected **do**
6: Select a white node u such that u is adjacent to most black-blue components
7: Color u blue
8: $B = B \cup \{u\}$
9: **end while**
10: Return $I \cup B$

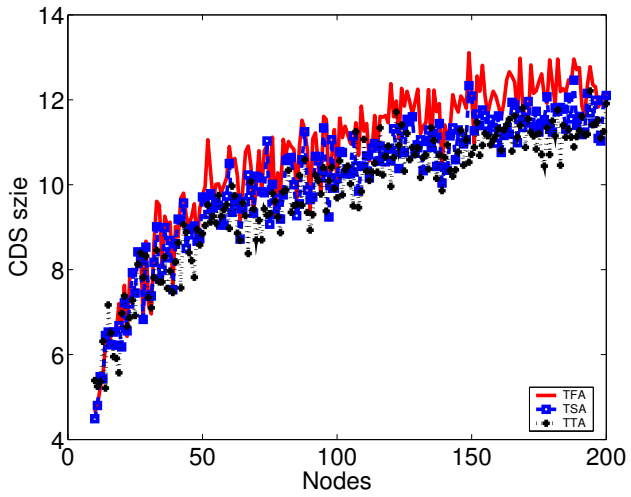
V. Simulation Results

In the previous section, we evaluate our algorithms through theoretical analysis. In this section, we conducted some simulation experiments to measure the performance (in terms of the size of CDS) of three algorithms: The First Algorithm (TFA), The Second Algorithm (TSA), and The Third Algorithm (TTA). Recall that the improvement of TSA over TFA is that we use the Steiner tree with the minimum number of Steiner nodes to interconnect all black nodes. The improvement of TTA over TSA is that we select nodes with largest transmission ranges as the black nodes. Moreover, we are interested in comparing the size of the black nodes obtained from each algorithm to see whether the choosing the biggest disks approach can return the smallest

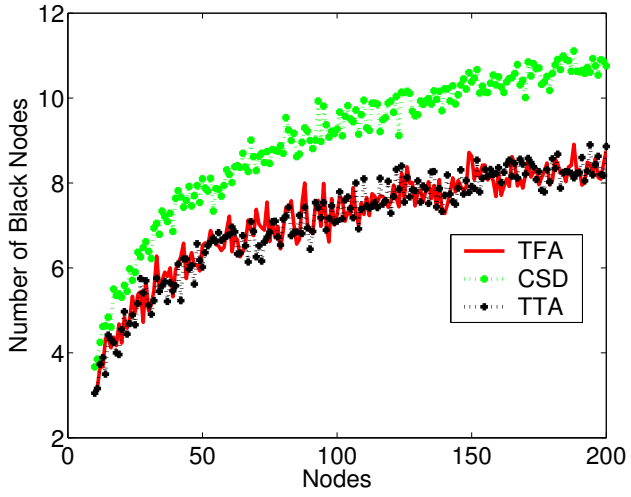
number of black nodes. Since the number of black nodes in TFA and TSA is the same, let I_1 denote the size of black nodes obtained from either TFA or TSA. Let I_b denote the size of black nodes obtained from TTA and I_s be the size of black nodes obtained from the Choose Smallest Disks (CSD) algorithm. We study three network parameters that may affect the algorithm performance:

1) n, the number of nodes in a given network
2) k, the ratio of the largest transmission range to the smallest transmission range, i.e., $k = \frac{r_{max}}{r_{min}}$
3) The network density, i.e., the number of nodes per area

A. Effects of Number of Nodes



(a) Compare the CDS Size



(b) Compare the MIS Size

Fig. 3. Effects of Number of Nodes

To evaluate the performance of the three proposed algorithms under different number of nodes, we randomly deployed n nodes to a fixed area of 800m x 800m. n changed from 10 to 200 with an increment of 1. Each node v_i randomly chose the transmission range $r_i \in [r_{max}, r_{min}]$ where $r_{max} = 600m$ and $r_{min} = 200m$. For each value of n, 1000 network instances were investigated and the results were averaged.

As can be seen in Figure 3(a), the size of a CDS obtained from TTA is smallest among all three algorithms. Specifically, the size of the CDS obtained from TTA is 3.3% smaller than that of TSA, and 8.9% smaller than that of TFA. Also, the size of the CDS obtained from TSA is 5.5% less than that of TFA. The results indicate that constructing the Steiner tree with the minimum number of Steiner nodes to interconnect the maximal independent set can reduce the size of the CDS. In addition, choosing the biggest disk as a black node can reduce the size of the CDS as well.

Figure 3(b) shows the comparison of the number of black nodes obtained from TFA, CSD, and TTA. The number of black nodes I_b obtained from TTA is smaller than that of TFA. The Choose Smallest Disks (CSD) algorithm returns the largest number of black nodes I_s as shown in Figure 3(b). This is consistent with our expectation as we have analyzed in the previous section.

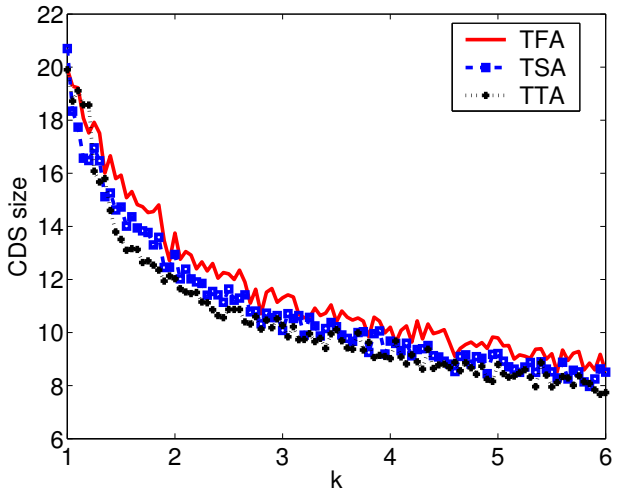
Figure 3 also shows how the number of nodes in a network affects the size of the CDS. In particular, the size of the CDS increases as the number of nodes increases. This fact is obvious since the number of nodes that need to be dominated is larger when we deploy more nodes.

B. Effects of the Transmission Range Ratio

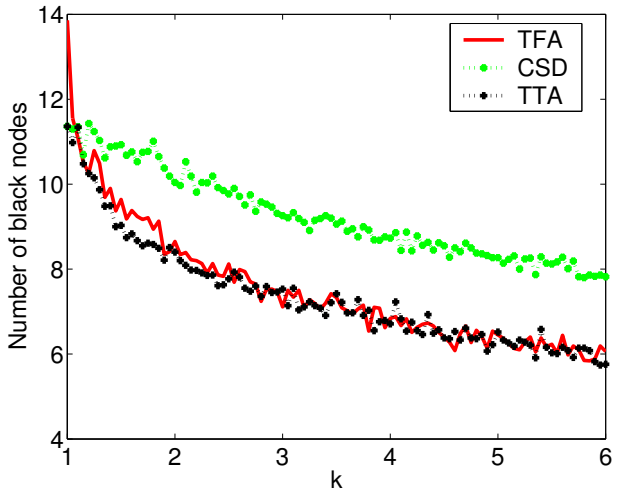
We also conducted simulations to compare the performance of all three algorithms when changing the transmission range ratio k as well as to see how this change affects the size of the obtained CDS. To change k, we fixed $r_{min} = 200m$ and changed r_{max} from 200m to 1200m with an increment of 10. In this experiment, we randomly deployed 100 nodes into a fixed area of size 800m x 800m. Each node randomly chose a transmission range in $[r_{min}, r_{max}]$. For each network instance, we ran the test for 1000 times.

Figure 4(a) compares the performance of three algorithms in terms of the CDS size. As shown in Figure 4(a), TTA is the best. In particular, the CDS size obtained from TTA is 10.7% smaller than that of TFA, and 4% smaller than that of TSA. The resultant CDS from TSA has a size 6.7% smaller than that of TFA. Again, these results reveal that using the Steiner tree to interconnect a dominating set can reduce the CDS size.

As expected, $I_b < I_1 < I_s$ as shown in Figure 4(b). Note that I_s is 21% bigger than I_b. This number is large and significant to increase the size of CDS. This very

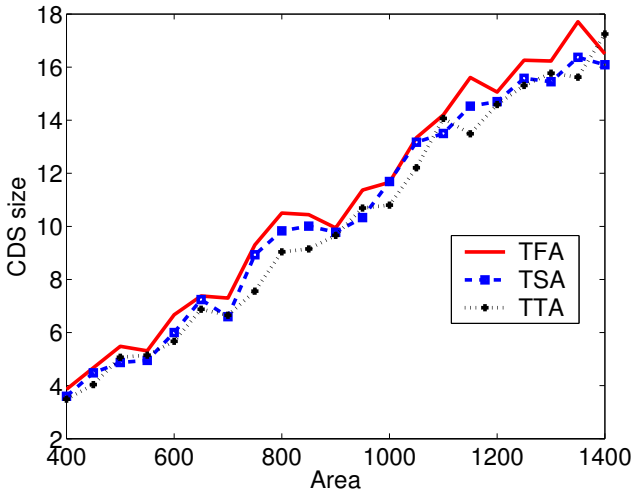


(a) Compare the CDS Size

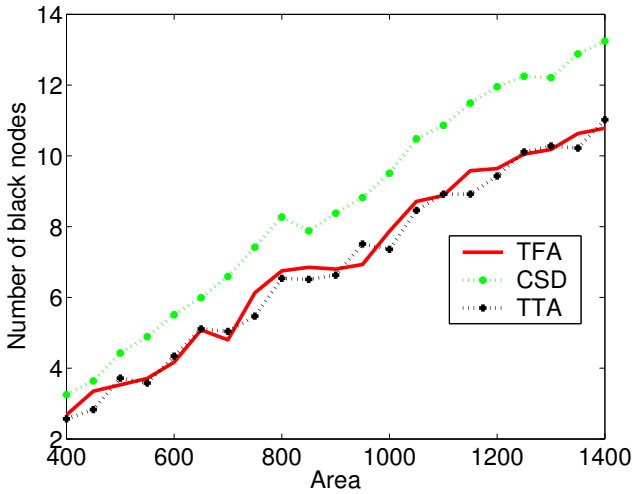


(b) Compare the MIS Size

Fig. 4. Effects of the Transmission Range Ratio



(a) Compare the CDS Size



(b) Compare the MIS Size

Fig. 5. Effects of the Network Density

high percentage is predicted since when k increases, K increases as well. Recall that K is the maximum number of independent neighbors of each node. Since $|I^*|/K \leq |I_b|$, I_b has the potential to decrease as K increases.

Figure 4 illustrates how the transmission ranges affect the CDS size. As can be seen in Figure 4, three curves show obvious trend of decrease. In other words, the CDS size decreases when the maximum transmission range increases. It is due to the fact that the larger the transmission range, the more nodes a node can dominate.

C. Effects of the Network Density

Simulations were also carried out to compare the performance of all three algorithms when changing the network density as well as to see how this change affects the CDS size. To change the network density, we fixed the number of nodes $n = 50$ and increased the area from 400m x 400m to 1,400m x 1,400m with an increment of 50. In this experiment, we randomly generated 50 nodes in an area with the size changing as described. Each node randomly chose a transmission range in $[r_{min}, r_{max}]$ where $r_{min} = 200m$ and $r_{max} = 600m$. For each network instances, we ran the simulations for 1000 times and the results were averaged.

Figure 5(a) provides the performance comparison of three algorithms in terms of the CDS size. As revealed by Figure 5(a), TTA still outperforms the other two in this case. And TSA outperforms TFA. Specifically, the CDS size obtained from TTA is 8% less than that of TFA and 3.2% less than that of TSA. Moreover, the CDS size obtained from TSA is 4.9% less than that of TFA. As predicted, Figure 5(b) indicates that $I_b < I_1 < I_s$. The number of black nodes obtained from TTA is slightly less than that of TFA but is much less than that of the CSD algorithm.

In addition, Figure 5 shows the obvious trend of increase of three curves, which implies that the CDS size gets bigger when the network density decreases. This is because when the network density decreases, the neighbors of each node decreases as well. Thus the CDS size need to be larger to dominate all nodes in a network.

In conclusion, for all aspects that we have studied, TTA is the best algorithm. Next is the TSA. Choosing nodes with the largest transmission ranges for the dominating set and using the Steiner tree with the minimum number of Steiner nodes to interconnect the dominating set can reduce the CDS size. Specifically, choosing nodes with the biggest radius can form a smaller dominating set. With the help of the Steiner tree, the number of blue nodes can be reduced. The size of a CDS obtained using these two mechanisms is about 10% less than that obtained without using them. In addition, the simulation results reveal that the CDS size increases as the number of nodes increases. The CDS size also can get larger if the network gets sparser. Furthermore, when the transmission ranges increase, the CDS size decreases.

VI. Distributed Implementations

From the practical point of view, all algorithms designed in wireless networks should be distributed. In this section, we discuss how to implement our three algorithms as distributed algorithms. There exist several distributed algorithms for constructing an MIS satisfying Lemma 2 in literature [2], [5]. Specifically, the authors constructed an arbitrary rooted spanning tree T by the distributed leader-election algorithm in [17]. This algorithm has an $O(n)$ time complexity and $O(n \log n)$ message complexity where n is the number of nodes in a given network. After constructing the spanning tree T, Wan *et. al* [2] introduced a distributed construction on how to find a maximal independent set using the color mechanism with $O(n)$ message complexity and $O(n)$ time complexity. We can use this construction for our TFA. Hence, the distributed implementation of TFA has $O(n \log n)$ message complexity and $O(n)$ time complexity. Now we present the distributed implementations for TSA and TTA.

A. Distributed Version of TSA

For The Second Algorithm, we first find the MIS that satisfies Lemma 2, which we can use the above method. We thus only present a distributed algorithm for the second phase, that is to find a Steiner tree to interconnect the maximal independent set. Note that after running the first phase, all nodes in an MIS are black and all other nodes are grey.

Algorithm 6 Distributed Version of TSA Second Phase

1: INPUT: A maximal independent set I and $G = (V, E)$, all nodes $v_i \in I$ are black, and $v_j \in V - I$ are grey
2: OUTPUT: Color connectors blue
3: Set ID_C of each black node equal to the Black nodes ID $\{ID_C$ is the black-blue component ID$\}$
4: Set ID_C of each grey node equal to -1
5: $ID_C = -1$ for all grey node v_j
6: Each grey node maintains the ADJ list which is the list of its adjacent black nodes in different black-blue components
7: Each grey node maintains a $COMPETITORS$ list
8: Each grey node maintains a global value B, $B = K$ initially
9: v_i sends a BLACK message contained its ID_C
10: Upon receiving the BLACK message, grey node v_j updates its ADJ and $COMPETITORS$ lists
11: v_j sends a GREY message contained its id and its $|ADJ|$
12: v_j turns blue if its $|ADJ| > |ADJ|$ of its neighbors and its $|ADJ| > 1$
13: Each blue node updates its ID_C to the smallest value in its ADJ list
14: A blue node then sends a BLUE message contained its new ID_C and new ADJ list
15: Upon receiving a BLUE message, black node v_i updates its ID_C and send a BLACK message
16: Upon receiving a BLUE message, a GREY node decreases B by 1
17: If $|ADJ|$ of a grey node v_j equal to 1, then do nothing

As described in Algorithm 6, all black nodes v_i in the maximal independent set I maintains its black-blue component id, i.e., ID_C. Initially, we have $|I|$ black-blue components. Hence, ID_C of each black node can be set to the node ID. Each grey node also maintains its black-blue component id and initially, $ID_C = -1$, which indicates that it does not belong to any black-blue component yet. Each grey node also maintains a list of its adjacent black-blue components with there ID_C values, called ADJ and a list of its competitors called $COMPETITORS$. The grey node is adjacent to a black-blue component if it is adjacent to a *black* node in the black-blue component. A grey node u is a competitor of a grey node v if the number of adjacent black-blue components of v and u are the same. At this time, the node with the smaller node id becomes a blue node. Hence the $COMPETITORS$ list contains a list

of competitors node id. Each grey nodes also maintain a global value B which represents the maximum number of independent neighbors. Initially, $B = K$.

Note that after finding a maximal independent set, we still has a spanning tree T. Thus each node also maintain a list of its children in T, called $CHILDREN$. Initially, a root node of T which is a black node sends the BLACK message contained its ID_C to its one hop neighbors. Upon receiving a BLACK message, the grey node v_j add the ID_C in the BLACK message to its adjacent black-blue components ADJ. If this number ID_C is already in ADJ, it does nothing. After updating its ADJ, the grey node then broadcasts the GREY message $< |ADJ|, id >$. Note that id is the grey node id and $|ADJ|$ is the size of the ADJ list. Upon receiving the GREY message, a grey node compares its $|ADJ|$ to the $|ADJ|$ in the GREY message. If its $|ADJ|$ is equal to the $|ADJ|$ in the GREY message, it adds the grey node id in the GREY message to its $COMPETITORS$ list.

When a node is a leaf, besides broadcasting the BLACK or GREY message depending on its color, it also broadcasts the END message. Upon receiving the END message, a GREY node turns to blue according to the follows:

- Its $|ADJ| \geq B > 1$ and
- Its id is smaller than all id in its $COMPETITORS$ list

After turning its color to blue, a blue node updates its ID_C to the smallest number in its ADJ list and decreases B by 1. The blue node then sends a BLUE message and keeps it color permanent. A BLUE message contains its id and its ADJ list. Upon receiving a BLUE message, all black nodes update their ID_C to the smallest number in the BLUE message and send the BLACK message out. Note that all nodes in the same black-blue component must have the same ID_C. At the end of this algorithm, the grey node keeps its color grey if its $|ADJ|$ is 1. A node stops sending message if it is adjacent to one black-blue component. This indicates that either the all black nodes are connected at this time or a node is just adjacent to only one black node. The main idea of this distributed version is shown in Algorithm 6.

Theorem 4: The distributed version of TSA has an $O(n \log n)$ message complexity and $O(n)$ time complexity.

Proof: The time and message complexity of the MIS construction phase is dominated by the time and message complexity of constructing the rooted spanning tree T which are $O(n)$ and $O(n \log n)$ respectively [2]. For the second phase, each node sends at most $O(n \log n)$ messages and takes at most linear time. Hence the message complexity of distributed TSA is $O(n \log n)$ where its time complexity is $O(n)$.

□

B. Distributed Version of TTA

The distributed version of TTA consists two phases as shown in Algorithm 7. The first phase is to find a dominating set such that at each iteration, we select a node with the largest transmission range. And the second phase is to connect the above dominating set.

Algorithm 7 Distributed Version of TTA

1: INPUT: A DGB $G = (V, E)$ with all nodes in white
2: OUTPUT: A CDS
3: Each white node maintains a $SORT$ list
4: Each white node v_i broadcasts a WHITE message $< id_i, r_i >$
5: Upon receiving a WHITE message, each node updates its $SORT$ list
6: A node with its id at the beginning of the $SORT$ list marks itself black and sends the BLACK message contained its id
7: Upon receiving a BLACK message, a white node marks itself grey and broadcasts the GREY message contained its id and id in the BLACK message
8: Upon receiving a GREY message, a white node updates its $SORT$ list
9: Use the Algorithm 6 to connect all black nodes

Initially, all nodes are white. Each node maintains a list of all node id in the decreasing order of the transmission ranges, called $SORT$ list. At the beginning, the $SORT$ list of each node contains its own id. Each white node v_i broadcasts a WHITE message containing its own id and its transmission range $< id_i, r_i >$. Upon receiving a WHITE message, each node updates its $SORT$ list by adding the id in the WHITE message in the decreasing order of transmission ranges. A node which has its id at the head of the $SORT$ list has the largest transmission range.

A white node which has its id at the head of the $SORT$ list marks itself black and sends the BLACK message to its neighbors. The BLACK message contains the black node id. Upon receiving the BLACK message, a white node marks itself grey. The grey node then broadcasts a GREY message which contains its own id and id in the BLACK message. Upon receiving the GREY message, a white node updates its $SORT$ list by removing the id in the grey message from the $SORT$ list.

Once a node makes itself black or grey, its color is unchanged. This process stops when there does not exist any white node. Note that after marking itself black or grey and sending out the BLACK or GREY message, this node will not join in the coloring process anymore.

At the end of this phase, all nodes in the network are either black or grey. All black nodes form a dominating set. Now, we need to connect these black nodes. The process is similar to the distributed version of TSA Second Phase.

Theorem 5: The distributed version of TTA has an $O(n^2)$ message complexity and $O(n^2)$ time complexity.
Proof: The time and message complexity of the first phase is dominated by the sorting part, i.e, to compute the $SORT$ list of each node. Since each node broadcasts a WHITE message, the time and message complexity is $O(n^2)$. The second phase uses $O(n \log n)$ message and takes at most linear time. Hence the message complexity of distributed TTA is $O(n^2)$ and its time complexity is also $O(n^2)$.

□

VII. Conclusion

In this paper, we have studied the Connected Dominating Set (CDS) problem in Disk Graphs with only Bidirectional links (DGB). The disk graphs can be used to model wireless ad hoc networks where nodes have different transmission ranges. We have proposed three approximation algorithms and shown that the obtained CDS is within a constant factor of the optimal CDS. The main approach in our algorithms is to construct a maximal independent set and then connect them. Through the theoretical analysis and simulation results, we have shown that using a Steiner tree with the minimum number of Steiner nodes to interconnect the maximal independent set can help to reduce the size of the CDS. In addition, choosing a node with the largest transmission range as a dominator can further reduce the CDS size.

Moreover, we have also presented the size relationship between an independent set and a CDS of a given network. We have pointed out some important characteristics of a DGB. In particular, given a DGB G, there exists a node such that the maximum number of its independent neighbors is 5. In addition, we have also proved the upper bound of the maximum number of independent neighbors of any node in a DGB.

When nodes in a network have different transmission ranges, a node u can communicate directly to a node v but node v might not be able to communicate directly back to node u. In this case, the edge (u, v) is a directed edge, called unidirectional links. Thus we are interested to study the CDS problem in the general disk graphs, where both unidirectional and bidirectional links exist. One simple way is to find a dominating set and then use a directed Steiner nodes algorithm to connect them. Note that the CDS in this case is directed. Hence we need to find a strongly connected CDS to help the routing.

References

[1] B. Clack, C. Colbourn, and D. Johnson, "Unit Disk Graphs," *Discrete Mathematics*, vol. 86, pp. 165–177, 1990.

[2] P.-J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed Construction on Connected Dominating Set in Wireless Ad Hoc Networks", *Proceedings of the Conference of the IEEE Communications Society (INFOCOM)*, 2002.

[3] Y. Li, M. T. Thai, F. Wang, C.-W. Yi, P.-J. Wang, and D.-Z. Du, "On Greedy Construction of Connected Dominating Sets in Wireless Networks", *Special issue of Wireless Communications and Mobile Computing (WCMC)*, 2005.

[4] S. Funke, A. Kesselman, U. Meyer, and M. Segal, "A Simple Improved Distributed Algorithm for Minimum CDS in Unit Disk Graphs", *1st IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2005.

[5] M. Cardei, M.X. Cheng, X. Cheng, and D.-Z. Du, "Connected Domination in Ad Hoc Wireless Networks", *Proceedings of the Sixth International Conference on Computer Science and Informatics (CSI)*, 2002.

[6] S. Guha and S. Khuller, "Approximation Algorithms for Connected Dominating Sets", *Algorithmica*, vol. 20, pp. 374–387, 1998

[7] L. Ruan, H. Du, X. Jia, W. Wu, Y. Li, and L.-I. Ko, "A Greedy Approximation for Minimum Connected Dominating Sets", *Theoretical Computer Science*, 2005. To appear

[8] J. Wu and H. Li, "On Calculating Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks", *Proceedings of the Third International Workshop Discrete Algorithms and Methods for Mobile Computing and Comm.*, 1999.

[9] Y. Li, S. Zhu, M. T. Thai, and D.-Z. Du, "Localized Construction of Connected Dominating Set in Wireless Networks", *NSF International Workshop on Thoretical Aspects of Wireless Ad Hoc, Sensor and Peer-to-Peer Networks*, 2004.

[10] K.M. Alzoubi, P.-J. Wang, and O. Frieder, "Message-Optimal Connected Dominating Sets in Mobile Ad Hoc Networks", *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2002.

[11] B. Das, R. Sivakumar, and V. Bharghavan, "Routing in Ad Hoc Networks Using a Spine", *International Conferfence on Computers and Communication Networks*, 1997

[12] B. Das and V. Bharghavan, "Routing in Ad Hoc Networks Using Minimum Connected Dominating Sets", *International Conference on Communications*, 1997.

[13] R. Sivakumar, B. Das, and V. Bharghavan, "An Improved Spine-based Infrastructure for Routing in Ad Hoc Networks", *IEEE Symposium on Computers and Communications*, 1998.

[14] M. R. Garey, D. S. Johnson, *"Computers and Intractability. A guide to the Theory of NP-completeness"*, Freeman, New York, 1979.

[15] K.M. Alzoubi, P.-J. Wan, and O. Frieder, "New Distributed Algorithm for Connected Dominating Set in Wireless Ad Hoc Networks", *Proceedings of the 35th Hawaii International Conference on System Scicences*, Hawaii, 2002.

[16] K.M. Alzoubi, P.-J. Wan, and O. Frieder, ”Distributed Heuristics for Connected Dominating Sets in Wireless Ad Hoc Networks”, *Journal of Communications and Networks*, vol. 4, no. 1, March 2002.

[17] I. Cidon and O. Mokryn, ”Propagation and Leader Election in Multihop Broadcast Environment”, *The 12th International Sysposium on DIStributed Computing (DISC)*, 1998