TRADE-OFF SCHEME FOR FAULT TOLERANT CONNECTED DOMINATING SETS ON SIZE AND DIAMETER IN WIRELESS AD-HOC NETWORKS

By

NING ZHANG

A THESIS PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

2008

To my parents, Heping Shi and Quangui Zhang

ACKNOWLEDGMENTS

My sincerest gratitude goes to my advisor, Professor My T. Thai, for her invaluable guidance and support throughout my master's research work. During the initial several months of my graduate work, Professor Thai was extremely patient and always led me toward the right direction whenever I would waver. Her acute insight into the research problems we worked on set an excellent example and provided me immense motivation. She has always emphasized the importance of high-quality technical writing and has spent several painstaking hours reading and correcting my technical manuscripts. She has been the best mentor I could have hoped for, and I shall always remain indebted to her for shaping my career and more importantly, my thinking.

I am also very thankful to Professor Ding-Zhu Du at UT-Dallas, for his guidance during my graduate study. His enthusiasm and constant willingness to help has always amazed me. Thanks are also due to Professor Jeffrey Ho, for his support during my graduate study. I take this opportunity to thank Professors Shigang Chen and Ye Xia, for taking the time to serve on my committee and for their helpful suggestions.

It was a pleasure working with Incheol Shin on various collaborative research projects. Several interesting technical discussions with Reza Mahjourian, Bo Li, Vishak Sivakumar and Ying Xuan provided a stimulating work environment in the Applied Optimization group.

This work would not have been possible without the constant encouragement and support of my family. My parents, Heping Shi and Quangui Zhang, always encouraged me to focus on my goals and pursue them against all odds. My grandfather and grandmother, Rongfu Shi and Guiying Lu, have always placed trust in my abilities and have been ideal examples to follow since my childhood.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

TRADE-OFF SCHEME FOR FAULT TOLERANT CONNECTED DOMINATING SETS ON SIZE AND DIAMETER IN WIRELESS AD-HOC NETWORKS

By

Ning Zhang

August 2008

Chair: My T. Thai
Major: Computer Engineering

Connected Dominating Set (CDS) has been a well known approach for constructing a virtual backbone to alleviate the broadcasting storm in Wireless Ad-hoc Network. Current research has focused on minimizing the size or diameter, or improving the fault tolerance of CDS. However, to our best knowledge, no existing research has considered these three important factors together in a single model. In this work, we introduce the fault tolerant model studying a joint optimization problem in which the objective is to minimize the CDS size as well as the diameter, leading to the decrease in network latency. This model also addresses the tradeoffs between the three objective functions. Simulation results show that our solutions can gain good tradeoffs between the three factors, which coincide with theoretical analysis. Moreover, our solutions could obtain a better performance than others' work.

CHAPTER 1
INTRODUCTION

1.1 Wireless Ad-hoc Network

A wireless ad-hoc network is a decentralized wireless network. The network is ad-hoc because each node is willing to forward data for other nodes, and so the determination of which nodes forward data is made dynamically based on the network connectivity. This is in contrast to wired networks in which routers perform the task of routing. It is also in contrast to managed wireless networks, in which a special node known as an access point manages communication among other nodes.

As we know, the key to improve performance of computer networks is to organize the clients in network into hierarchy. However, due to the lack of topology of network, wireless ad-hoc network are flat in nature. In order to achieve high performance, some algorithms have appeared that rely on a virtual backbone, which groups all clients into a hierarchy. The virtual backbone is the first application of Connected Dominating Set (CDS) in wireless ad-hoc network [37].

1.2 Connected Dominating Set (CDS): A Virtual Backbone in Wireless Ad-hoc Network

CDS has been applied in wireless ad-hoc networks performing a lot of functions. For example, the CDS worked as virtual backbone to include media access coordination [1–3], unicast [4–6], multicast/broadcast [7–13], and location-based routing [14]; energy conservation [15–19]; and topology control [16, 20]. CDS can also be employed to achieve the discovery of resource in mobile ad-hoc network [21, 22].

In Section 3.1 and 3.2, we are going to survey the CDS construction algorithms in previous research work.

1.3 Work Overview

This work begins with a description of network model, CDS application, quality issues of CDS and contributions of work in Chapter 2, then followed by a survey of related work in Chapter 3. The detailed analysis of a joint optimization model that solves our problem,

are discussed in Section 4. The results of experiments from the proposed solutions and a evaluation of the proposed solutions follow in Chapter 5. Finally, we present conclusions and potential future work in Chapter 6.

CHAPTER 2
NETWORK MODEL, CDS QUALITY ISSUES

The following sections provides background information for the CDS in wireless ad-hoc networks. We first present a mathematical model for the networks under consideration and introduce useful terminology and definitions from graph theory. Then we sketch the application of CDS in wireless ad-hoc network and talk about the quality issues of CDS.

2.1 Network Model

Previous work focused on computing the minimum connected dominating set (will be defined later) under the assumption that the transmission range of each node is equal. However, in practice, the transmission ranges of all nodes are not necessarily equal. Nodes in a network may have different powers due to differences in functionalities. The node's transmission range may be adjusted differently based on the node distribution in a network and the requirements of the applications. 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 the two dimensional 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* (DG). 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 this work, we study the fault tolerant 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. Fig. 2-1 gives an example of DBG representing a network. In Fig. 2-1, the dotted circles represent the transmission ranges and the black nodes represent a CDS, we will give the definition of CDS in the following.

A number of definitions from graph theory are used in this section. Fig. 2-2 can help to illustrate the following concepts:

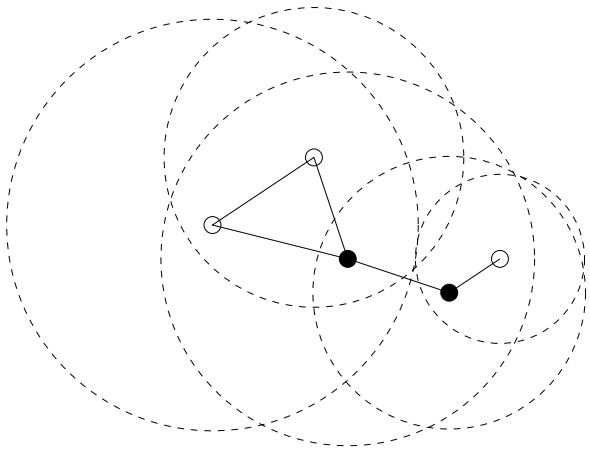


Figure 2-1. A disk graph with bidirectional links

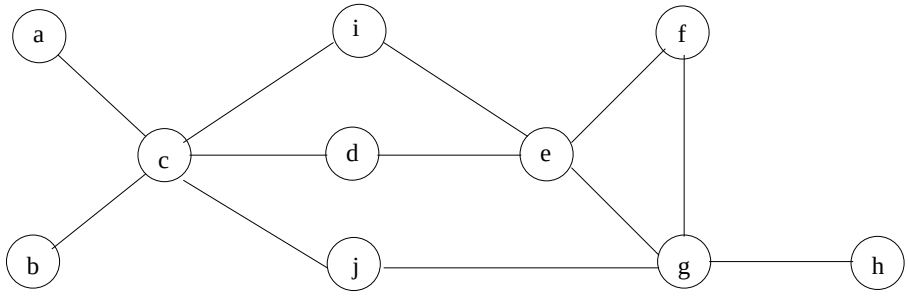


Figure 2-2. Representation of a wireless network with ten nodes as a graph

Independent Set (Neighbor), is a subset of V such that no two vertices within the set are adjacent in V. For example, a, b, f, h, j is an independent set in Fig. 2-2.

Maximal Independent Set (MIS), is an independent set such that adding any vertex not in the set breaks the independence property of the set. Thus, any vertex outside of the maximal independent set must be adjacent to some nodes in the set. The previous independent set a, b, f, h, j must have node d added to become an MIS.

Dominating Set (DS), S, is defined as a subset of V such that each node in $V - S$ is adjacent to at least one node in S. Thus, every MIS is a dominating set. However, since nodes in a dominating set may be adjacent to each other, not every dominating set is an MIS. Finding a minimum-sized dominating set is NP-Hard [29].

Connected Dominating Set (CDS), C, is a DS of G which induces a connected subgraph of G. One approach to construct a CDS is to find MIS, and then add additional nodes to connect the nodes in the MIS. A CDS in Fig. 2-2 is c, d, e, g.

Minimum Connected Dominating Set (MCDS) is the CDS with minimum cardinality. Given that finding minimum sized DS is NP-Hard, it should not be surprising that finding the MCDS is also NP-Hard [29]. In Fig. 2-2, c, j, g is a MCDS.

k-Connected Graph is a graph G if it is connected and removing any $k-1$ nodes from G will not partition G, i.e, G is still connected.

Fault Tolerant Connected Dominating Set (k-m-CDS) is a subset $C \subseteq V$ which satisfying the following two conditions: (i) the subgraph induced by C, i.e., $G[C]$, is a k-connected graph, and (ii) each node not in C is dominated (adjacent) by at least m nodes in C. In Fig. 2-2, c, d, j, e, g is a 2-1-CDS.

Diameter of a connected graph G is the number of hop counts in the longest shortest path between any pair of nodes in G. Denote d_{ij} as the number of hops in the shortest path between node i and node j. Then the diameter of a CDS $d(CDS) = max(d_{ij})$, where i and j are any two nodes in CDS. For example, in Fig. 2-2, the diameter of the graph is 4.

2.2 The Quality Issues of CDS

The major goal of many works is to determine a minimum CDS (MCDS) in unit disk graph (UDG) [23]. Thus, the major goal is to calculate the performance ratio. Minimizing the cardinality of the CDS can decrease the control overhead since broadcasting for route discovery [24, 25] and topology updates [26] is restricted to a small set of nodes [23]. Therefore, broadcasting storm problem [27] inherent to flooding can be greatly released.

Since the nodes in a CDS need to carry other nodes traffic, *fault tolerance* must be also considered. Unfortunately, CDS is often very vulnerable due to frequent node failure and link failure, which is inherent in wireless ad hoc networks. Therefore, constructing a

fault tolerant CDS that continues to function during node or link failure is an important research problem.

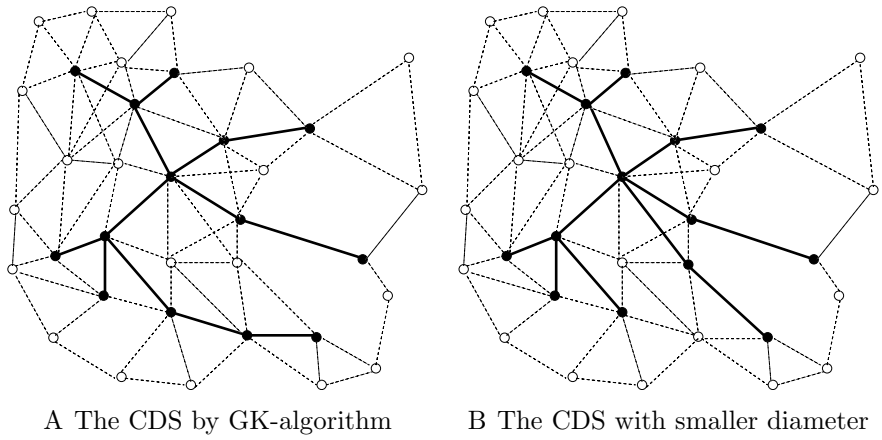


Figure 2-3. An example for the diameter of CDS

A CDS with large diameter often leads to an increase in the propagation error. On the other hand, a CDS with small diameter is certainly preferred for reliable message delivery and short delay in this case. For example, as discussed in [45], a CDS, shown in Fig. 2-3(a), produced by the well known GK-algorithm [28] has a relatively smaller size but larger diameter. However, the network admits a CDS with smaller diameter as shown in Fig. 2-3(b).

2.3 The Contributions of Work

Since a CDS problem is NP-hard [29] and it is easy to reduce the CDS problem to our model in polynomial time. Therefore, we expect that our model is also NP-hard.

As to our best knowledge, no existing research has considered these three important factors together in a single model. In this work, we first study the problem of constructing a CDS by considering the three factors together. The approximation algorithm for k-Connected m-Dominating Sets (k-m-CDS, also known as fault tolerant CDS) with bounded diameter is presented. We minimize the size and the diameter of k-m-CDS while maintaining its fault tolerance. The tradeoffs between objective functions are shown by proving its' approximation ratios. Meanwhile, as 1-Connected Dominating Sets (1-CDS)

with bounded diameter is the prerequisite of our model, two distributed algorithms are addressed for constructing 1-CDS, which consider the size and diameter of it at the same time. Solid proofs for their approximation ratios are presented, which indicate the tradeoffs on size and diameter also exists in 1-CDS. Moreover, our model allows user-defined inputs to balance the size and diameter of 1-CDS. In the end, we evaluate our proposed algorithms through the experiments.

The contributions of this work are as follows:

An approximation algorithm for k-m-CDS with bounded diameter is proposed. Three factors are optimized at the same time and the tradeoffs between objective functions are presented in theoretical analysis and simulation. To our best knowledge, it is the first work to study this model.

Two approximation algorithms to minimize the size and diameter of 1-CDS in disk graphs are presented in distributed manner for our model as a whole. The benefits of proposed algorithm are either featured with low time complexity or effective in minimizing the size and diameter of 1-CDS.

The performance of our model is adjustable by the user-defined input. Through extensive simulation, we verify this fact and the results show our algorithms will have good tradeoffs between the three factors, which coincide with theoretical analysis.

Comparing with CDS-BD algorithm proposed in [47], the simulation results show that our algorithms outperform CDS-BD under the same network condition.

CHAPTER 3
RELATED WORK

In this section, we describe the main ideas of many related work on constructing a CDS with their theoretical analysis results.

3.1 Basic Results

3.1.1 General Graph

Several work have been studied in general graph. In [28], two polynomial time algorithms to construct a CDS in a general graph is proposed by the authors. The first algorithm has performance ratio of $2(H(\delta) + 1)$, where H is a harmonic function and δ is the maximum degree of G. The idea of the first algorithm is to identify the node with a maximum degree as the root and then build a spanning tree T at the root, grow T until all nodes are added to T. Then, all leaf nodes are cut off and the remaining nodes in T are a CDS. The second algorithm is a progress of the first algorithm. The second algorithm consists of two steps. The first step is to construct a dominating set and the second step is to connect the dominating set with a Steinter tree. With such improvement, the second algorithm has a better performance factor of $H(\delta) + 2$. Later, the two algorithms were simulated by Das *et al.* in [4, 30, 31].

3.1.2 Unit Disk Graph

In UDG, most of proposed algorithms are to find an MIS and then connect the MIS with minimum number of nodes. In [32–34], the authors presented a distributed algorithm with a constant performance ratio of 8. Later, Cardei *et al.* presented another distributed algorithm in [35]. This algorithm has the same performance ratio as previous work. However, the message complexity is lower than that of [32].

As we know that distributed algorithm has a better performance than localized algorithms. In the localized algorithms, in [36], Alzoubi *et al.* proposed a localized algorithms with a performance ratio of 192. Although the performance of [36] can not compete with that of [32] and [35]. Their algorithm only need one hop neighbors

information. Therefore, once a node knows that it has the smallest ID among it neighbors, it becomes a dominator. Then, the dominators can be connected by the intermediate nodes in the next step. In [38], Li *et al.* proposed another localized algorithm with a performance ratio of 172, which is better than [36].

3.1.3 Disk Graphs with Bidirectional links

Since the specific geographical characteristics of DGB, not all CDS construction algorithms that are applicable in UDG can be applied to DGB. As far as we know, the algorithms in [32, 35] are applicable in DGB. In [39], Thai *et al.* first proposed the performance ratio of CDS on size in DGB and the two proposed algorithms can be implemented by distributed ways. However, the only difference between two algorithms is the strategy to select MIS, the first algorithm employed Wan's algorithm [32] to choose the nodes in MIS, while the second algorithm used the greedy strategy, that is to include the minimum number of nodes in MIS, thus leading to a better performance than the first algorithm.

3.2 Recent Work

The main goal of above work is to minimize the size (the number of nodes) of CDS. In the following, we will discuss the recent work on quality issues of CDS: *fault tolerance* and *diameter*.

In [40], Wang *et al.* introduced the problem of constructing k-m-CDS in UDG and proposed a constant approximation algorithm. However, Wang *et al.* only studied a special case where $k = 2$ and $m = 1$, which is not general for the k-m-CDS problem. In [41], Shang *et al.* studied the k-m-CDS problem with $k = 2$ and m is an arbitrary number. The results in [41] are still not general, since k is restricted to a constant number. Based on the extension of [40], Thai *et al.* independently proposed an approximation ratio of k-m-CDS in general case [42], where k and m are not bounded to constant numbers.

Besides the approximation algorithms, several distributed algorithm without performance ratios are also proposed in [43, 44] for fault tolerant CDS. In [44], Dai *et*

al. address the problem of constructing k-connected k-dominating virtual backbone which is k-connected and each node not in the backbone is dominated by at least k nodes in the backbone. They propose three localized algorithms. Two algorithms, k-Gossip algorithm and color based k-CDS algorithm, are probabilistic. In k-Gossip algorithm, each node decides its own backbone status with a probability based on the network size, deploying area size, transmission range, and k. Color based k-CDS algorithm proposes that each node randomly selects one of the k colors such that the network is divided into k-disjoint subsets based on node colors. For each subset of nodes, a CDS is constructed and k-CDS is the union of k CDSs. The deterministic algorithm, k-Coverage condition, only works in very dense network and no upper bound on the size of resultant backbone is analyzed. In [43], Wu *et al.* proposed one centralized heuristic algorithm CGA and two distributed algorithm, DDA which is deterministic and DPA which is probabilistic, to construct a k-m-CDS for general k and m.

The diameter of CDS has not been studied extensively up till now. Mohammed *et al.* mentioned the problem of constructing CDS with small diameter [45]. they modified the stepping of GK-algorithm in breadth first manner mimicking the construction of the breadth first search (BFS) tree [46]. The algorithm starts the construction of the tree from a selected node as the root. The nodes are processed in the order of increasing hop distance from the root. However, they did not give a guaranteed performance in their model. In [47], Li *et al.* studied the CDS problem with bounded diameter in UDG and proposed a constant approximation algorithm, called CDS-BD. The main idea of CDS-BD is to select the nodes in MIS level by level and then connect the MIS by constant number of nodes and hops. The level of each node is defined as the number of hops from the root. However, their algorithm is centralized and can only be applied to UDG.

CHAPTER 4
JOINT OPTIMIZATION MODEL

As to our best knowledge, no existing research has considered these three important factors, *size, fault tolerance and diameter of CDS*, together in a single model. In this work, we first study the problem of constructing a CDS by considering the three factors together. The approximation algorithm for k-m-CDS (fault tolerant CDS) with bounded diameter and size is presented. We minimize the size and the diameter of k-m-CDS while maintaining its fault tolerance. The tradeoffs between objective functions are shown by proving its' approximation ratios. Meanwhile, as 1-Connected Dominating Sets (1-CDS) with bounded diameter is the prerequisite of our model, two distributed algorithms are addressed for constructing 1-CDS, which consider the size and diameter of it at the same time. Solid proofs for their approximation ratios are presented, which indicate the tradeoffs on size and diameter also exists in 1-CDS. Moreover, our model allows user-defined inputs to balance the size and diameter of 1-CDS. In the end, we evaluate our proposed algorithms through the experiments.

The contributions of this work are as follows:

An approximation algorithm for k-m-CDS with bounded diameter is proposed. Three factors are optimized at the same time and the tradeoffs between objective functions are presented in theoretical analysis and simulation. To our best knowledge, it is the first work to study this model.

Two approximation algorithms to minimize the size and diameter of 1-CDS in disk graphs are presented in distributed manner for our model as a whole. The benefits of proposed algorithm are either featured with low time complexity or effective in minimizing the size and diameter of 1-CDS.

The performance of our model is adjustable by the user-defined input. Through extensive simulation, we verify this fact and the results show our algorithms will have good tradeoffs between the three factors, which coincide with theoretical analysis.

Comparing with CDS-BD algorithm proposed in [47], the simulation results show that our algorithms outperform CDS-BD under the same network condition.

4.1 Fault Tolerant Model

In this section, we introduce our problem and provide a solution for k-m-CDS with bounded size and diameter, under the condition that $1 \leq k \leq m+1$.

4.1.1 Problem Definition

Before we introduce the definition of the problem, we need to give the following definitions: A *separating set or cut-vertex* of a graph $G = (V, E)$ is a set $S \subseteq V$, such that $G - S$ has more than one component. When $|S| = 1$, S is called a cut vertex. A *k-block* of a graph is a maximal k-connected subgraph of G that has no separating set. If G itself is k-connected and has no separating set, then G is a k-block. The k-m-CDS with bounded diameter problem could be formally defined as follows:

Definition 1. ***k-m-CDS Problem with Bounded Diameter:*** *Given a DGB $G = (V, E)$ representing a network and two positive integers k and m, find a subset $C_{km} \subseteq V$ satisfying the following three conditions: (1) the subgraph induced by C_{km}, i.e., $G[C_{km}]$, is k-connected, and (2) each node not in C_{km} is dominated (adjacent) by at least m nodes in C_{km}. (3) the size and diameter of C_{km} are bounded.*

4.1.2 Notations

Some notations for better understanding of the algorithm description are list below.

I_i be any Maximal Independent Set (MIS) in $G - (I_1 \cup I_2 \cup ... \cup I_{i-1})$

$N_{ID}(u)$ be the independent neighborhood of node u

$V(L)$ denote the vertex set of subgraph L

r be the transmission range ratio, i.e. $r = \dfrac{r_{max}}{r_{min}}$

In the previous work [39], Thai *et al.* have provided the upper bound on the size of the independent neighborhood of any node u in DGB as follows:

Lemma 1. *[39] In a DGB, the size of $N_{ID}(u)$ is bounded by K, i.e, $|N_{ID}(u)| \leq K$ where $K = 5$ if $r = 1$, otherwise, $K = 10(\lceil \frac{\ln(r)}{\ln(2\cos(\frac{\pi}{5}))} \rceil + 1)$*

Note that when $r = 1$, a DGB is a UDG. Hence all the analysis in this work are also applied for a UDG.

4.1.3 Algorithm Description

For above problem, we will propose a general solution, where $1 \leq k \leq m + 1$, called Fault Tolerant Approximation Algorithm (FTAA). The main idea of FTAA is as follows. Given a network $G = (V, E)$, the algorithm consists of five main steps:

1.Use the CDSMIS (to be described later) algorithm to construct a 1-Connected m-Dominating Set (1-m-CDS) of G.

2.Compute all the k'-block in 1-m-CDS. Initially, $k' = 2$.

3.If there is more than one k'-block in 1-m-CDS, find the shortest path in the original graph that satisfies the two requirements: (i) the path can connect a k'-leaf block in 1-m-CDS to other portion of 1-m-CDS. (ii) the path does not contain any nodes in 1-m-CDS except the two end points. Then add all intermediate nodes in this path to 1-m-CDS.

4.Repeat step 3) until there is only one k'-block in 1-m-CDS.

5.Increase k' by 1 at each iteration and repeat Step 3) until C is k-m-CDS.

Based on an input of 1-Connected 1-Dominating Set (1-CDS), we first build a 1-m-CDS by calling the function CDSMIS. Then merge all the k'-blocks in 1-m-CDS into only one k'-block by adding extra nodes, where $k' = 2$ initially. Then, we increase k' by 1 and repeat the above operation until $k' = k$. The formal presentation of FTAA is shown in Algorithm 2.

As just mentioned earlier, we will employ a function called Connected Dominating Set by Maximal Independent Sets (CDSMIS) to build a 1-m-CDS. The main steps of CDSMIS is: (i) given a 1-CDS with a bounded size and diameter. (ii) iteratively add nodes into a CDS to make it a 1-m-CDS. As shown in Algorithm 1, the CDSMIS have two stages. At the first stage, we can use any 1-CDS with bounded size and diameter as the input of Algorithm 1 such that the 1-CDS includes an MIS I_1 of G. However, in order to make

the solution adjustable by the user, an (α, β)-CDS, to be introduced in Section 4.2.2, is preferred to be an input of Algorithm 1. Then, we remove I_1 from G at this stage.

At the second stage, we iteratively find an MIS I_i in the remaining graph $G-(I_1 \cup I_2 \cup ... \cup I_{i-1})$ by choosing a biggest transmission range and delete its neighbors until there is no nodes exists, then add I_i into the 1-CDS at current iteration. After running the second stage $(m-1)$ times, the resulting CDS is 1-m-CDS.

Algorithm 1 Connected Dominating Set by Maximal Independent Set Algorithm (CDSMIS) for 1-m-CDS

INPUT: An m-connected DGB $G = (V, E)$ and a 1-CDS C_{11} with bounded diameter and size, C_{11} must include an MIS I_1 of G
OUTPUT: A 1-m-CDS C_{1m} of G
$C_{1m} = C_{11}$.
Remove I_1 from the graph
for $i = 2$ to m **do**
    Construct an MIS I_i in $G - (I_1 \cup I_2 \cup ... \cup I_{i-1})$
    $C_{1m} = C_{1m} \cup I_i$
end for
Return C_{1m} where C_{1m} is the 1-m-CDS

The intuition of this algorithm is that a k-m-CDS is also a 1-CDS, thus by given a 1-CDS with bounded size and diameter, we do not introduce any unnecessary nodes. Moreover, we only add nodes that are necessary to make the k-connected and m-dominating portion larger. In total, the size and diameter of k-m-CDS can be bounded, we will prove this in the following sections.

4.1.4 Correctness of FTAA

In the first step, we employ CDSMIS to build a 1-m-CDS C_{1m} of G. Note that this guarantees C_{1m} is a m-dominating set. Therefore, to prove the correctness of FTAA algorithm, we need to prove C_{1m} is k-connected after the algorithm terminates and the algorithm runs in bounded time.

Lemma 2. *In line 8 (of FTAA), a k'-leaf block always exists when C_{1m} is not k'-m-CDS.*

Proof: By contradiction, suppose there is no k'-leaf block in C_{1m}. According to the definitions of separating set and k-leaf block, the removal of any $(k'-1)$ nodes will

Algorithm 2 Fault Tolerant Approximation Algorithm (FTAA)

1: INPUT: A k-connected graph $G = (V, E)$
2: OUTPUT: A k-m-CDS C_{km} with bounded diameter and size
3: Construct a 1-m-CDS C_{1m} by calling CDSMIS
4: Initialize $k' = 2$;
5: B = ComputekBlock(C_{1m}, k');/* B is a list of all k'-blocks in C_{1m} */
6: **while** $k' \leq k$ **do**
7: **while** B contains more than one k'-block **do**
8: L = findkLeafBlock(B, k'); /* L is one k'-leaf block*/
9: **for** each node $v \in V(L)$ and v is not a node in separating set **do**
10: **for** each node $u \in C - V(L)$ **do**
11: Construct G' from G by deleting all nodes in C_{1m} (except u and v) and all edges incident to those nodes;
12: **if** there exist at least one uv-path in G' **then**
13: P_{uv} = shortestPath (v, u, G'); /*P_{uv} is the shortest uv-path containing only those nodes not in C_{1m} as the intermediate nodes */
14: **end if**
15: $P = P \cup P_{uv}$
16: **end for**
17: **end for**
18: P_{ij} = the path with shortest length among all paths in P;
19: $C_{1m} = C_{1m} \cup$ intermediate nodes on P_{ij};
20: B = ComputekBlock(C_{1m}, k');
21: **end while**
22: $k' + +$
23: **end while**
24: Return C_{1m} where C_{1m} is a k-m-CDS C_{km}.

partition C_{1m}. Therefore, C_{1m} is at least k'-m-CDS, which contradicts to the fact that C_{1m} is not k'-m-CDS.

□

Lemma 3. *In line 18, the shortest path P_{ij} always exists.*

Proof: Consider C_{1m} such that C_{1m} is not k'-connected. To prove P_{ij} always exists, we prove that there always exists a path with only those nodes not in C_{1m} to connect a k'-leaf block to a k'-block of C_{1m}. This is true because G is k-connected and $k' \leq k$. If we delete the separating set of C_{1m}, there must exist a path P in the original graph G connecting the k'-leaf block to a k'-block. On the other hand, there are $(k' - 1)$ disjoint paths which

connect the k'-leaf block to other k'-blocks through the separating set with size of $(k'-1)$ in C_{1m}, thus all nodes except the two endpoints on P are not in C_{1m}.

□

Lemma 4. *[48] Two k'-blocks in $(k'-1)$-m-CDS share at most $(k'-1)$ nodes.*

When two k'-blocks of $(k'-1)$-m-CDS share $(k'-1)$ nodes, it is straight that the set of $(k'-1)$ nodes must be a separating set with the size of $(k'-1)$.

Lemma 5. *From line 7 to 21 (the inner while loop), a k'-leaf block is merged into a k'-block through path P_{ij} to form a larger k'-block without generating any new k'-block.*

Proof: Consider a subgraph H, which is composed of a k'-leaf block and a k'-block. At this step, H is $(k'-1)$-connected, with a separating set S. Let H' be a subgraph obtained from H by adding a new path P_{ij}. We argue that a separating set S' of H' must have size at least k', which means H' is at least a k'-block. There are two possibilities: (i) one or more nodes on P_{ij} exist in S'. (ii) no node on P_{ij} falls into S'.

For case (i), the way to separate H' with minimum size of separating set is to separate H. Because if we separate any node on P_{ij}, says x, from the remaining part of H', $|S'| \geq k'+1$, since x is dominated by at least k' different nodes in H and at least one node on P_{ij} exists in S'. If we separate H, from Lemma 4, we have $|S| \geq k'-1$. Thus, $|S'| \geq |S|+1 \geq k'-1+1 = k'$.

For case (ii), since no node on P_{ij} belongs to S' and assume y is any node in P_{ij}, if $N(y) \subseteq S'$, then $|S'| \geq k$, because at this time y is not in H, thus, y must have at least k' different neighbors in H. Otherwise, y and $N(y) - S'$ lie in a single component of $H' - S'$. Thus S' must separate H' and $|S'| \geq k'$. Therefore, H' is a k'-block.

□

From Lemma 5, at each iteration in the inner loop, the number of k'-block is decreased at least by one. Thus, when the inner loop terminates, C_{1m} must be k'-m-CDS. In the outer loop, we increase k' by 1 and construct a k'-m-CDS. Therefore, base on the induction method, we conclude that our obtained C_{1m} is a k-m-CDS at the end.

4.1.5 Theoretical Analysis

In this section, we first discuss the performance bound on size of C_{km} constructed by FTAA, then the analysis of the diameter of C_{km} is also proposed as well.

4.1.5.1 The analysis on size

It is easy to see that the union of 1-m-CDS and the number of nodes we will add in the 1-m-CDS so as to make it k-connected, will be the upper bound of k-m-CDS. Therefore, we will present the upper bounds of the two parts separately and add them together in the end.

The 1-m-CDS in CDSMIS is the union of m MISs and a set B, where B is a set of nodes that make I_1 connected. Note that the union of I_1 and B is an input of 1-CDS with bounded size and diameter. To analyze the approximation ratio of CDSMIS, we first compare the size of any MIS in G with the size of an optimal 1-m-CDS. Let I denotes an MIS of a remaining graph G at any step, C_{1m} be our solution obtained from CDSMIS, D_m^* be an optimal m-dominating set in G, and C_{1m}^*, and C_{11}^* be the optimal solutions of 1-m-CDS and 1-CDS on size respectively. We have the following lemma:

Lemma 6. *Let $G = (V, E)$ be any DGB with bounded transmission range ratio r, then $|I| \leq (\frac{K}{m} + 1)|DS_m^*|$*

Proof: Let us consider $I - DS_m^*$, there are two possibilities: (i) $I - DS_m^* = \emptyset$, that is $I \subseteq DS_m^*$, and (ii) $I - DS_m^* \neq \emptyset$.

Case (i): Because $I \subseteq DS_m^*$, we have: $|I| \leq |DS_m^*|$.

Case (ii): For all $u \in I - DS_m^*$, let $D_u = |DS_m^* \cap N(u)|$. As DS_m^* is an m-dominating set of G, $D_u \geq m$ for each $u \in I - DS_m^*$ and we have:

$$\sum_{u \in I - DS_m^*} D_u \geq m|I - DS_m^*|$$

For all $v \in DS_m^*$, let $d_v = |(I - DS_m^*) \cap N(v)|$. From Lemma 1, for all $v \in DS_m^*$ there are at most K independent nodes in its neighborhood and $d_v \leq K$. Therefore, we have:

$$K|DS_m^*| \geq \sum_{v \in DS_m^*} d_v$$

However, note that:

$$\begin{aligned} \textstyle\sum_{u \in I - DS_m^*} D_u &= |(u,v) \in E \mid u \in I - DS_m^*, v \in DS_m^*| \\ &= \textstyle\sum_{v \in DS_m^*} d_v \end{aligned}$$

From the above, we have:

$$\begin{aligned} m|I - DS_m^*| &\leq \textstyle\sum_{u \in I - DS_m^*} D_u \\ &= \textstyle\sum_{v \in DS_m^*} d_v \leq K|DS_m^*| \end{aligned}$$

Therefore,

$$m|I - DS_m^*| \leq K|DS_m^*|$$

Thus it follows that:

$$|I| \leq (\frac{K}{m} + 1)|DS_m^*|$$

Therefore in two cases (i) and (ii), we conclude that: $|I| \leq (\frac{K}{m} + 1)|DS_m^*|$. □

Lemma 7. *The number of nodes added in m MISs is at most $(K + m)|C_{1m}^*|$.*

Proof: The nodes added in m MISs are $|I_1 \cup I_2 \cup ... \cup I_m|$. Clearly, $|I_i| \leq |I|$ and from Lemma 6, we have:

$$\begin{aligned} |I_1 \cup I_2 \cup ... \cup I_m| &\leq m|I| \\ &\leq m(\tfrac{K}{m} + 1)|DS_m^*| \\ &\leq (K + m)|DS_m^*| \\ &\leq (K + m)|C_m^*| \end{aligned}$$

□

Suppose we already have an input of 1-CDS with α-approximation ratio on size. Then, we will have the following important conclusion.

Theorem 1. *If given an input of 1-CDS with α-approximation on its size, CDSMIS can produce a 1-m-CDS with a performance bound of $(\alpha + K + m - 1)$.*

Proof: From Lemma 7, we have:

$$\begin{aligned} |C_{1m}| &= |B| + |I_1 \cup I_2 \cup ... \cup I_m| \\ &\leq \alpha|C^*_{11}| + (K + m - 1)|C^*_{1m}| \\ &\leq (\alpha + K + m - 1)|C^*_{1m}| \end{aligned}$$

□

The above theorem concludes that if the transmission range ratio r is bounded and the input of 1-CDS has constant approximation on size, then CDSMIS can construct a 1-m-CDS with approximation factor of $O(1)$.

In the following context, we will focus on the discussion on how to calculate the upper bound of the number of nodes we have to add in 1-m-CDS.

From Lemma 4, we have known that if two k'-blocks share at most $(k' - 1)$ nodes, the set of the shared $(k' - 1)$ nodes must be a separating set with the size of $(k' - 1)$. Now, we prove the following lemma:

Lemma 8. *A k'-block and a k'-leaf block have only one separating set with the size of $(k' - 1)$.*

Proof: By the contradiction method, assume that we have another separating set S_2 with the size of $(k' - 1)$ besides the separating set S_1, which is exactly the set of all $(k' - 1)$ shared nodes by the two blocks B_1 and B_2. When we delete $(k' - 1)$ nodes from S_2, what remains is connected, because we retain a path for any two nodes in $B_1 \cup B_2$. Thus, we must delete at least k' nodes in S_2 to cause partition of the subgraph composed by B_1 and B_2. So, $|S_2| \geq k'$, contradicting our assumption that $|S_2| = (k' - 1)$.

□

As mentioned before, the value of k is in the range from 1 to $m+1$. Then, given any positive number for k and m, where $k \leq m+1$, we will divide our discussion into two case: (1) $k \leq m$ and, (2) $k = m+1$. For Case (1):

Lemma 9. *When $k \leq m$, at most 2 new nodes are added into C_{1m} at each augmenting step. That is, the shortest path P_{ij} has at most 2 nodes not in C_{1m}.*

Proof: Each node in P_{ij} must be m-dominated because we first build a 1-m-CDS at the first step. From lemma 8, we have only one separating set S with size of $(k'-1)$. Therefore, each node on P_{ij} must be dominated by at least 1 node in C_{1m} but not in S. Hence, P_{ij} has at most 2 nodes not in C_{1m}.

□

Lemma 10. *We employ at most $|C_{1m}-1|$ augmenting steps in each iteration to change C_{1m} from k'-connected to $k'+1$-connected.*

Proof: Suppose the current C_{1m} is k'-connected and FTAA has to run *at least* $|C_{1m}|$ augmenting steps to make C_{1m} $k'+1$-connected. We have to use at most $|C_{1m}-1|$ steps to make all nodes in C_{1m} to be $k'+1$-connected. That results in the algorithm will employ at least one step to connect a node not in C_{1m} to a $k'+1$-leaf block to make them $k'+1$-connected. However, by the definition of 1-m-CDS, each node not in C_{1m} must be dominated by at least m nodes in C_{1m}, where $k'+1 \leq m$. Thus, each node not in C_{1m} has been m-connected with the rest of the graph and the algorithm will not run the extra augmenting steps to connect the nodes not in C_{1m} with the rest of graph. Therefore, we only need at most $|C_{1m}-1|$ augmenting steps in each iteration. □

Theorem 2. *When $k \leq m$, the FTAA algorithm produces a k-m-CDS with size bounded by $(\alpha+K+m-1)(2k-1)$ in a DGB.*

Proof: Let C^*_{km} be an optimal solution for k-m-CDS and C_{km} be the solution obtained from our FTAA algorithm. Note that the construction of C_{km} can be divided into two parts: building a 1-m-CDS and adding some paths P_{ij} with at most 2 nodes in it. Since we already have the bound on 1-m-CDS (theorem 1), we just need to analyze the number

of intermediate nodes we will add into C_{1m} in the second part. From Lemma 9 and Lemma 10, at most 2 nodes are added in each step and we employ at most $|C_{1m}| - 1$ steps to change C_{1m} from $(k'-1)$-m-CDS to k'-m-CDS. In order to construct a k-m-CDS, we have to repeat this procedure $(k-1)$ times. Therefore, we have:

$$\begin{aligned} |C_{km}| &\leq |C_{1m}| + 2(k-1)(|C_{1m}| - 1) \\ &\leq (2k-1)|C_{1m}| \\ &\leq (\alpha + K + m - 1)(2k-1)|C^*_{1m}| \\ &\leq (\alpha + K + m - 1)(2k-1)|C^*_{km}| \end{aligned}$$

□

Now, we will consider the Case 2 in details.

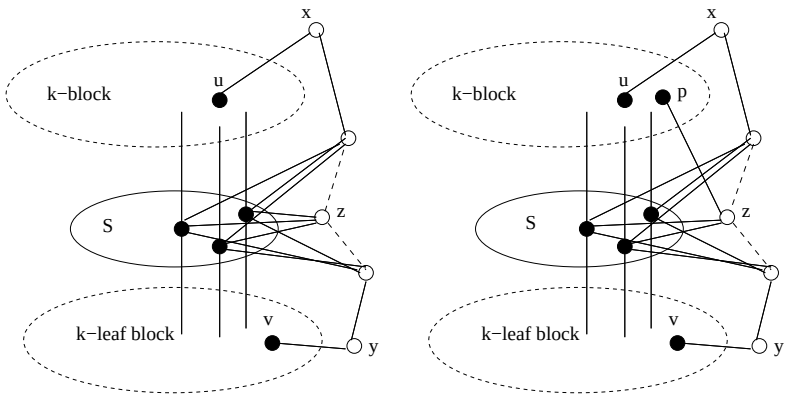
In Lemma 9, when $k \leq m$, we prove that at most 2 new nodes are added into C_{1m} at each augmenting step. However, when $k = m + 1$, at each augmenting step in the last outer iteration, the number of added nodes is quite different.

In Case 2, m is one smaller than k. Let $C_{k'm}$ be the k'-m-CDS at the iteration $k' = k - 1$ (an iteration right before the last iteration, at this time $k' = m$). Thus, we first build a 1-m-CDS, and then run FTAA to make become a k'-m-CDS. Note that each node not in C_{km} will be dominated by at least $(k-1)$ different nodes in C_{km}. Therefore, Lemma 9 may not hold at the last iteration where $k' = k$. Instead, we have the following lemma:

Lemma 11. *At the last iteration when $k' = k$, we add at most $2(K+1)$ nodes to $C_{k'm}$ at each augmenting step in order to make it become a k-m-CDS, where $k = m + 1$.*

Proof: Suppose we mark the nodes in $C_{k'm}$ with BLACK and the remaining nodes with GRAY. Suppose L is a k-leaf block of $C_{k'm}$ and S is the separating set. Suppose nodes u and v, where $u \in V(L)$ and $v \in C_{k'm} - V(L)$, are the two black nodes connected by the shortest path without any black nodes, there are three possibilities that nodes u and v are connected, that is u and v are connected by one connector, two connector, and more than two connectors. Fig. 4-1A illustrates the scenario of existing more than two connectors in the shortest path.

We claim that if the shortest path between u and v, called P_{uv} has more than two intermediate nodes, all intermediate nodes in the shortest path except x and y must be dominated by all nodes in separating set S. (See Fig. 4-1A) This is true because: the $C_{k'm}$ is $(k-1)$-dominating, suppose P_{uv} is $u, x, ..., y, v$ and one of the intermediate nodes, let's say node z is dominated by one node in separating set S (as illustrated in See Fig. 4-1B), z must have another black node neighbor p or else z is not dominated by $(k-1)$ different nodes in the $C_{k'm}$, which is contrary to the fact that we build 1-$(k-1)$-CDS at first. If so, the path between pu or pv has a shorter distance than P_{uv}, which contradicts that P_{uv} has the shortest distance.



A Scenarios for more than two connectors B Path pv is smaller than uv

Figure 4-1. All intermediate nodes are neighbors of the nodes in separating set

We show that there exists a path connecting a k-leaf block to another k-block with a limited number of intermediate nodes. The position of nodes u and v has four possibilities: (i) nodes u and v are both neighbors of one node, says s, in separating set S. (ii) node u is the neighbor of node s in separating set S, but node v is not. (iii) node v is the neighbor of node s in separating set S, but node u is not. iv) neither node u nor v are neighbors of any node in separating set S.

Case (i) is illustrated in Fig.4-2. From Lemma 1, we know that the size of $N_{ID}(u)$ is bounded by K. We can divide the neighborhood of node s into K subregions by using the same dividing strategy proposed in [39]. Therefore, in each subregion, there is at most one

independent neighbor of node s. The dash circle is the transmission range of nodes u and v. All the interconnecting nodes are marked with a small white circle. To maximize the number of interconnecting nodes, we let the transmission range of nodes u and v be r_{min} and the distance between u and v is slightly larger than r_{min}, so u is not a neighbor of v, and vice versa. Note that if two nodes fall in the same subregion, they will be connected to be a path because each subregion can have at most one independent neighbor of node s. Therefore, there are at most 2 nodes to form a shortest path in each subregion. (If we have 3 or more nodes in each subregion, the path must not be the shortest.) Since we divide the neighborhood of s into K subregions and with some algebraic step, we can figure out there are at most $K(\frac{2\pi-\beta}{\pi})$, where $\beta = 2arcsin\frac{r_{min}}{r_{max}} = 2arcsin\frac{1}{2r}$.

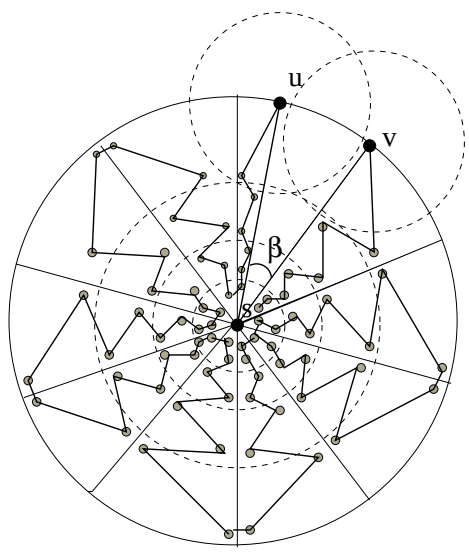


Figure 4-2. s is node in separating set, node u and v are both neighbors of node s

Case (ii) is illustrated in Fig. 4-3. Since s is the node in separating set, there must exist another black node, called a in the graph, that is a neighbor of s, otherwise the CDS is not connected anymore. Since P_{uv} has the shortest length, then there could not exist interconnecting nodes in region A and B, otherwise, there path P_{ua} has a shorter a length than P_{uv}. Same as Case i), at most 2 nodes will form a shortest path in each subregion. There might be another interconnecting node which is a neighbor of v but not

of s, called y in Fig. 4-3. Thus, if only u is in the transmission range of s, there are at most $K(\frac{2\pi-\beta-\delta}{\pi})+1$ interconnecting nodes, where $\beta=\delta=2arcsin\frac{1}{2r}$.

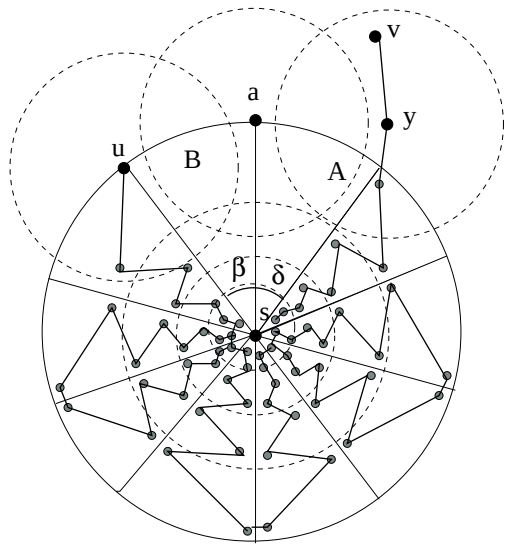


Figure 4-3. s is node in separating set, u is in its k-leaf block is a neighbor of s, while v is not in the k-leaf block and v is not a neighbor of s

Similar to case (ii), case (iii) has the same number of maximal interconnecting nodes.

Case (iv) is illustrated in Fig. 4-4. Since s is the node in separating set and neither u nor v is in its transmission range, there must exist two other black nodes, called a and b in Fig. 4-4, that are two neighbors of s. Since path P_{uv} has the shortest length, then there could not exist interconnecting nodes in region B, A and C, otherwise either path P_{ab}, P_{ub} or P_{va} has the shorter length than P_{uv}. Again, at most 2 nodes will form a shortest path in each subregion. There might be two other interconnecting nodes which are a neighbor of u and v but not of s respectively, called x and y in Fig. 4-4. Thus if neither u nor v is in the transmission range of s, there are at most $K(\frac{2\pi-\beta-2\delta}{\pi})+2$ interconnecting nodes between u and v.

In summary, we prove that for all cases, at most $2(K+1)$ interconnecting nodes are necessary to connect a k-leaf block to other k-blocks.

Theorem 3. *The constructed k-m-CDS is of size at most $(2k+2K-1)(K+\ln K+k+1)|C^*_{km}|$ in DGB, where $k=m+1$ and C^*_{km} is an optimal solution of k-m-CDS.*

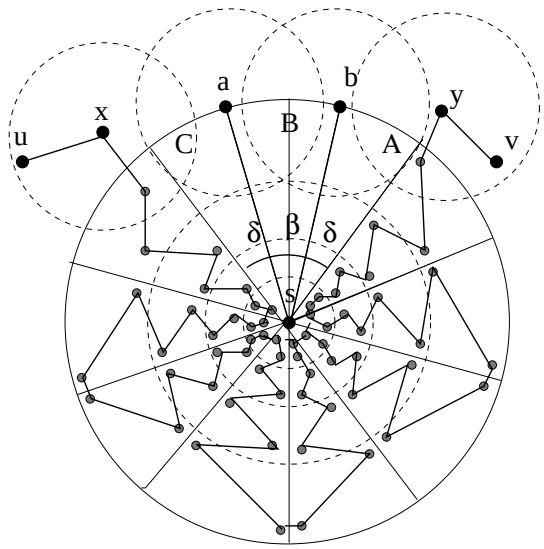


Figure 4-4. s is node in separating set, u is in its k-leaf block while v is not in its k-leaf block and neither u and v is are neighbors of s

Proof: Again, let C_{km} denote our obtained solution and C_{1m} be the 1-m-CDS obtained from CDSMIS. We have:

$$\begin{aligned}
|C_{km}| &\leq |C_{1m}| + 2(k-2)(|C_{1m}|-1) \\
&+2(K+1)(|C_{1m}|-1) \\
&\leq (2k+2K-1)|C_{1m}| \\
&\leq (2k+2K-1)(\alpha+K+m-1)|C^*_{1m}| \\
&\leq (2k+2m+1)(\alpha+K+m-1)|C^*_{km}|
\end{aligned}$$

□

From the discussion on the two cases, we can conclude that if the transmission range ratio r is bounded and the input of 1-CDS has constant approximation on size, then FTAA has an approximation factor of $O(1)$ on size.

4.1.5.2 The analysis on diameter

Lemma 12. *$D^* \leq D^*_{km}$, where D^*_{km} and D^* denote the optimal diameters of k-m-CDS and 1-CDS respectively.*

Proof: It is straightforward that $D^* \leq D^*_{km}$, since a k-m-CDS is also a 1-CDS. Suppose $D^* > D^*_{km}$, D^* is not the optimal diameter of 1-CDS. Thus, $D^* \leq D^*_{km}$. □

Lemma 13. $d(C_{1m}) \leq d(C_{11}) + 2$.

Proof: Since each node not in C_{11} is dominated by at least one node in C_{11}. Therefore, when we add more nodes into C_{11} in order to make it to be C_{1m}, we only increase $d(C_{11})$ by at most 2 hops. □

Lemma 14. $d(C_{km}) \leq d(C_{1m}) + 2$.

Proof: Suppose two nodes u and v are in C_{km}. The position of node u and v has three possibilities: (1) $u, v \in C_{1m}$. (2) $u \in C_{km} - C_{1m}$, $v \in C_{1m}$. (3) $u, v \in C_{km} - C_{1m}$. For case (1), the number of hops between u and v is bounded by $d(C_{1m})$. For case (2), u must be dominated by a node in C_{1m}. Therefore, u is only one hop away from its dominator in C_{1m} and the number of hops between u and v is bounded by $d(C_{1m}) + 1$. For case (3), u and v are dominated by different nodes in C_{1m}. However, u and v are only one hop away from their dominators. Thus, the number of hops between u and v is bounded by $d(C_{1m}) + 2$. □

Theorem 4. *If the diameter of input 1-CDS is bounded by βD^*, the approximation ratio of the constructed k-m-CDS on diameter is $\beta D^*_{km} + 4$.*

Proof: From Lemma 12, 13 and 14, we have the following inequality:

$$d(C_{km}) \leq d(C_{1m}) + 2 \leq d(C_{11}) + 2 + 2 \leq \beta D^* + 4 \leq \beta D^*_{km} + 4$$

□

In conclusion, If given an (α, β)-CDS as the input, where α is the approximation ratio of 1-CDS on size and β is that on diameter, then the k-m-CDS obtained from FTAA has constant approximation factors of $O(\alpha(K + m))$ on size and $O(\beta)$ on diameter.

4.1.5.3 Time complexity

Theorem 5. *Suppose n is the number of nodes in original graph, the time complexity of FTAA is $O(kn^3)$.*

Proof: Time complexity of constructing a 1-m-CDS is easy to analyze, since computing an MIS takes $O(n)$ time [32]. Therefore, the time complexity of CDSMIS is dominated by finding $m - 1$ MISs, which is $O(mn)$.

In the second step, the way we compute k'-block is based on the distributed strategy. Each node in C_{1m} collects multi-hop-away neighborhood information till a k'-block can be built. This strategy is reasonable, although we need multi-hop-away neighborhood information. If at least one k'-block exists in C_{1m}, we can always find it. The worst case is that we need all nodes' information in C_{1m}. Then, all the nodes who found the same k'-block identify each other by broadcasting and return the k'-block as a whole. Therefore, the complexity for second step is $O(n^2)$.

The time complexity of third step is dominated by the $ShortestPath$ function, which runs in $O(n^2)$. The second and third step are executed at most $n-1$ (the maximum number of k'-blocks) times in the fourth step. Therefore, the time complexity of fourth step is $O(n^3)$.

The fifth step repeats the second, third and fourth step $k-1$ times, Therefore, the total time complexity of FTAA is $O(kn^3)$. □

Now, we have completed the analysis for FTAA. If positive k, m and α are given, we will get the constant performance guarantee on size.

4.2 Basic Model

In this section, we introduce two algorithms for 1-CDS (also known as CDS) with bounded size and diameter. One is called Basic Distributed Approximation Algorithm (BDAA) and another is called Progressive Distributed Approximation Algorithm (PDAA). The two algorithms could be used as an input of FTAA. The benefit of BDAA is the low time complexity on constructing CDS, while PDAA performs well on optimizing the size and diameter of CDS.

4.2.1 Problem Definition

The CDS problem with bounded diameter could be formally defined as follows:

Definition 2. ***CDS Problem with Bounded Diameter:*** *Given a DGB $G = (V, E)$ representing a network, construct a CDS C such that: (1) each node in V is either in C*

or has at least one neighbor in C, the subgraph induced by C, i.e., $G[C]$, is connected, (2) minimize the diameter of C, and (3) minimize the size of C.

To construct a CDS, we often employ an Maximal Independent Set (MIS) which is also a subset of all the nodes in the network. The nodes in MIS are pairwise nonadjacent and no more nodes can be added to preserve this property. Therefore, each node which not in MIS is adjacent to at least one node in MIS. Therefore, an MIS is indeed a DS. If the nodes in MIS are connected by adding more nodes to the MIS, a CDS can be constructed.

4.2.2 Basic Distributed Approximation Algorithm

4.2.2.1 Algorithm description

The main idea of BDAA is as follows:

1.Using Wan's distributed MIS algorithm [32] to construct an MIS. Color all the nodes in MIS black.

2.Randomly choose a black node as the root, and assign a level to each node, which is based on the number of hops away from the root.

3.Connect the nodes in MIS from low level to high level with minimum number of hops.

One existing distributed algorithms for MIS [32] is executed to obtain a DS. The obtained MIS satisfies the following lemma:

Lemma 15. *Any pair of complementary subsets of a constructed MIS has a distance of exactly two hops. [32]*

In order to implement this algorithm in distributed manner, each node maintains a local status which is initialized to *unexplored* and set to *explored* after proceeded by the algorithm. Each node also maintains a local variable which stores the ID of message sender and is initially empty.

The following operations for connecting the nodes in MIS with minimum number of hops may be conducted as described in the algorithm below:

Algorithm 3 Basic Distributed Approximation Algorithm (BDAA)

INPUT: A connected DGB $G = (V, E)$ and an MIS computed by Wan's algorithm [32]
OUTPUT: A CDS T_{CDS} with minimum diameter
Each node maintains a unique node ID and a status of *unexplored* initially
Color all nodes in MIS black and color every node adjacent to a black node in grey
Randomly choose a root r in MIS and set r to *explored*. Each node y is assigned a level k such that $k = HopCount(r, y)$, where $0 \leq k \leq k^*$. Suppose k^* is the maximum value of k.
r broadcasts $EXPLORE$ messages to its neighbors at level 1, where r is at level 0
Upon receiving $EXPLORE$ messages, an *unexplored* grey node z at level i sets itself *explored* and check if it has a black neighbor y at level i or $i + 1$, if true, its color is set **blue**, the ID of the message sender is stored, and sends $EXPLORE$ messages to its black neighbors at level i and $i + 1$, if possible.
Upon receiving an $EXPLORE$ message, an *unexplored* black node y at level $i(i \geq 2)$ sets itself *explored* and the ID of the message sender is stored, then it employs the stored node IDs to trace a 2-hops-away black node x at level $i - 2$ or level $i - 1$ via a blue node z, add the path (x, z, y) into T_{CDS} and then sends $EXPLORE$ messages to its grey neighbors at level i and $i + 1$, if possible.
The algorithm stops until there is no node changed from grey to blue.
The union of black and blue nodes is a CDS.

4.2.2.2 Theoretical analysis

Note that if a black node x at level $(i - 2)$ do not have a 2-hops-away black node y at level i, then x must have a 2-hops-away black node y at level $(i - 1)$, since Lemma 15 holds. Therefore, for each black node y we color exactly one grey node in blue to make x and y connected. So, the number of nodes changed from grey to blue is exactly $|MIS| - 1$.

Note that the CDS constructed by BDAA is the union of MIS and a set of blue nodes that connects MIS. Thus, We have the following theorem:

Theorem 6. *Denote T_{CDS} as our solution obtained from BDAA, then $|T_{CDS}| \leq 2K|CDS^*| - 1$ and $d(T_{CDS}) \leq 4D^* + 4$ in a DGB.*

Proof: It is known that for an MIS in a DGB, $|I| \leq K|CDS^*|$ [39]. From the observation that the number of nodes we have to add to connect the nodes in MIS is exactly $|I| - 1$, thus, $|T_{CDS}| \leq 2|I| - 1 \leq 2K|CDS^*| - 1$. For diameter of CDS, every black node at level k is away from r within a distance at most $2k$ hops. Suppose G has diameter D, then $D \geq k^*$, and the minimum diameter of CDS is at least $D - 2$.

In the worst case, two nodes in T_{CDS} at k^* level are separated by $2 * 2k^*$ hops since each node is away from r at most $2k^*$ hops. In addition, we note that no black node exists at level 1, the black nodes in level 2 can connect with r with 2 hops. Therefore, $d(T_{CDS}) \leq 2 * 2(k^* - 2) + 4 \leq 4D^* + 4$

□

4.2.2.3 Time complexity

Theorem 7. *The BDAA has $O(n)$ time complexity and $O(n \log n)$ message complexity.*

Proof: Construction of an MIS takes $O(n)$ time complexity and sends $O(n \log n)$ messages [32]. After that, we use linear message and take at most linear time to connect the nodes in MIS. Overall, BDAA has $O(n)$ time complexity and $O(n \log n)$ message complexity.

□

4.2.3 Progressive Distributed Approximation Algorithm

In this section, we introduce (α, β)-CDS into our model to be the input of FTAA. Also, it can solve the CDS problem with bounded diameter. It approximately satisfies the size constraint and the diameter constraint by constructing a CDS. As we intent to balance the size and diameter, the definition of (α, β)-CDS is given in wireless networks as follows:

Definition 3. (α, β)***-CDS:*** *For a fixed $\alpha > 1$ and $\beta \geq 1$, a CDS C of G meeting the following two requirements is called an (α, β)-CDS.*

(Size) The size of C is at most α times the minimum CDS size.

(Diameter) For any pair of vertex u and v in C, $d(C)$ is at most β times the minimum diameter of CDS plus a constant number.

In (α, β)-CDS, β is an user-defined input, and usually α is a function of β. Therefore, the value of α depends on the user-defined input β. In the following, we will describe how to generate an (α, β)-CDS and study the tradeoff between the size and diameter.

Since an (α, β)-CDS might be used as an input of FTAA, the tradeoff is still preserved for k-m-CDS.

4.2.3.1 Algorithm description

The general idea of our PDAA is as follows.

1.Construct a CDS T_{CDS} root at r by using BDAA. Root r should locate at the *center* of network, which is the mid-point of the longest shortest path between two nodes in graph G.

2.Construct a Shortest Path Tree (SPT) T_{SPT} rooted at r, which only includes all the shortest paths from r to every other node in T_{CDS}.

3.Traverse T_{CDS} in a depth-first manner. When visiting a node u, if the number of hops from r to u in T_{CDS} is larger than a user-defined threshold β times the number of hops from r to u in T_{SPT}, then a new path from r to u in T_{SPT} is added in T_{CDS}.

If we denote $D_{CDS}(u, v)$ as the number of hops from u to v in T_{CDS} and $D_{SPT}(u, v)$ as the number of hops from u to v in T_{SPT}. The details of PDAA is as follows:

Root Selection and CDS Tree Construction: With Dijkstra's algorithm [46], which is used to solve the single-source shortest-paths problem, and a global variable, which stores the maximum shortest path in the graph G, we could find the mid-point of the longest shortest path by running Dijkstra's algorithm on each node and regard the root r as the mid-point. While constructing T_{CDS} rooted at r by using BDAA, each node u needs to maintain a pointer $\pi[u]$ for its parent on the tree T_{CDS} and an upper bound $d[u]$ for the number of hops to r. We use the $INITIALIZE$ and $RELAX$ algorithms in [49] to initialize and maintain both of these attributes.

Shortest Path Tree Construction: T_{SPT} rooted at r is constructed by using Dijkstra's algorithm. It only contains all the shortest paths from the root r to every other node in T_{CDS}.

Depth First Search (DFS): Traverse the T_{CDS} in a DFS manner beginning from the root r along the paths from r to all the other nodes in T_{CDS}. When node u is reached for

Algorithm 4 Progressive Distributed Approximation Algorithm (PDAA)

PDAA(β)

```
Locate the center of network and choose r at the center.
Build a T_CDS based on r
Use Dijkstra's algorithm to construct an SPT T_SPT
C = FIND (T_CDS, T_SPT, r, β)
return C
```

FIND (T_{CDS}, T_{SPT}, r, β)

```
INITIALIZE (T_CDS, r)
DFS (r)
return a desired CDS C
```

INITIALIZE (G, r)

```
for each vertex v ∈ T_CDS do
    d[v] ← ∞
    π[v] ← NIL
end for
d[r] ← 0
```

RELAX (u,v)

```
if d[v] > d[u] + D_CDS(u, v) then
    d[v] = d[u] + D_CDS(u, v)
    π[v] ← u
end if
```

DFS (u)

```
if d[u] > β D_SPT(r, u) then
    ADD-PATH (u)
end if
for each child v of u in MSA do
    RELAX (u, v)
    DFS (v)
    RELAX (v, u)
end for
```

ADD-PATH (v)

```
if d[v] > D_SPT(r, v) and parent_SPT(v) != NIL then
    ADD-PATH (parent_SPT(v))
    RELAX (parent_SPT(v), v)
end if
```

the first time, if $d[u]$ is greater than $\beta * D_{SPT}(r, u)$, then the shortest P_{ru} in T_{SPT} is added to T_{CDS} and $d[u]$ and $\pi[u]$ are updated. After this, node u's parent v needs to be checked if the updated path from r to u will result in reducing the number of hops from r to v. If so, then v's parent will be checked and so on until the root r is reached.

With the execution of BDAA, distributed SPT (dSPT) (e.g.[50]), and distributed DFS (dPFS) (e.g.[51]), T_{CDS}, T_{SPT} and a DFS traversal order could be achieved. In this way, with a *Manager* (e.g. root node), PDAA could be easily initiated and terminated according to the details illustrated in Algorithm 4.

To evaluate the correctness of the PDAA, we examine whether the two constraints in the definition has been satisfied. Taking β as an user-defined input, we derive a relationship between α and β, which shows the relationship between the size of the constructed CDS and the optimal solution of CDS on size. We also analyze the time complexity of the PDAA.

4.2.3.2 Theoretical analysis

Define $w(T_{CDS})$ as the total weight of T_{CDS} in G, where we assume each edge has been assigned the unit weight of 1. Then $D_{SPT}(u, v)$ and $D_{CDS}(u, v)$ are equal to the weight of $T_{SPT}(u, v)$ and $T_{CDS}(u, v)$ respectively. Another observation is that $|T_{CDS}| = w(T_{CDS}) + 1$, since the number of node in a tree equals to the total number of edges, which also equals to $w(T_{CDS})$ plus 1. Meanwhile, as we mentioned before, the lower bound of minimum diameter of CDS is $D - 2$. Actually, the upper bound for the minimum diameter of CDS is D, i.e., all the nodes in G are in CDS, therefore, $D^* = D$.

Due to the specific structures of CDS, we will classify the following proofs into two cases. Case (1): the diameter of SPT T rooted at r that spans all nodes in G is equal to D and all other situations are classified into Case (2).

Lemma 16. *For any pair of nodes u and v in C, the number of hops between u and v is at most β times $(D^* + 2)$, when $d(T) = D$.*

Proof: When a vertex v is visited, if $d[v] > \beta D_{SPT}(r, v)$, then shortest path between r and u is added into T_{CDS} by calling $ADD - PATH$. Also, we know that the maximum value for $D_{SPT}(r, v)$ is the height h of T_{SPT}, we will prove that $2h \leq D^* + 2$ in the following. After v is visited, $d[v]$ is at most $\beta D_{SPT}(r, v)$, which is less or equal to βh and subsequently never increases. For u, the same analysis can also be applied. Therefore, the total number hops between v and u in C is at most $2\beta h$, therefore at most $\beta(D^* + 2)$.

Now, we prove that $2h \leq D^* + 2$. First, it is easy to see that $2h \leq d(T)$ and $d(T) = D$. Then we have the following:

$$2h - 2 \leq d(T) - 2 \leq D - 2 \leq D^*$$

Therefore, we prove that $2h \leq D^* + 2$.

□

Lemma 17. *In case (2), for any pair of nodes u and v in C, the number of hops between u and v is at most 2β times $(D^* + 1)$.*

Proof: If $d(T) \neq D$, the worst case is that $d(T) = 2D^*$. A simple example to illustrate that is a *ring*, the degree of each node in the ring is only 2 and all the nodes in G are included in CDS, see Fig. 4-5. Therefore, $h \leq D^* + 1$, then the maximum number of hops between u and v is at most $2h$, that is $2\beta(D^* + 1)$. □

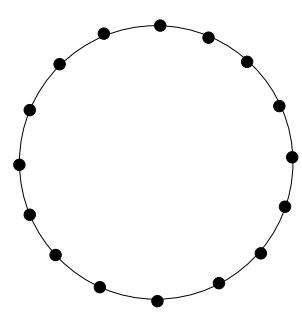


Figure 4-5. All the nodes in the ring are a CDS with diameter of 8

In real wireless network, case (2) rarely happens, since it requires all the hosts (nodes) are uniformly deployed, such as a ring. However, in most cases, they are deployed

randomly. Therefore, the diameter of CDS returned by PDAA is bounded by $\beta(D^* + 2)$ in most cases.

Lemma 18. *The total number of nodes on the added shortest paths is at most* $\frac{(5-\beta)K}{\beta-1}|C_{11}^*|+3$.

Proof: Let $v_0 = r$ and $v_1, v_2.....v_k$ be the vertices that caused shortest path to be added during the traversal, in the order they were encountered. When the shortest path from r to $v_i (i \geq 1)$ was added, the number of hops of the added path was $D_{SPT}(r, v_i)$. Also, the nodes on the path to v_i has been relaxed in order, so that $d[v_i] \leq D_{SPT}(r, v_{i-1}) + D_{CDS}(v_{i-1}, v_i)$. The shortest path to v_i was added because $\beta D_{SPT}(r, v_i) < d[v_i]$. Combining the inequalities,

$$\beta D_{SPT}(r, v_i) \leq D_{SPT}(r, v_{i-1}) + D_{CDS}(v_{i-1}, v_i)$$

Summing over i bounds from 1 to k, the number of hops of the added paths:

$$\beta \sum_{i=1}^{k} D_{SPT}(r, v_i) \leq \sum_{i=1}^{k} (D_{SPT}(r, v_{i-1}) + D_{CDS}(v_{i-1}, v_i))$$

and therefore

$$(\beta - 1) \sum_{i=1}^{k} D_{SPT}(r, v_i) \leq \sum_{i=1}^{k} D_{CDS}(v_{i-1}, v_i)$$

The DFS traversal traverses each edge exactly twice, and hence the sum on the right-hand side is at most twice $w(T_{CDS})$, since one hop corresponds to a unit weight of 1, i.e.,

$$\sum_{i=1}^{k} D_{CDS}(v_{i-1}, v_i) \leq 2w(T_{CDS})$$

We note that the number of new nodes on the added shortest path is exactly equal to $D_{SPT}(r, v_i) - 1$ and $|T_{CDS}| = w(T_{CDS}) + 1$. Therefore,

$$(\beta-1)\sum_{i=1}^{k}(D_{SPT}-1)(r,v_i)+k(\beta-1)\leq 2(w(T_{CDS})+1)-2$$

$$(\beta-1)\sum_{i=1}^{k}(D_{SPT}-1)(r,v_i)+k(\beta-1)\leq 2|T_{CDS}|-2$$

Here, we intend to maximize k in order to have a tighter bound on $\sum_{i=1}^{k}(D_{SPT}-1)(r,v_i)$, which is the total number of new nodes on the added shortest paths, Let denote $\sum_{i=1}^{k}(D_{SPT}-1)(r,v_i)$ as P_{size} for clear representation.

Intuitively, k is at most $|I|$ since all black nodes in MIS of T_{CDS} may cause shortest paths to be added during the traversal. However, the root r and at least two black nodes at level 2 will not be counted in k. Therefore, k is at most $|I|-3$.

$$\begin{aligned}(\beta-1)P_{size} &\leq 2|T_{CDS}|-2-(|I|-3)(\beta-1)\\ &\leq 4|I|-4-(|I|-3)(\beta-1)\\ &\leq (5-\beta)|I|+3\beta-7\end{aligned}$$

Since $|I|\leq K|C_{11}^{*}|$ [39], we have:

$$P_{size}\leq \frac{(5-\beta)K}{\beta-1}|C_{11}^{*}|+3$$

□

Theorem 8. *Given the value of β, the approximation ratio α of CDS on size is $\frac{(\beta+3)K}{\beta-1}$.*

Proof: From the above analysis, C is the union of T_{CDS} and the added shortest paths. Therefore, combining the Theorem 6 and Lemma 18,

$$\begin{aligned}|C| &= |T_{CDS}|+P_{size}\\ &\leq 2K|C_{11}^{*}|-1+\tfrac{(5-\beta)K}{\beta-1}|C_{11}^{*}|+3\\ &\leq \tfrac{(\beta+3)K}{\beta-1}|C_{11}^{*}|+2\end{aligned}$$

□

4.2.3.3 Time complexity

Theorem 9. *The time complexity of the PDAA algorithm is $O(n^2)$, and the message complexity of the PDAA algorithm is $O(n^2)$.*

Proof: From Theorem 7, the time complexity and message complexity for BDAA are $O(n)$ and $O(n \log n)$ respectively and dSPT and dDFS run at most $O(n^2)$ time complexity and send $O(n^2)$ messages [50] [51]. Now, we analyze the procedure of finding the center of network. The dSPT is executed at each node x simultaneously, after that, x needs to broadcast the longest path in SPT rooted at x and compare it with the longest paths returned by other nodes. Therefore, this procedure needs $O(n^2)$ time complexity and $O(n^2)$ message complexity. Since all other operation only take at most $O(n)$ time complexity and $O(n)$ message complexity, the overall message complexity and time complexity of PDAA are $O(n^2)$ and $O(n^2)$.

□

CHAPTER 5
SIMULATION RESULTS

In this section, we conduct the simulation experiments to measure the diameter and size of CDS constructed by our proposed algorithms. Moreover, we are interested in comparing the CDSs returned by CDS-BD [47] and PDAA. Since the running time for PDAA and BDAA has been discussed in Section 4.2, we also would like to verify the running time of the two algorithms in practice. In addition, we do various experiments by adjusting the user-defined parameter β in PDAA, in order to see how the CDS size and diameter could be balanced. At last, we evaluate the performance of FTAA (Algorithm 2) by comparing to PDAA so that the tradeoff between the three factors could be systematically discovered.

5.1 Network Parameters

We simulate the network with the following tunable parameters that may affect the algorithm performance:

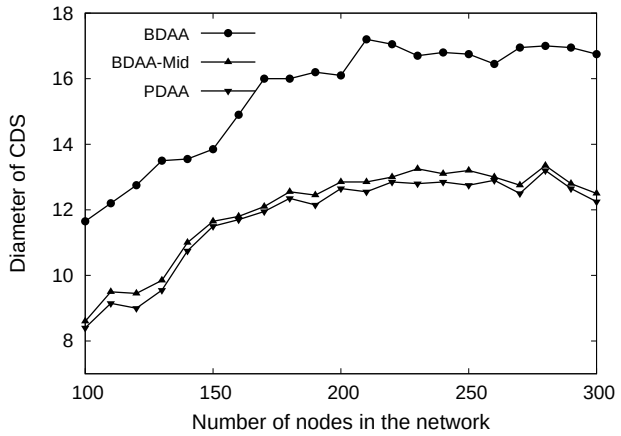
n, the number of nodes in a given network r, the transmission range ratio, i.e., $r = \frac{r_{max}}{r_{min}}$ The network density, i.e., the number of nodes per area

5.2 Effects of Number of Nodes

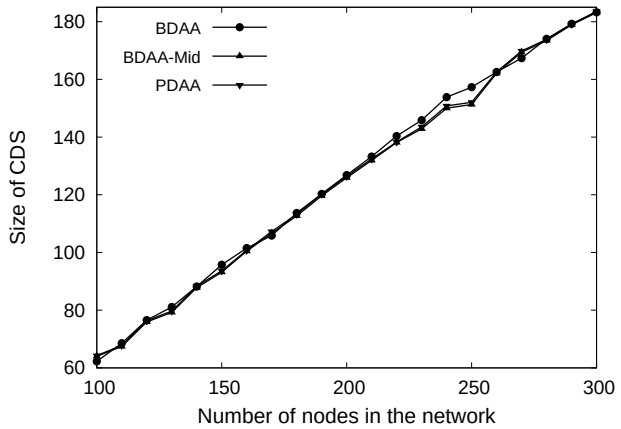
To evaluate the performance of the proposed algorithms under different number of nodes, we randomly deployed n nodes to a fixed area of 3,000m x 3,000m. n changed from 100 to 300 with an increment of 10. Each node v_i randomly chose the transmission range $r_i \in [r_{min}, r_{max}]$ where $r_{min} = 100m$ and $r_{max} = 300m$. For each value of n, 1,000 network instances were investigated and the results were averaged.

5.2.1 Simulations for BDAA and PDAA

The purpose of this simulation is to evaluate the performance of our proposed algorithms under different number of nodes and verify the importance of root selection at the same time. In order to highlight the root selection, we use a variation of BDAA, called BDAA-Mid, as a reference. Compared to BDAA, BDAA-Mid selects the center of network



A Compare the Diameter of CDS



B Compare the Size of CDS

Figure 5-1. Simulations for BDAA and PDAA

as the root instead of choosing randomly. Also, we admit PDAA into this simulation and β is set to 1.

Fig. 5-1A compares the diameter of CDS constructed by the three algorithms. It is shown that, under different number of nodes deployed in networks, the CDS built by PDAA has the smallest diameter. We observe that the gap between BDAA and BDAA-Mid is shown clearly, which indicates that the CDS could achieves small diameter with the root locating at the center of the network. On the other hand, the difference between BDAA-Mid and PDAA is small, which highlights an important fact that if the center of network is detected, the diameter of CDS rooted at the center will be nearly

optimal, even using an algorithm that only guarantees a loose bound on diameter, such as BDAA. In order to see how far the diameter of CDS returned by BDAA-Mid from the optimal solution, we set β to 1 in PDAA. Since with $\beta = 1$, PDAA will produce a CDS with minimum diameter mostly.

In Fig. 5-1B, we present the size of CDS obtained from all three algorithms, depending on the number of nodes deployed. The sizes of CDS generated from the three algorithms are quite close to each other and they all increase with the increase of number of nodes. Also, considering the same number of nodes, BDAA returns a larger size of CDS than PDAA and BDAA-Mid. Although the gaps between these algorithm look small in Fig. 5-1B, the difference between BDAA and BDAA-Mid is clear to observe in the comparison of real data, which illustrates that the size of CDS can be reduced by choosing the center of network as the root. Therefore, the center of network appears to be an important issue in the construction of CDS.

In Table 5-1, we present the running time for the three algorithms. As the complexity analysis indicates, the runtime of BDAA-Mid and PDAA are much higher than that of BDAA. This is due to the long time spent on detecting the center of network. Moreover, we show in Table 5-1 that the BDAA-Mid still runs faster than PDAA, since PDAA needs to compute T_{SPT} to shorten the diameter. When the number of nodes increases, PDAA and BDAA-Mid spend more time on detecting the center of network. Therefore, it is a tradeoff between the size (diameter) of CDS and running time of the proposed algorithms.

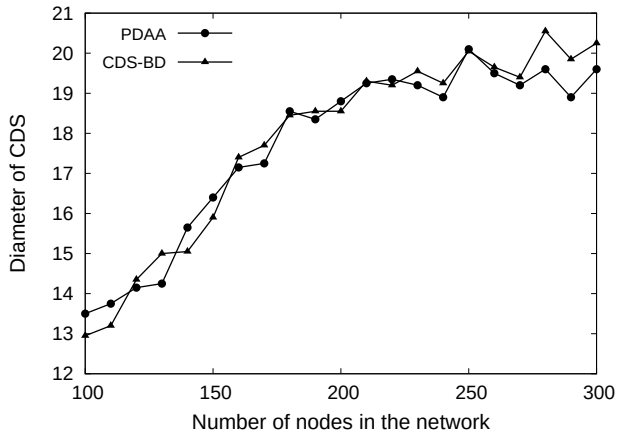
5.2.2 Simulations for CDS-BD and PDAA

We also conducted simulations to compare the performance of CDS-BD and PDAA. CDS-BD is an algorithm proposed in [47] to construct a CDS with bounded diameter and size. It selects a root randomly and spans a CDS from the root. The approximation ratios of CDS-BD are 11.4 and 3 on size and diameter respectively. For the purpose of fairness, we set $\beta = 3$ (the approximation ratio of PDAA on diameter) in PDAA and also choose the root of CDS randomly.

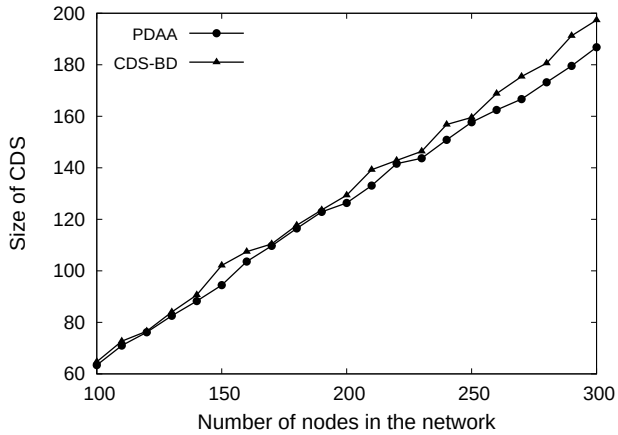
Number of Node	BDAA Runtime	BDAA-Mid Runtime	PDAA Runtime
100	0.0030	1.2640	1.6460
110	0.0040	1.8270	2.3550
120	0.0035	2.6260	3.4025
130	0.0040	3.6260	4.6545
140	0.0065	4.8320	6.2470
150	0.0055	6.4380	8.3000
160	0.0080	8.3955	10.790
170	0.0095	10.732	13.783
180	0.0105	13.612	17.588
190	0.0135	16.688	21.450
200	0.0150	20.604	26.650
210	0.0185	25.342	32.406
220	0.0195	30.427	38.916
230	0.0235	41.962	54.045
240	0.0245	49.698	63.839
250	0.0280	58.496	74.494
260	0.0350	78.144	100.67
270	0.0345	90.052	116.11
280	0.0385	104.91	134.48
290	0.0460	134.32	172.55
300	0.0480	155.49	197.78

Table 5-1. Running time(ms) for BDAA, BDAA-Mid and PDAA

Fig. 5-2A shows that the diameters of CDS built by the two algorithms are quite close to each other and the two curves intersect with each other when different number of nodes deployed in the network. For example, when the number of nodes deployed is 130, PDAA achieves smaller diameter than CDS-BD, while at 140, CDS-BD has smaller value. The reason why they look close to each other is that they all guarantee a constant approximation ratio of 3 on diameter. Even though PDAA does not always outperform CDS-BD from this result, out of the 21 points in Fig. 5-2A, PDAA outperforms CDS-BD at 16 points, which is around 76% in probability. So statistically, if the number of nodes deployed in the network is within the range of 100 to 300, which is the simulation environment in our model, PDAA is still better than CDS-BD in reducing the diameter.



A Compare the Diameter of CDS



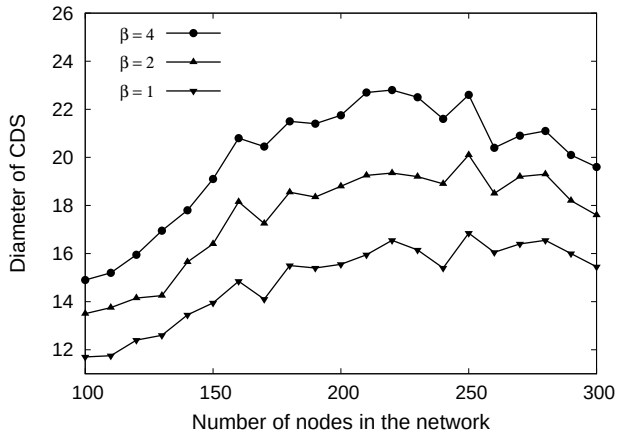
B Compare the Size of CDS

Figure 5-2. Simulations for CDS-BD and PDAA

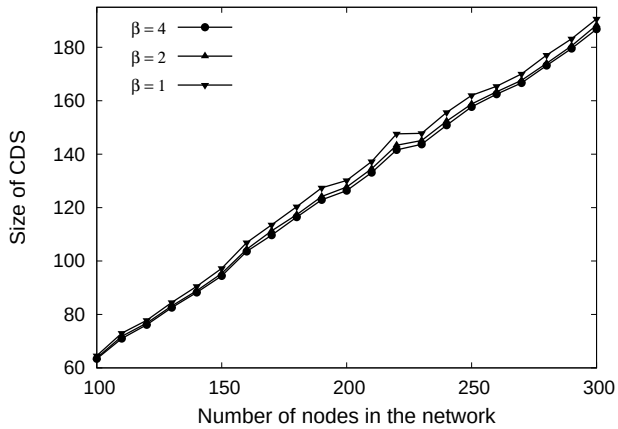
Fig. 5-2B provides the performance comparison of the two algorithms on the size of CDS. It shows PDAA always constructs a CDS with smaller size then CDS-BD, which is much better than theoretical analysis we gave in Section 4.2. Therefore, we can conclude that PDAA outperforms CDS-BD on size and on diameter with high probability as well.

5.2.3 Simulations Based on Different β

In the above simulations, β is set to 1 or 3. In this section, we conduct the simulations with different values of β. We study the relationship between β and the size of CDS and the relationship between β and diameter of CDS. As the root selection will not affect the comparison, we randomly choose the root of CDS in this group of simulations. Results are shown in Fig. 5-3.



A Compare the Diameter of CDS



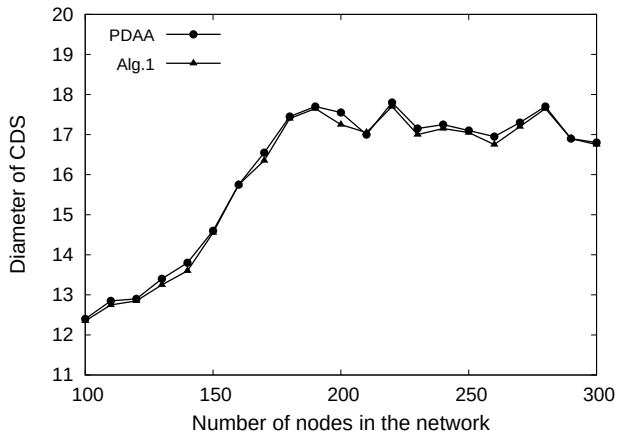
B Compare the Size of CDS

Figure 5-3. Simulations based on different β

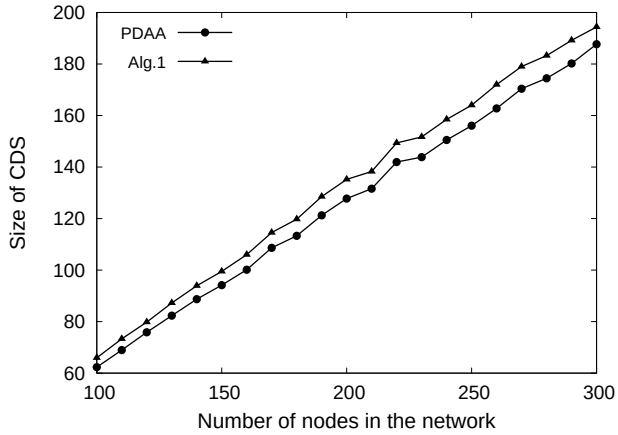
In Fig. 5-3A, each line represents the diameter of CDS based on one of different values of β. When β is set to 1, PDAA adds a shortest path from v to r if $D_{CDS}(r, v)$ is larger than $D_{SPT}(r, v)$. Therefore, PDAA with $\beta = 1$ returns a CDS with the smallest diameter. When β is set to 4, PDAA will not cause the shortest paths to be added in, since PDAA only adds the path from v to r in T_{CDS} under the condition that $D_{CDS}(r, v)$ is greater than 4 times of $D_{SPT}(r, v)$, however, the upper bound of PDAA on diameter is 4. Thus, the CDS by PDAA with $\beta = 4$ has the largest diameter. For $\beta = 2$, the corresponding line is in the middle. Therefore, as we expected, the diameter of CDS built by PDAA could be controlled by adjusting the values of β.

In Fig. 5-3B, each line represents the size of CDS based on one of different values of β. When β is set to 1, if $D_{CDS}(r, v)$ is larger than $D_{SPT}(r, v)$, PDAA adds a shortest path from v to r. This strategy will incur more nodes to be added. On the opposite, when β is set to 4, no shortest path is needed, which results in a CDS with smaller size. For $\beta = 2$, the corresponding line is in the middle, the same situation as in Fig. 5-3A. In conclusion, the performance of PDAA can be balanced depending on the value of β and the tradeoff between size and diameter is clear.

5.2.4 Simulations for k-m-CDS



A Compare the Diameter of CDS



B Compare the Size of CDS

Figure 5-4. Simulations for k-m-CDS

In this section, we are interested in evaluating the performance of FTAA. We intend to illustrate that FTAA improves the fault tolerance of 1-CDS by adding marginal

overhead (in terms of the number of nodes added into 1-CDS). We generate a 1-CDS using PDAA with random root selection and β is set to 2 here. We take the 1-CDS generated using PDAA as the input of FTAA afterwards, and we set k=2 and $m = 1$.

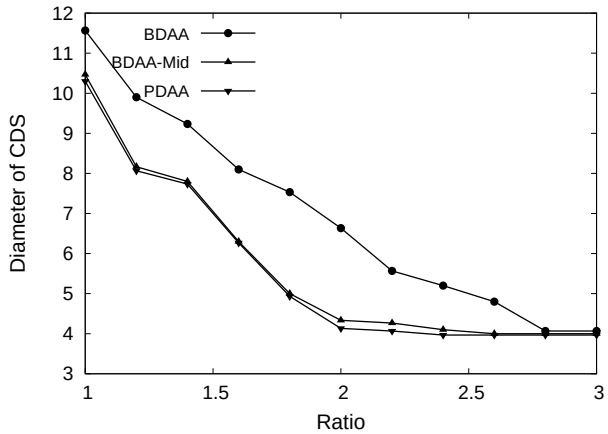
Fig. 5-4A compares the performance of FTAA and PDAA in terms of the diameter of CDS. As we expected, there is little difference on the diameter of CDS based on the two algorithms, which perfectly matches our theoretical analysis for the diameter of k-m-CDS. Therefore, FTAA enhances the fault tolerance of CDS without affecting its diameter greatly.

Meanwhile, as observed from Fig. 5-4B, the size of k-m-CDS obtained from FTAA is certainly larger than 1-CDS by PDAA. Specifically, the performance of the two algorithms is relatively proportional. As observed from our experiments, the size of k-m-CDS obtained from FTAA is almost 1.06 times the size of CDS returned by PDAA. The results indicate that considering the fault tolerance will increase the size of the CDS at the same time. However, the increase in size is still bounded and predictable. Therefore, it is clear to see the tradeoffs between the three factors.

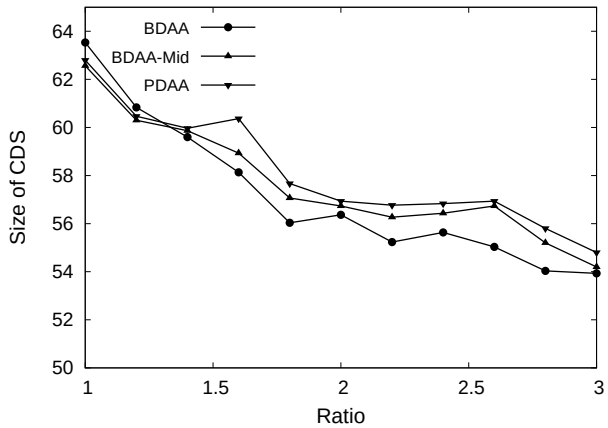
5.3 Effects of Transmission Ratio

We also conducted simulations to compare the performance of all algorithms when changing the transmission range ratio r as well as to see how this change affects the size and diameter of an obtained CDS. To change r, we fixed $r_{min} = 100m$ and changed r_{max} from 100m to 300m with an increment of 20. In this experiment, we randomly deployed 100 nodes into a fixed area of size 3,000m x 3,000m. Each node randomly chose a transmission range in $[r_{min}, r_{max}]$. For each network instance, we ran the test for 1000 times.

Fig. 5-5A compares the performance of the proposed algorithms in terms of the CDS diameter. As shown in Fig. 5-5A, PDAA is the best. In particular, the CDS diameter obtained from PDAA is only 5 % smaller than that of BDAA-Mid and 27 % smaller than



A Compare the Diameter of CDS



B Compare the Size of CDS

Figure 5-5. Effects of transmission ratio

that of BDAA. Again, these results reveal that selecting the center of the network as the CDS root can reduce the CDS diameter.

As we expected, in Fig. 5-5B, the CDS produced by PDAA has a bigger size than BDAA-Mid and BDAA, since PDAA need to add more nodes in CDS in order to achieve a shorter diameter. Apparently, BDAA performs much better than the two others, just because it puts less concern on CDS diameter.

Fig. 5-5 illustrates how the transmission ranges affect the CDS size and diameter. As can be seen in Fig. 5-5, three curves show obvious trend of decrease. In other words, the CDS size and diameter decrease 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.

5.4 Effects of 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 and diameter. To change the network density, we fixed the number of nodes n = 100 and increased the area from 500m x 500m to 1,500m x 1,500m with an increment of 20. 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 $r_{min} = 100m$ and $r_{max} = 300m$. For each network instances, we ran the simulations for 1000 times and the results were averaged.

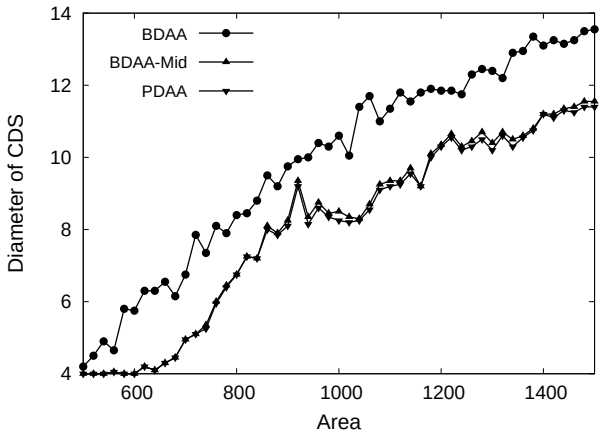
Fig. 5-6A provides the performance comparison of three algorithms in terms of the CDS diameter. As revealed by Fig. 5-6A, PDAA still outperforms the other two in this case. And BDAA-Mid outperforms BDAA. Specifically, the CDS diameter obtained from PDAA is only 1.3 % less than that of BDAA-Mid and 15.8 % less than that of BDAA.

As predicted, Fig. 5-6B indicates that BDAA can build up a CDS with smaller size than BDAA-Mid and PDAA. The difference between BDAA and PDAA is obvious when the network density is high. However, the three curves begin to merge as the network density decreases.

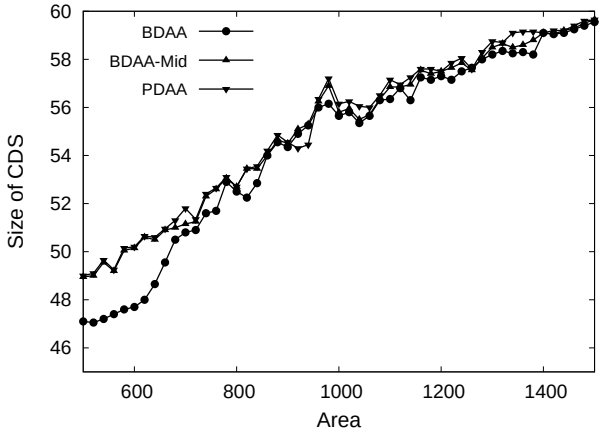
In addition, Fig. 5-6 shows the obvious trend of increase of three curves, which implies that the CDS size and diameter get bigger when the network density decreases. This is because when the network density decreases, the number of neighbors of each node decreases as well. Thus the CDS size and diameter need to be larger to dominate all nodes in a network.

5.5 Summarization

The simulation results can be summarized as follows:



A Compare the Diameter of CDS



B Compare the Size of CDS

Figure 5-6. Effects of network density

The CDS rooted at the center of network always has a nearly minimum diameter and smaller size compared with all other CDSs, which choose the root randomly. However, the tradeoff between diameter (size) and running time of proposed algorithms exists, since time complexity of finding the center of network is high.

Under the same network conditions, the results shows PDAA performs better than CDS-BD proposed in [47] in most of time, even though we select the root randomly.

PDAA offers a method to balance the diameter and size of CDS through an user-defined input by adjusting the value of β.

The simulation results show that FTAA improves the fault tolerance of 1-CDS with only marginal extra overhead. Therefore, the tradeoffs between the three factors are shown clearly.

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this work, we investigate the fault tolerant CDS problem with bounded diameter in wireless networks. We propose an approximation algorithm for a general case of the k-m-CDS with bounded diameter, and two approximation algorithms for 1-CDS, which could be applied into the solution of the k-m-CDS model. We analyze the approximation ratios of the these algorithms in DGB and they guaranteed constant ratios for those factors considered. Moreover, the proposed algorithms for 1-CDS can be implemented in distributed manner and the analysis of time and message complexities is presented as well. Through extensive simulations, we verify that our proposed algorithms can effectively reduce the diameter and size of CDS and outperform CDS-BD [47].

6.2 Future Work

Besides extending our results in this work to the directed graph, we are also interested in finding a number of CDSs to balance the energy consumption. Most of the research work on the CDS problem focuses on constructing a single CDS. However, as nodes in the CDS will be used heavily, they can quickly run out of power, thereby shortening the network lifetime. Therefore, it is natural to find a number of different CDSs, and dynamically rotate the roles of these CDSs to carry out the network tasks.

REFERENCES

[1] B. An and S. Papavassiliou, "A mobility-based clustering approach to support mobility management and multicast routing in mobile ad-hoc wireless networks," *international Journal of Network Management*, vol. 11, no. 6 pp. 387–295, Nov./Dec. 2001.

[2] M. Gerla and J. T. Tsai, "Multicluster, mobile, multimedia, radio network," *ACM/Baltzer Journal on Wireless Networks*, vol. 1, no. 3, pp. 255–265, Sep. 1995.

[3] R. Sivakumar, P. Sinha and V. Bharghavan, "CEDAR: a core-extraction distributed ad hoc routing algorithm" *Selected Areas in Communications, IEEE journal on*, vol. 17, no. 8, pp. 1454–1465, Aug. 1999.

[4] B. Das and V. Bharghavan, "Routing in ad hoc networks using minimum connected dominating sets," *International Conference on Communications*, Montreal, Canada, June 1997.

[5] B. Das, R. Sivakumar and V. Bharghavan, "Routing in ad hoc networks using a spine," *International Conference on Computers and Communications Networks*, Las Vegas, NV, Sept. 1997.

[6] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," *Proc. of the Third International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pp. 7–14, 1999.

[7] H. Lim and C. Kim, "Multicast tree construction and flooding in wireless ad hoc networks," *Proc. of 3rd International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile System*, Boston, MA, pp. 61–68, 2000.

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

[9] I. Stojmenovic, M. Seddigh and J. Zunic, "Dominating sets and neighbor elimination based on broadcasting algorithms in wireless networks," *Parallel and Distributed Systems, IEEE Transactions on* , vol. 13, no. 1, pp. 14–25, 2002.

[10] J. Wu and F. Dai, "Broadcasting in ad hoc networks based on self-pruning," *IEEE INFOCOM*, 2003.

[11] J. Wu and B. Wu, "A transmission range reduction scheme for power-aware broadcasting in ad hoc networks using connected dominating sets," *Proc. of 2003 IEEE Semiannual Vehicular Technology Conference*, Oct. 2003.

[12] J. Wu, B. Wu and I. Stojmenovic, "Power-aware broadcasting and activity scheduling in ad hoc wireless networks using connected dominating sets," *Proc. of IASTED International Conference on Wireless and Optical Communication (WOC 2002)*, 2002.

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

[14] S. Datta and I. Stojmenovic, "Internal nodes and shortcut based routing with guaranteed delivery in wireless networks," *Cluster Computing*, vol. 5, no. 2, pp. 169–178, 2002.

[15] B. Chen, K. Jamieson, H. Balakrishnan and R. Morris, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Wireless Networks*, vol. 8, no. 5, pp. 481–494, 2002.

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

[17] J. Shaikh, J. Solano, I. Stojmenovic and J. Wu, "New metrics for dominating set based energy efficient activity scheduling in ad hoc networks," *Proc. of WLN Workshop (in conjunction to IEEE Conference on Local Computer Networks)*, pp. 726–735, Oct. 2003.

[18] J. Wu, F. Dai, M. Gao and I. Stojmenovic, "On calculating power-aware connected dominating set for efficient routing in ad hoc wireless network," *Journal of Communications and Networks*, vol. 5, no. 2, pp. 169–178, March 2002.

[19] Y. Xu, J. Heidemann and D. Estrin, "Geography-informed energy conservation for ad hoc routing," *MobiCom 2001*, pp. 70–84, 2001.

[20] B. Deb, S. Bhatnagar and B. Nath "Multi-resolution state retrieval in sensor networks," *First IEEE Workshop on Sensor Network Protocols and Applications*, pp. 19–29, 2003.

[21] A. Helmy "Efficient resource discovery in wireless adhoc networks: con- tacts do help," *Manuscript*, 2004.

[22] U. Kozat and L. Tassiulas, "Service discovery in mobile ad hoc networks: an overall perspective on architectureal choices and network layer support issues," *Ad Hoc Networks Journal*, vol. 2, no. 1, pp. 23–44, Jan. 2004.

[23] X. Cheng, M. Ding D. H. Du and X. Jia, "On the construction of connected dominating set in ad hoc wireless network," *in Special Issue on Ad Hoc Wireless Communications and Mobile Computing*, 2004.

[24] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *in Mobile Computing (ed. T. Imielinski and H. Korth)*, Kluwer Academic Publishers, 1996.

[25] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90–100, 1999.

[26] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler and L. Viennot, "Optimized Link State Routing Protocol," *IETF Internet Draft*, October, 2001.

[27] S. Y. Ni, Y. C. Tseng, Y. S. Chen and J. P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Proc. MobiCom*, pp. 151–162, Seattle, Aug. 1999.

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

[29] M. R. Garey and D. S. Johnson, *Computers and Intractability: A guide to the theory of NP-completeness*, Freeman, San Francisco 1978.

[30] B. Das, R. Sivakumar and V. Bharghavan, "Routing in ad hoc networks using a spine," *International Conference on Computers and Communications Networks*, Las Vegas, NV. Sept. 1997.

[31] R. Sivakumar, B. Das and V. Bharghavan, "An improved spine-based infrastructure for routing in ad hoc networks," *IEEE Symposium on Computers and Communications*, Athens, Greece. June 1998.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] Jeremy Blum, Min Ding, Andrew Thaeler and Xiuzhen Cheng, "Connected Dominating Set in Sensor Networks and MANETs," *Handbook of Combinatorial Optimization*, pp. 329–369, 2005.

[38] 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.

[39] M. T. Thai, F. Wang, D. Liu, S. Zhu and D. Z. Du, "Connected dominating sets in wireless networks with different transmission ranges," *IEEE Transactions on Mobile Computing*, vol. 6, no. 7, July 2007.

[40] F. Wang, M. T. Thai and D. Z. Du, "On the construction of 2-connected virtual backbone in wireless network," *IEEE Transactions on Wireless Communications, accepted with revisions*, 2006. (The first version is in 2005 a technical report)

[41] W. Shang, F. F. Yao, P. J. Wan, X. Hu, "Algorithms for minimum m-connected k-dominating set problem," *COCOA 2007*, pp. 182–190, 2007.

[42] M. T. Thai, N. Zhang, R. Tiwari, and X. Xu, "On approximation algorithms of k-connected m-dominating sets in disk graphs," *Journal of Theoretical Computer Science*, vol. 385, pp.49-59, 2007.

[43] Y. Wu, F. Wang, M. T. Thai and Y. Li, "Constructing k-connected m-dominating sets in wireless sensor networks," *2007 Military Communications Conference (MILCOM 2007)*, Orlando, FL, October 2007

[44] F. Dai and J. Wu, "On construction k-connected k-Dominating set in wireless network," *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*, 2005.

[45] K. Mohammed, L. Gewali, and V. Muthukumar, "Generating quality dominating sets for sensor network," *Proceedings of the Sixth International Conference on Computational Intelligence and Multimedia Applications*, pp. 204-211, August 2005.

[46] T. H. Cormen, C. E. Leiserson R. L. Rivest and C. Stein, *Introduction to Algorithms 2 ed.*, McGraw Hill 2001.

[47] Y. Li, D. Kim, F. Zou, D.-Z. Du, "Constructing Connected Dominating Sets with Bounded Diameters in Wireless Networks," *International Conference on Wireless Algorithms, Systems and Applications*, Chicago, IL, Auguest 1-3 2007.

[48] D. Matula, "k-blocks and Ultrablocks in Graphs," *Journal Combin. Theory Ser. B.*, vol. 24, pp. 1-13, 1978.

[49] Y. Li, M. T. Thai, F. Wang, D.-Z. Du, "On the Construction of a Strongly Connected Broadcast Arborescence with Bounded Transmission Delay," *IEEE Transactions on Mobile Computing*, vol. 5, no. 10, pp. 1460-1470, 2006.

[50] L. Brim, I. Cerna, P. Krcal and R. Pelanek, "Distributed shortest paths for directed graphs with negative edge lengths," *Techmical report FIMU-RS-2001-01*, Faculty of Informatics, Masaryk University, http://www.fi.muni.cz/informatics/reports, 2001.

[51] M. B. Sharma, S. S. Iyengar and N. K. Mandyam, "An optimal distributed depth-first-search algorithm," *In Proc. of the seventeenth annual ACM conference on Computer science: Computing trends in the 1990's*, Louisville, Kentucky, pp.287-294, 1989.

BIOGRAPHICAL SKETCH

Ning Zhang received his bachelor of science in computer science from Beijing University of Technology, China in summer 2006. After graduation, he joined the graduate school at the University of Florida in fall 2006, where he received his master of science in summer 2008 from the Department of Computer and Information Science and Engineering. His research interests are centered on the combinatorial optimization and its applications to computer networks. More specifically, the focus of his research is to design and analyze efficient algorithms (mainly approximation algorithms) for computationally hard problems in wireless sensor networks, wireless networks. In fall 2008, he will join the Department of Computer Science in University of Wisconsin at Madison, a top ten computer science department in the U.S., as a PhD student.