Trade-off Scheme for Fault Tolerant Connected Dominating Sets on Size and Diameter

Ning Zhang* Incheol Shin * Feng Zou† Weili Wu † My T. Thai*

ABSTRACT

Connected Dominating Set (CDS) has been a well known approach for constructing a virtual backbone to alleviate the broadcasting storm in wireless networks. Current research has focused on minimizing the size and 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 paper, 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 network latency. This model also addresses the tradeoffs between the objective functions. We next propose one approximation algorithm and two distributed algorithms with constant ratios for the model. Simulation results show that our algorithms can gain good tradeoffs between the three factors, which coincide with theoretical analysis. Moreover, our algorithms could obtain a better performance than previous work.

Categories and Subject Descriptors: C.2.1 [Network Architecture and Design]: Wireless communication

General Terms: Algorithms.

Keywords: Wireless Ad-Hoc Networks, Connected Dominating Set, Fault Tolerance, Bounded Diameter.

1. INTRODUCTION

Connected Dominating Set (CDS) has been a well known approach for constructing a virtual backbone to alleviate the broadcasting storm [2] in wireless networks. With the help of the CDS, only nodes in CDS need to forward the messages. Meanwhile, routing becomes much easier and can adapt quickly to network topology changes [6]. Furthermore, using a CDS as forwarding nodes can efficiently reduce the energy consumption, which is also a critical concern in wireless networks.

*Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, 32611. Email: {nzhang,ishin,mythai}@cise.ufl.edu

†Department of Computer Science and Engineering, University of Texas - Dallas, Richardson, TX, 75083. Email: phenix.zou@student.utdallas.edu, weiliwu@utdallas.edu

FOWANC'08, May 26, 2008, Hong Kong SAR, China.

Since every node in CDS works as the central management agents with heavy load, constructing a CDS with minimized size can greatly help to reduce transmission interference and the number of control messages [6]. Another issue that recent research [8] [9] has addressed is the diameter of CDS, which is the longest shortest path between any pair of nodes in CDS. Considering the situation that the receiver is not within the transmission range of the sender, communicate through multi-hop links by using some intermediate nodes to relay the messages, which is called multi-hop routing, is needed. Since a CDS with large diameter often leads to an increase in propagation error and transmission latency, a CDS with small diameter is certainly preferred for reliable message delivery and short delay in this case.

As CDS is often very vulnerable due to frequent node failure and link failure, which is inherent in wireless networks, constructing a fault tolerant CDS that continues to function during node or link failure is another issue. The previous work [14]- [16] have addressed this issue. However, they only considered the size of CDS together with the fault tolerance, without the diameter of CDS.

Since a CDS problem is NP-hard [1] 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 paper, we first study the problem of constructing a CDS by considering the three factors together. The approximation algorithm for k-Connected m-Dominating Sets (km-CDS, also known as fault tolerant CDS) with bounded diameter is presented. We minimize the size and the diameter of km-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 paper are as follows:

1. An approximation algorithm for km-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.

2. 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.

3. 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.

4. Comparing with CDS-BD algorithm proposed in [9], the simulation results show that our algorithms outperform CDS-BD under the same network condition.

The rest of the paper is organized as follows. Section 2 describes the related work of this problem. The wireless communication model and some preliminaries are presented in Section 3. our proposed model is studied in Section 4. Two distributed algorithms for 1-CDS with bounded diameter and their theoretical analysis are presented in Section 5.1 and Section 5.2 respectively. Section 6 provides the simulation results and Section 7 concludes this paper.

2. RELATED WORK

Although the CDS problem has been extensively studied [3]-[7], little work has been done on fault tolerant CDS problem with bounded diameter. Mohammed *et al.* mentioned the problem of constructing CDS with small diameter [8]. However, they did not give a guaranteed performance in their model. In [9], Li *et al.* studied the CDS problem with bounded diameter in Unit Disk Graph (UDG) and proposed a constant approximation algorithm, called CDS-BD. However, their algorithm is centralized and no experimental results are provided. In contrast, our algorithms can be implemented in distributed manner for 1-CDS not only in UDG, but also disk graphs. The constant approximation ratios are also preserved.

In summary, none of the previous work address our model in literature. The contributions of this paper is multi-fold: First, the performance of this model is not fixed, it can be adjusted in a range by an user-defined input. Second, we prove that the tradeoffs exist between the three factors, that is, it is hard to optimize them at the same time. Third, the experiments reveal some important properties of our model, which could not be discovered in theoretical analysis.

3. WIRELESS COMMUNICATION MODEL AND PRELIMINARIES

In this paper, we model the wireless network using a Disk Graph with Bidirectional links (DGB) $G = (V, E)$. The nodes in V are located in the two dimensional Euclidean plane and each node $v_i \in V$ has a transmission range $r_i \in [r_{min}, r_{max}]$. R is denoted as the transmission range ratio, i.e. $R = r_{max}/r_{min}$. The edge set E represents all links in the network. There is an edge between a pair of nodes if they are within the transmission range of each other, which means that an edge (v_i, v_j) in E if the distance between two nodes $d(v_i, v_j) \leq min\{r_i, r_j\}$. Note that if denote K as the independent neighbors of a node u in DGB, then $K = 5$ if $R = 1$, otherwise, $K = 10(\lceil \frac{\ln(k)}{\ln(2\cos(\frac{\pi}{5}))} \rceil + 1)$ [3].

A Dominating Set (DS) of this graph is a subset $C \subset V$ such that each node either belongs to C or is adjacent to at least one node in C. A CDS is a DS which induces a connected subgraph. The size of a CDS is the number of nodes in CDS. 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 nodes in CDS. We also denote D^* as the minimum diameter of CDS of G, and CDS^* as the CDS with smallest size.

4. FAULT TOLERANT CDS WITH BOUNDED DIAMETER

In this section, we introduce our problem and provide a solution for km-CDS, where $1 \leq k \leq m + 1$.

Before we introduce the definition of the problem, we need to give the following definitions in graph theory: A graph G is k-*connected* if it is connected and removing any $k - 1$ nodes from G will not partition G, i.e, G is still connected. 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 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 fault tolerant CDS with bounded diameter problem could be formally defined as follows:

DEFINITION 1. ***Fault Tolerant 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.*

The joint optimization problem considers three factors (size, diameter and fault tolerance) together, and the three factors are optimized at the same time. Some recent work [14]- [16] has addressed the general fault tolerant CDS Problem. However, none of them mentioned how to bound the diameter of it.

In our previous work [15], we have proposed a solution for km-CDS problem, where $1 \leq k \leq m + 1$, as illustrated in Algorithm 1. In this paper, we still use this algorithm to solve our joint optimization problem. However, a new analysis is proposed for the diameter of km-CDS. The main idea of Algorithm 1 is that merging all the k'-blocks in 1-Connected m-Dominating Set (1m-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 $k' = k$. We can use any 1-CDS with bounded size and diameter as the input of Algorithm 1. However, in order to make the solution adjustable by the user, an (α, β)-CDS, to be introduced in Section 5.1, is preferred to be an input of Algorithm 1.

Algorithm 1 km-CDS Algorithm [15]

1: INPUT: A connected DGB $G = (V, E)$ and a 1-CDS C_{11} with bounded diameter and size
2: OUTPUT: A km-CDS C_{km} with bounded diameter and size
3: Step 1: Based on the input C_{11}, construct a 1m-CDS C_{1m} by using CDSMIS Algorithm in [15]
4: Step 2: Compute all the k'-blocks in C_{1m}, initially $k' = 2$.
5: Step 3: If there is more than one k'-block in C_{1m}, find the **shortest path** in the original graph that satisfies the two requirements: (i) the path can connect two k'-blocks sharing a same separating set to be one k'-block of C_{1m}. (ii) the path does not contain any nodes in C_{1m} except the two end points. Then add all intermediate nodes in this path to C_{1m}.
6: Step 4: Repeat Step 2 and 3, until there is only one k'-block in C_{1m}.
7: Step 5: Increase k' by 1 and then repeat Step 2, 3 and 4, until $k' = k$. The resultant C_{1m} will be C_{km}.

4.1 Theoretical Analysis

THEOREM 1. *If the input 1-CDS has an approximation ratio of α on size ($\alpha > 1$), then Algorithm 1 produces a km-CDS with $(2K + 2m + 1)\alpha-$approximation on size, where C^*_{1m} and C^*_{km} are the $1m$-CDS and km-CDS with optimal solution on size respectively.*

Proof: C_{km} is the union of C_{1m} and the nodes added into C_{1m}, in order to make C_{1m} k-connected. The number of nodes we added to make C_{1m} k-connected is at most $2(k-2)(|C_{1m}|-1)+2(K+1)(|C_{1m}|-1)$ [15]. Therefore,

$$\begin{aligned} |C_{km}| &= |C_{1m}| + 2(k-2)(|C_{1m}|-1) + 2(K+1)(|C_{1m}|-1) \\ &\le (2K + 2k - 1)|C_{1m}| \end{aligned}$$

However, in our previous work [15], we already concluded the following inequality,

$$|C_{1m}| \le \alpha|CDS^*| + (K + m - 1)|C^*_{1m}|$$

Thus,

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

□

LEMMA 1. $d(C_{1m}) \le d(C_{11}) + 2$.

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 2. $d(C_{km}) \le d(C_{1m}) + 2$.

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

Proof: From Lemma 1 and 2, we have the following inequality:

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

□

5. 1-CDS WITH BOUNDED DIAMETER

In this section, we introduce two algorithms for 1-CDS (also known as CDS) with bounded diameter. One is called Basic Distributed Algorithm (BDA) and another is called Progressive Distributed Algorithm (PDA). The two algorithms could be used as an input of Algorithm 1. The benefit of BDA is the low time complexity on constructing CDS, while PDA performs well on optimizing the size and diameter of CDS. We will prove the two facts through simulation. The definition of this problem is same as Definition 1 when $k = m = 1$.

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.

5.1 Basic Distributed Algorithm and Analysis

The main idea of BDA is as follows. First, use Wan's distributed MIS algorithm [4] to construct an MIS. Color all the nodes in MIS black. Second, 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. Third, connect the nodes in MIS from low level to high level with minimum number of hops.

One existing distributed algorithms for MIS [4] is executed to obtain a DS. The obtained MIS satisfies the following lemma:

LEMMA 3. *Any pair of complementary subsets of a constructed MIS has a distance of exactly two hops. [4]*

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 Algorithm 2:

Algorithm 2 Basic Distributed Algorithm (BDA)

1: INPUT: A connected DGB $G = (V, E)$ and an MIS computed by Wan's algorithm [4]
2: OUTPUT: A CDS T_{CDS} with minimum diameter
3: Each node maintains a unique node ID and a status of *unexplored* initially
4: Color all nodes in MIS black and color every node adjacent to a black node in grey
5: 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 $0 \le k \le k^*$. Suppose k^* is the maximum value of k.
6: r broadcasts $EXPLORE$ messages to its neighbors at level 1, where r is at level 0
7: 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.
8: Upon receiving an $EXPLORE$ message, an *unexplored* black node y at level $i (i \ge 2)$ 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.
9: The algorithm stops until there is no node changed from grey to blue.
10: The union of black and blue nodes is T_{CDS}.

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 3 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 we have to add is exactly $|MIS|-1$.

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

THEOREM 3. *Denote T_{CDS} as our solution obtained from BDA, 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, $|I| \leq K|CDS^*|$ [3]. 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 nodes exist at level 1, the black nodes in level 2 can connect with r with 2 hops. Therefore, $d(T_{CDS}) \leq 2*2(k^*-2)+4 \leq 4D^*+4$ □

THEOREM 4. *The BDA 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 [4]. After that, we use linear message and take at most linear time to connect the nodes in MIS. Overall, BDA has $O(n)$ time complexity and $O(n \log n)$ message complexity.

□

5.2 Progressive Distributed Algorithm and Analysis

In this section, we introduce (α,β)-CDS into our model to be the input of Algorithm 1. Also, it can solve the CDS with bounded diameter problem proposed in [9]. 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 in given in wireless networks as follows:

DEFINITION 2. (α,β)***-CDS:*** *For a fixed $\alpha \geq 1$ and $\beta \geq 1$, a CDS C of G meeting the following two requirements is called an (α,β)-CDS.*

1. *(Size) The size of C is at most α times the minimum CDS size.*
2. *(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 of it. Since an (α,β)-CDS might be used as an input of Algorithm 1, the tradeoff is still preserved for km-CDS.

The general idea of our PDA is as follows.

1. Construct a CDS T_{CDS} rooted at r by using BDA. 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 PDA is as follows:

Algorithm 3 Progressive Distributed Algorithm (PDA)

```
PDA(β)
  Locate the center of network and choose r at the center.
  Build a T_CDS rooted at 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 T_CDS 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
```

1. Root Selection and CDS Tree Construction: With Dijkstra's algorithm [13], which is used to solve the single-source shortest-paths problem, each node maintains a global variable, which stores the current longest 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, if a longer shortest path is found, the global variable of each node will be updated. While constructing T_{CDS} rooted at r by BDA, 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 [10] to initialize and maintain both of these attributes.

2. 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}.

3. 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 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 BDA, distributed SPT (dSPT) (e.g. [11]), and distributed DFS (dPFS) (e.g. [12]), T_{CDS}, T_{SPT} and a DFS traversal order could be achieved. In this way, with a *Manager* (e.g. root node), PDA could be easily initiated and terminated according to the details illustrated in Algorithm 3.

To evaluate the correctness of the PDA, 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 PDA.

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 4. *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 $2h \le D^* + 2$ 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 $2h \le D^* + 2$. First, it is easy to see that $2h \le d(T)$ and $d(T) = D$. Then we have the following:

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

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

□

LEMMA 5. *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 $d(T) \ne D$, 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. 1. Therefore, $h \le D^* + 1$, then the maximum number of hops between u and v is at most $2h$, that is $2\beta(D^* + 1)$. □

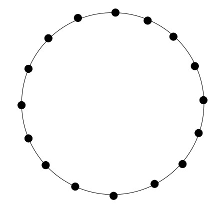


Figure 1: All the nodes in the ring are a CDS with diameter of 8

In real wireless network, case (2) rarely happens, since it requires all the hosts (nodes) are uniformly deployed as a ring. However, in most cases, they are deployed randomly. Therefore, the diameter of CDS returned by PDA is bounded by $\beta(D^* + 2)$ in most cases.

LEMMA 6. *The total number of nodes on the added shortest paths is at most $\frac{(5-\beta)K}{\beta-1}|CDS^*| + 3$.*

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 $v_i (i \ge 1)$ 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 $d[v_i] \le D_{SPT}(r, v_{i-1}) + D_{CDS}(v_{i-1}, v_i)$. The shortest path to v_i was added because $\beta D_{SPT}(r, v_i) < d[v_i]$. Combining the inequalities,

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

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

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

and therefore

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

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.,

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

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,

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

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

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 $|I| \leq K|CDS^*|$ [3], we have:

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

□

THEOREM 5. *Given the value of β, the approximation ratio α on the size of CDS 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 3 and Lemma 6,

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

□

THEOREM 6. *The time complexity of the PDA algorithm is $O(n^2)$, and the message complexity of the PDA algorithm is $O(n^2)$.*

Proof: From Theorem 4, the time complexity and message complexity for BDA 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 [11] [12]. 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 PDA are $O(n^2)$ and $O(n^2)$.

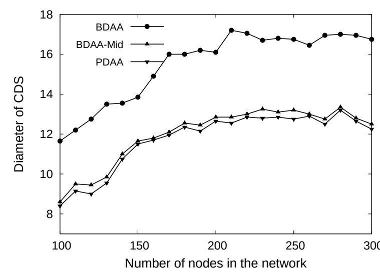
□

6. SIMULATION RESULTS

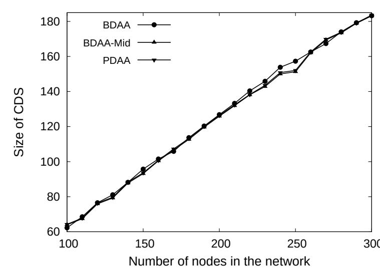
In this section, we conducted the simulation experiments to measure the diameter and size of CDS constructed by our proposed algorithms. Moreover, we are interested in comparing the CDSs returned by CDS-BD [9] and PDA. Since the running time for PDA and BDA has been discussed in Section 5, we also would 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 PDA, in order to see how the CDS size and diameter could be balanced. At last, we evaluate the performance of Algorithm 1 by comparing to PDA so that the tradeoffs between the three factors could be systematically discovered.

To simulate the network, 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.

6.1 Simulations for BDA and PDA



(a) Compare the Diameter of CDS



(b) Compare the Size of CDS

Figure 2: Simulations for BDA and PDA

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 BDA, called BDA-Mid, as a reference. Compared to BDA, BDA-Mid selects the center of network as the root instead of choosing randomly. Also, we include PDA in this simulation and β is set to 1.

Fig. 2(a) 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 PDA has the smallest diameter. We observe that the gap between BDA and BDA-Mid is shown clearly, which indicates that the CDS could achieves smaller diameter with the root locating at the center of network. On the other hand, the difference between BDA-Mid and PDA 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 BDA. In order to see how far the diameter of CDS returned by BDA-Mid from the optimal solution. we set β to 1 in PDA. Since with $\beta = 1$, PDA will produce a CDS with minimum diameter mostly.

In Fig. 2(b), we present the size of CDS obtained from all three algorithms, depending on the number of nodes deployed. The sizes of CDSs returned by the three algorithms are close to each other and they all increase with the number of nodes. Also, considering

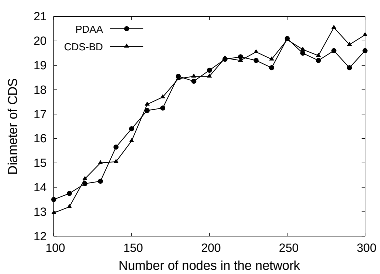
Number of Node	BDA Runtime	BDA-Mid Runtime	PDA Runtime
100	0.0030	1.2640	1.6460
120	0.0035	2.6260	3.4025
140	0.0065	4.8320	6.2470
160	0.0080	8.3955	10.790
180	0.0105	13.612	17.588
200	0.0150	20.604	26.650
220	0.0195	30.427	38.916
240	0.0245	49.698	63.839
260	0.0350	78.144	100.67
280	0.0385	104.91	134.48
300	0.0480	155.49	197.78

Table 1: Runtime(ms)

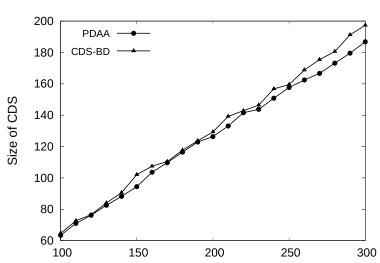
the same number of nodes, BDA returns a larger size of CDS than PDA and BDA-Mid. Although the gaps between these algorithm look small in Fig. 2(b), the difference between BDA and BDA-Mid is clear to observe in the comparison of real data, which illustrates that the size of CDS can be reduced by choosing 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 1, we present the running time for the proposed algorithms. As the complexity analysis indicates, the runtime of BDA-Mid and PDA is much higher than that of BDA. This is due to the long time spent on detecting the center of network. Moreover, we show in Table 1 that the BDA-Mid still runs faster than PDA, since PDA needs to compute T_{SPT} to shorten the diameter. When the number of nodes increases, PDA and BDA-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.

6.2 Simulations for CDS-BD and PDA



(a) Compare the Diameter of CDS



(b) Compare the Size of CDS

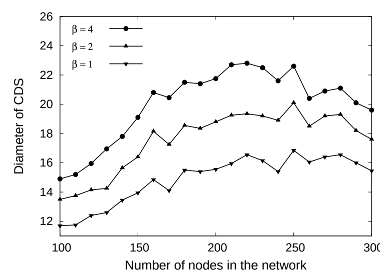
Figure 3: Simulations for CDS-BD and PDA

We also conducted simulations to compare the performance of CDS-BD and PDA. CDS-BD is an algorithm proposed in [9] 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 PDA on diameter) in PDA and also choose the root of CDS randomly.

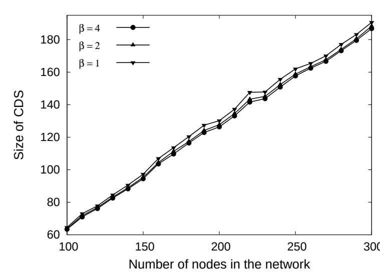
Fig. 3(a) shows that the diameters of CDS built by the two algorithms are quite close to each other and the two curves intersect with each other when different number of nodes deployed in the network. For example, when the number of nodes deployed is 130, PDA achieves smaller diameter than CDS-BD, while 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 PDA does not always outperform CDS-BD from this result, out of the 21 points in Fig. 3(a), PDA 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, PDA is still better than CDS-BD in reducing the diameter.

Fig. 3(b) provides the performance comparison of the two algorithms on the size of CDS. It shows PDA always constructs a CDS with smaller size than CDS-BD, which is much better than theoretical analysis we gave in Section 5. Therefore, we can conclude that PDA outperforms CDS-BD on size and on diameter with high probability as well.

6.3 Simulations Based on Different β



(a) Compare the Diameter of CDS



(b) Compare the Size of CDS

Figure 4: Simulations Based on Different β

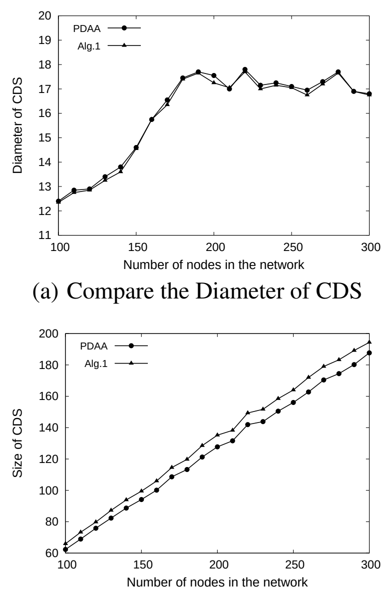
In the above simulations, β is fixed. In this section, we conduct the simulations with different values of β. We study the relationship between β and the size of CDS and the relationship between β and diameter of CDS. As the root selection will not affect the comparison, we randomly choose the root of CDS in this group of simulations. Results are shown in Fig. 4.

In Fig. 4(a), each line represents the diameter of CDS based on one of different values of β. When β is set to 1, PDA adds a shortest path from v to r if $D_{CDS}(r, v)$ is larger than $D_{SPT}(r, v)$.

Therefore, PDA with $\beta = 1$ returns a CDS with the smallest diameter. When β is set to 4, PDA will not cause the shortest paths to be added in, since PDA 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 PDA on diameter is 4. Thus, the CDS by PDA 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 PDA could be controlled by adjusting the values of β.

In Fig. 4(b), 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 $D_{SPT}(r, v)$, PDA 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. 4(a). In conclusion, the performance of PDA can be balanced depending on the value of β and the tradeoff between size and diameter is clear.

6.4 Simulations for km-CDS



(a) Compare the Diameter of CDS

(b) Compare the Size of CDS

Figure 5: Simulations for km-CDS

In this section, we are interested in evaluating the performance of Algorithm 1. We intend to illustrate that Algorithm 1 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 PDA with random root selection and β is set to 2 here. We take the 1-CDS generated using PDA as the input of Algorithm 1 afterwards, and we set $k = 2$ and $m = 1$.

Figure 5(a) compares the performance of Algorithm 1 and PDA 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 km-CDS. Therefore, Algorithm 1 enhances the fault tolerance of CDS without affecting its diameter greatly.

Meanwhile, as observed from Fig. 5(b), the size of km-CDS obtained from Algorithm 1 is certainly larger than 1-CDS by PDA. Specifically, the performance of the two algorithms is relatively proportional. As observed from our experiments, the size of km-CDS obtained from Algorithm 1 is almost 1.06 times the size of CDS returned by PDA. 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.

7. CONCLUSIONS

In this paper, we investigate the fault tolerant CDS problem with bounded diameter in wireless networks. We propose an approximation algorithm for a general case of the km-CDS, and two algorithms for 1-CDS, which could be applied into the solution of the km-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 [9].

8. REFERENCES

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

[2] S,-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The Broadcast Strom Problem in a Mobile ad hoc Network," *Proceedings of MOBICOM*, 1999.

[3] 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.

[4] 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.

[5] 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.

[6] B. Das, and V. Bharghavan, "Routing in ad hoc Networks Using Minimum Connected Domianting Sets", *International Conference on Communications*, 1997

[7] S. Basagni, M. Mastrogiovanni, A. Panconesi, and C. Petrioli, "Localized Protocols for ad hoc Clustering and Backbone Formation: A Performance Comparison," *IEEE Transactions on Parallel and Distributed Computing*, vol. 17, no.4, pp. 292–306, April 2006.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] C. E. Leiserson, R. L. Rivest, T. H. Cormen and C. Stein, *"Introduction to Algorithms"'*, MIT Press and McGraw-Hill Book Company, 1976.

[14] Y. Wu, F. Wang, M. T. Thai, and Y. Li, "Constructing k-Connected m-Dominating Sets in Wireless Sensor Networks", *in Proceedings of Military Communications Conference (MILCOM 2007)*, October, 2007.

[15] 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.

[16] F. Wang, M. T. Thai, 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)