Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 05-037

On the Construction of 2-Connected Virtual Backbones in Wireless Networks

Feng Wang, My T. Thai, and Ding-zhu Du

November 29, 2005

On the Construction of 2-Connected Virtual Backbone in Wireless Networks

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

***Abstract*— Virtual backbone has been proposed as the routing infrastructure to alleviate the broadcasting storm problem in ad hoc networks. Since the nodes in the virtual backbone need to carry other node's traffic, and node and link failure are inherent in wireless networks, it is desirable that the virtual backbone is fault tolerant. In this paper, we propose a new algorithm called Connecting Dominating Set Augmentation (CDSA) to construct a 2-connected virtual backbone which can resist the failure of one wireless node. We prove that CDSA has guaranteed quality, because the size of the CDSA constructed 2-connected backbone is within a constant factor of the optimal 2-connected virtual backbone size. Through extensive simulations, we demonstrate that CDSA can improve the fault tolerance of virtual backbone with only marginal extra overhead.**

I. Introduction

Ad-hoc networks are formed of wireless nodes without any underlying physical infrastructure. In order to enable data transfers in such networks, all the wireless nodes need to frequently *flooding* control messages thus causing a lot of redundancy, contentions and collisions (known as "broadcast storm problem" [1]). As a result, virtual backbone has been proposed as the routing infrastructure of ad hoc networks [2]. With virtual backbones, routing messages are only exchanged between the backbone nodes, instead of being broadcasted to all the nodes. Prior work [2] has demonstrated that virtual backbones could dramatically reduce routing overhead.

It is desirable that the virtual backbone is fault tolerant since the nodes in the virtual backbone need to carry other node's traffic. However, virtual backbones are often very vulnerable due to frequent node failure and link failure, which are inherent in wireless networks. Hence, how to construct a fault tolerant virtual backbone that continues to function during node or link failure is an important research problem.

We model an ad hoc network with the widely used Unit Disk Graph (UDG), assuming that each node has the same transmission range. Fault tolerant virtual backbone problem is formulated as follows: Given a UDG $G = (V, E)$ that models the network, find a subset of nodes B with minimum size and satisfies: i) each node not in B is dominated by at least m nodes in B, ii) B is k-node-connected. The nodes in B are called backbone nodes. In this paper, we study a special case of this problem for $m = 2$ and $k = 1$, i.e., to construct a 2-connected 1-dominating virtual backbone. This problem is essentially equal to 2-connected dominating set problem, which is a well known NP-hard problem [1]

We propose a centralized approximation algorithm called Connected Dominating Set Augmentation (CDSA) to construct a 2-connected virtual backbone that can accommodate the failure of one wireless node. To the best of our knowledge, this paper is the first one to address the 2-connected virtual backbone problem. The main idea is to first construct a connected dominating set, then augment it to be 2-connected by adding new nodes to the backbone. We prove that CDSA has a constant performance ratio of 120, thus the quality of CDSA is guaranteed.

Through extensive simulations, we demonstrate that our algorithm can improve the fault tolerance of virtual backbone with only marginal extra overhead. Specifically, 20% of all nodes are selected into the 2-connected virtual backbone when the average node degree is 20, which is only 5% higher than a connected virtual backbone. If the average node degree is 40, CDSA selects only 10% of the nodes into 2-connected virtual backbone.

The rest of this paper is organized as follows. Section

[1] In the remaining of this paper, we will use 2-connected (or 1-connected) virtual backbone, 2-connected 1-dominating (or 1-connected 1-dominating) virtual backbone, 2-connected (or connected) dominating set, 2-CDS (or CDS) interchangeably.

2 describes the related work. In section 3, we present CDSA algorithm and prove its correctness and analyze the time complexity. The approximation ratio of CDSA is analyzed in section 4. In section 5, we show the simulation results. Section 6 concludes this paper.

II. Related Work

In [3], 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 CDS's. The deterministic algorithm, k-Coverage condition, only works in very dense network and no upper bound on the size of resultant backbone is analyzed. The key difference between our work and their work is that we address the 2-connected 1-dominating virtual backbone problem. Our work is not a special case of [3] because we require different value of connectivity and domination. In addition, our algorithm can construct a smaller virtual backbone and has a constant approximation ratio.

Recent work on sensor deployment and repairing [4][5] addresses the problems of deploying a sensor network from scratch or repairing a sensor network by adding new sensors to satisfy a certain connectivity requirement. These problems can be mapped into minimum size k-connected Euclidean Steiner network problem [6][7]. In our study, the node location has already been decided. In other words, we need to choose a subset of nodes out of a pre-deployed network, instead of adding new nodes into the network.

In [8], Agrawal *et al.* propose approximations for General Steiner Network that addresses the problem of finding a subset of nodes of a given network that satisfies a certain edge connectivity requirement. In this paper, we focus on node connectivity. In addition, [9] [10] study how to construct 2-connected spanning subgraph with minimum weight or minimum number of edges. Our work differs from theirs in that we select a subset of nodes, but not a spanning subgraph.

In summary, none of the previous work address the k-connected m-domination problem. This paper is the first one to study 2-connected 1-dominating virtual backbone problem and to propose an efficient approximation with a guaranteed quality.

III. A New Algorithm for 2-Connected Virtual Backbone

In this section, we present a Connected Dominating Set Augmentation algorithm (CDSA) for constructing a 2-connected virtual backbone. We first introduce some definitions used in the algorithm, then present the detailed algorithm. Subsequently, we prove the correctness of the algorithm and analyze its time complexity.

A. Preliminaries

Before we introduce the algorithm, we need to give the following definitions: A *cut-vertex* of a connected graph G is a vertex x such that the graph $G - \{x\}$ is disconnected. A *block* is a maximal subgraph of G without cut-vertices. A *biconnected* graph is a graph without cut-vertices. Clearly a block with more than three nodes is a biconnected component. A *leaf block* of a connected graph G is a subgraph of G which is a block with only one cut-vertex.

B. Algorithm

The main idea of CDSA is: i) construct a small-sized Connected Dominating Set (CDS) as a starting point of the backbone, ii) iteratively augment the backbone by adding new nodes to connect a leaf block in the backbone to other block (or blocks), iii) the augmentation process stops when all backbone nodes are in the same block, i.e., the backbone nodes are 2-connected. The intuition of CDSA is that a 2-CDS is also a CDS, thus by constructing a small-sized CDS, we do not introduce any unnecessary nodes. Moreover, we only add nodes that are necessary to make the 2-connected portion larger. In total, the size of the 2-connected virtual backbone is bounded.

CDSA is illustrated in Algorithm 1. Given a network N modeled by a unit disk graph G, the algorithm consists of four main steps:

1) Use any CDS construction algorithm to construct a CDS C of G. We adopt the algorithm proposed in [11] instead of the well-known algorithm in [12] because [11] interleaves the process of finding Maximum Independent Set (MIS) and the process of connecting MIS. It is better than [12] in terms of CDS size.

2) Compute all the blocks in C using the standard algorithm in [13] which is based on the depth first search for computing the bi-connected components.
3) Calculate the shortest path in the original graph that satisfies the requirements: i) the path can connect a leaf block in C to other portion of C, ii) the path does not contain any nodes in C except the two endpoints. Then add all intermediate nodes in this path to C.
4) Repeat steps 2) and 3) until C is 2-connected.

Algorithm 1 Connected Dominating Set Augmentation Algorithm (CDSA) for constructing a 2-Connected virtual backbone

```
1:  INPUT: A 2-connected graph G = (V, E)
2:  OUTPUT: A 2-connected 1-dominating subgraph H of G
3:  C = computeCDS(G); /* C is a connected dominating set of G */
4:  B = computeBlocks(C); /* B is a list of all blocks in C */
5:  while (B contains more than one block)
6:     L = findLeafBlock(B); /* L is one leaf block */
7:     for (each node v ∈ L && v is not a cut-vertex)
8:       for (each node u ∈ V − L)
9:          Construct G' from G by deleting all nodes in C(except u and v) and all the edges incident to those nodes;
10:         if there exists at least one uv-path in G'
11:            P_uv = shortestPath(v,u, G');
          /* P is the shortest uv-path containing only non-backbone nodes as the intermediate nodes*/
12:         P = P ∪ P_uv;
13:      endfor
14:    endfor
15:    P_ij = the path with shortest length among all paths in P;
16:    C = C∪ intermediate nodes on P_ij;
17:    B = computeBlocks(C);
18: end while
```

Fig. 1 gives an example of the original network topology and the constructed 2-connected virtual backbone by CDSA. The network consists of 100 nodes which are randomly placed in a $1000 \times 1000m^2$ area. The transmission range is $250m$. Dark lines give the contour of the 2-connected virtual backbone, while the gray lines illustrate the original network topology. As we can see, the 2-connected virtual backbone is much smaller than the original topology.

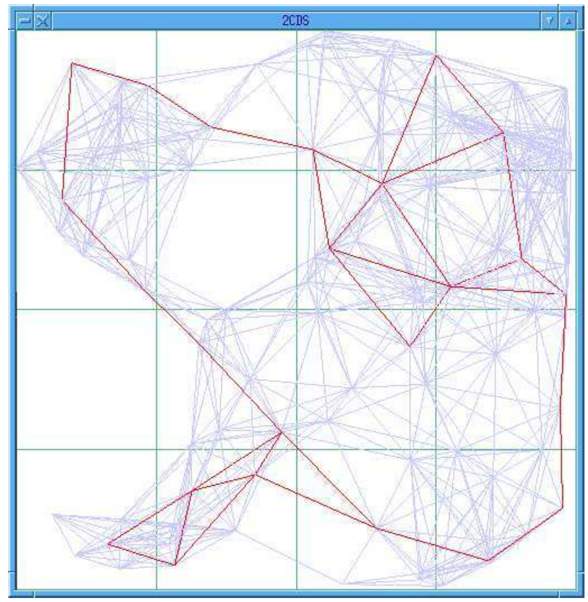


Fig. 1. An example of 2-Connected virtual backbone

C. Correctness

Now we prove that our algorithm guarantees a 2-connected virtual backbone. We argue that in Algorithm 1: i) in line 6, a leaf block always exists, ii) in line 14, a path P_{ij} always exists, and iii) the while loop from line 5 to line 17 will run in bounded time.

For i), it has been proved in [14] that if a graph G is not 2-connected, at least one block in G has precisely one cut-vertex of G, i.e, at least one leaf block exists. For ii), P_{ij} always existing means there always exists a path with only non-backbone nodes to connect a leaf block to another block if the CDS is not 2-connected. This is true because G is 2-connected. If we delete the cut-vertex from G, there must exist a path P in the original G connecting the leaf block to other blocks in CDS. In CDS, the only way to connect this leaf block to other blocks is through the cut-vertex, thus all nodes except the two endpoints on P are non-backbone nodes.

For iii), if the original graph G is connected, the number of blocks always decreases[2] by adding new nodes because at least the leaf block is merged into another block to form a bigger block without generating new blocks. Thus, suppose there are s blocks in the CDS, at most $s - 1$ steps are needed to build a 2-connected virtual backbone from the CDS.

[2]Note this is not true if G is not connected in the first place. In that case, the number of blocks might increase or be the same by adding new connectors

D. Time Complexity

Theorem 1: Suppose n is the number of nodes in the original graph, the time complexity of CDSA is $O(n^3)$.

Proof. Time complexity of constructing a CDS of the graph is $O(n)$, the first step needs $O(n)$ time. Suppose m is the number of edges in the original graph, the time complexity of computing blocks of the graph using Depth First Search scheme is $O(n+m)$, thus the second step needs $O(n^2)$ time since m 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 blocks) times. Therefore, the time complexity of CDSA is $O(n^3)$. □

IV. THEORETICAL ANALYSIS

In this section, we prove that CDSA has guaranteed quality. First we prove that at each augmenting step, limited number of nodes are added into the backbone, then we show that CDSA has a constant approximation ratio of 64.

Lemma 1: At most 10 new nodes are added into the backbone at each augmenting step (step 3 in the description of CDSA).

Proof. Suppose we mark the backbone nodes with BLACK and the remaining nodes with GRAY. Suppose L is a leaf block of CDS and w is the cut-vertex. Suppose nodes u and v, where $u \in L$ and $v \in V_{Backbone} - L$, are the two black nodes connected by the shortest possible path without any black nodes[3], there are three possibilities that nodes u and v are connected, that is u and v are connected by one connector, two connectors, and more than two connectors. Fig. 2.(a) illustrates the scenario of existing more than two connectors.

We claim that if the shortest path between uv called P_{uv} has more than two intermediate nodes, all intermediate nodes except x and y must be a neighbor of the cut-vertex w. This is true because: suppose P_{uv} is $u, x, ..., y, v$ and one of the intermediate nodes, let's say node z is not a neighbor of w(as illustrated in Fig. 2.(b)), z must have another black neighbor p or else z is not dominated by any CDS nodes, contradicting to CDS nodes dominate the network. If so, the path between pu or pv has a shorter distance than P_{uv}, which contradicts that P_{uv} has the shortest distance.

Now we show that there exists a path connecting a leaf block to another block with a limited number of intermediate nodes. The position of node u and v has four possibilities: i) nodes u and v are both neighbors of node w, ii) node u is a neighbor of w, but node v is not. iii) node v is a neighbor of w, but node u is not. iv) neither node u nor v are neighbors of node w.

Case i) is illustrated in Fig. 3. We can divide the neighborhood of w into 6 regions marked from 1 to 6 as shown in the graph by the dashed straight lines. The dashed circle is the neighborhood of the node in the center of the circle. All the interconnecting nodes are marked with gray color. Note that all nodes (not shown

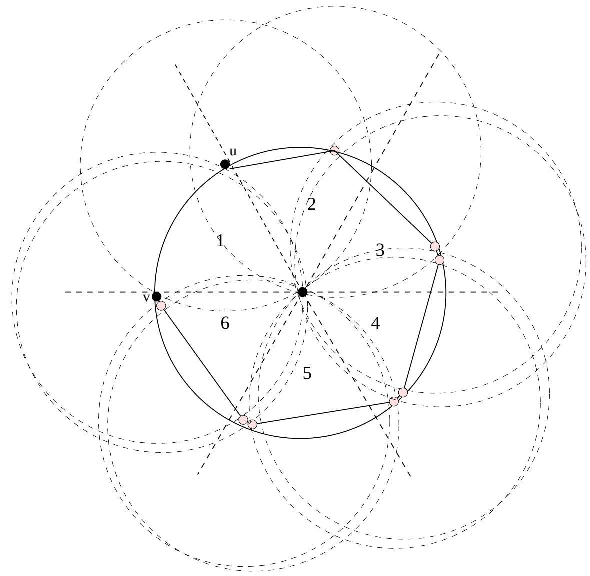


Fig. 3. w is the cut-vertex, node u and v are both neighbors of node w

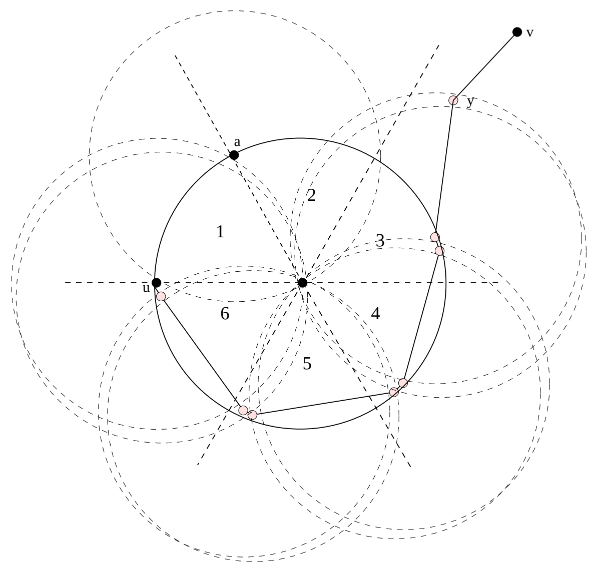
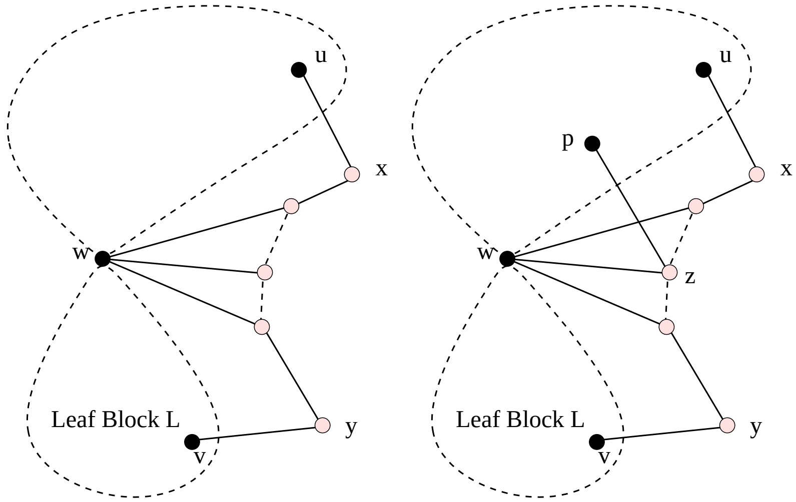


Fig. 4. w is the cut-vertex, u is in its leaf block is a neighbor of w, while v is not in the leaf block and v is not a neighbor of w

[3] As we have proved in correctness analysis, such a path always exists



(a) Scenarios for more than two connectors

(b) Path pv is smaller than path uv

Fig. 2. All intermediate nodes are neighbors of the cut-vertex

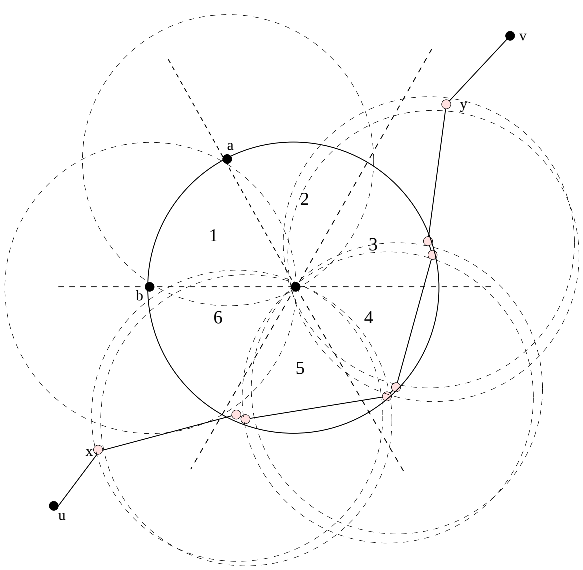


Fig. 5. w is the cut-vertex, u is in its leaf block while v is not in its leaf block and neither u nor v are neighbors of w

all on the figure) fall in the same region compose a clique because they are in each other's transmission range. Thus there are at most 2 interconnecting nodes in region 3, 4, and 5. In region 2, there are at most 1 interconnecting node because any nodes in region 2 are neighbors of u. For the same reason, there are at most 1 interconnecting node in region 6. Thus if u and v are in the transmission area of w, there are at most 8 interconnecting nodes between them.

Case ii) is illustrated in Fig 4. Since w is a cut-vertex, there must exist another black node, called a in the graph, that is a neighbor of w, otherwise the CDS is not connected anymore. Since path P_{uv} has the shortest length, then there could not exist interconnecting nodes in region 2, otherwise path P_{ua} has a shorter length than P_{uv}. In regions 3,4,6, there are at most two interconnecting nodes. In region 1, there are 1 interconnecting node. There might be another interconnecting node which is a neighbor of v but not of w, called y in Fig 4. Thus if only u is in the transmission range of w, there are at most 8 interconnecting nodes between u and v. Similar to case ii), case iii) has at most 8 interconnecting nodes.

Case iv) is illustrated in Fig. 5. Since w is a cut-vertex and neither u nor v is in its transmission range, there must exist two other black nodes, called a and b in Fig 5, that are two neighbors of w. Since path P_{uv} has the shortest length, then there could not exist interconnecting nodes in regions 1,2 and 6, otherwise either path P_{ab}, or P_{ub} or P_{va} has shorter length than P_{uv}. Again, in regions 3,4,5, there are at most 2 interconnecting nodes in each of them. There might be two other interconnecting nodes which are a neighbor of u and v but not of w respectively, called x and y in Fig 5. Thus if neither u nor v is in the transmission range of w, there are at most 8 interconnecting nodes between u and v.

In summary, we prove that for all possible scenarios, at most 8 interconnecting nodes are necessary to connect a leaf block to other blocks. $\square$

For simplicity, we introduce the following notations. Let OPT be a 2-connected 1-dominating set with minimum size, $MCDS$ be a connected dominating set with minimum size, CDS be the connected dominating set constructed from the first step of our first algorithm, and $2CDS$ be the 2-connected dominating set resulting from CDSA. We have following lemmas and theory.

Lemma 2: [11] $|CDS| \leq 8|MCDS| + 1$.

Lemma 3: $|MCDS| \leq |OPT| - 2$.

Proof. It is straightforward that $|MCDS| \leq |OPT|$ because a 2-CDS is also a CDS. Now We give an example illustrating that in the worst case $|MCDS| = |OPT| - 2$. As shown in Fig. 6, $2n$ nodes are aligned as around a cycle. Each node x has two neighbor nodes with id x-1 and x+1. MCDS includes all nodes except node 0 and 2n-1 thus $|MCDS| = 2n-2$. OPT includes all 2n nodes thus $|OPT| = 2n$. □

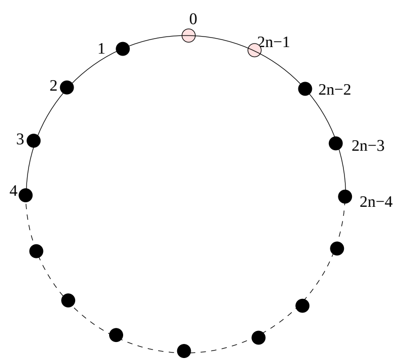


Fig. 6. The relationship between minimum connected dominating set and minimum 2-connected dominating set

Theorem 2: CDSA has a constant approximation ratio of 64.

Proof. In the CDSA algorithm, first a CDS is constructed, then in at most $|CDS| - 1$ steps and each step at most 8 nodes are added, we construct a 2-connected virtual backbone. Hence, $|2CDS| \leq |CDS| + 8*(|CDS| - 1) = 8|CDS| - 7$. From Lemmas 2, 3, we have $|CDS| \leq 8|MCDS| + 1 \leq 8(|OPT| - 2) + 1 = 8|OPT| - 15$. Thus $|2CDS| \leq 8(8|OPT| - 15) - 7 = 64|OPT| - 127$ for all $|OPT| \geq 2$. □

[15] further reduced the approximation ratio of CDS to 6.91. By applying their results, our algorithm has an approximation ratio of $6.91 + 8 * 6.91 = 62.19$. In the following section, we evaluate the performance of the proposed algorithm using simulations.

V. Performance Evaluation

In our simulation, we randomly generate various network topology of different settings in an $1000 \times 1000m^2$ region. Only topologies that are 2-connected are considered. For each setting, we perform the simulation for 500 time and compute the average value.

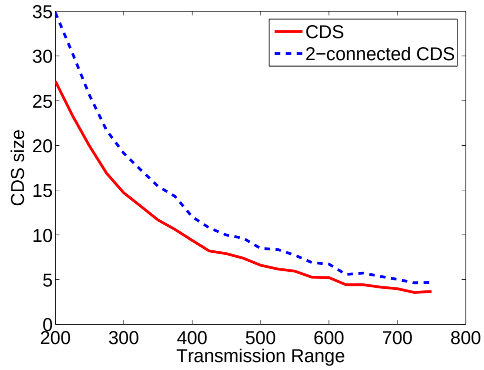
We carry out two set of simulations. In the first set, we fix the number of nodes and vary the transmission range to evaluate the impact of transmission range on the backbone size. For each transmission range, we also calculate the average node degree and evaluate the impact of node density on the backbone size. In the second set, we fix the transmission range and vary the number of nodes in the area to evaluate the impact of the number of the nodes in the network on the backbone size. For each set of simulations, both connected dominating set and 2-connected dominating set size are recorded and then compared to evaluate the effectiveness of our algorithm in terms of the backbone size.

A. Impact of Transmission Range and Node Density

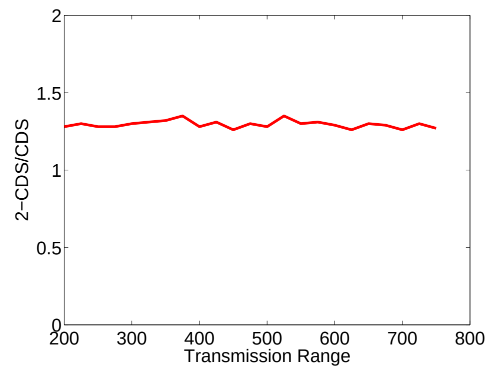
In this simulation, we randomly place 100 nodes in an $1000 \times 1000m^2$ region. The node transmission range varies from $200m$ to $750m$.

Fig. 7 shows the impact of transmission range and node degree on the backbone size. In Fig. 7.(a), x-axis is the transmission range and y-axis is the backbone size. The solid line is the average size of 1-connected backbone, and the dashed line is the average size of 2-connected backbone. Clearly, 2-CDS size is only a little greater than the 1-CDS size. For example, when the transmission range is $500m$, which is half of the region edge, the 2-connected backbone size is 10 out of 100 nodes, which is only 3 nodes more than the size of the 1-connected backbone. Another observation is that as the transmission range increases from $200m$ to $750m$, the size of both 1-connected and 2-connected backbone decrease. In addition, the difference between their size gets smaller as the transmission range increases. The underlying reason is that as the transmission range increases, node density increases, smaller number of nodes can dominate the whole network.

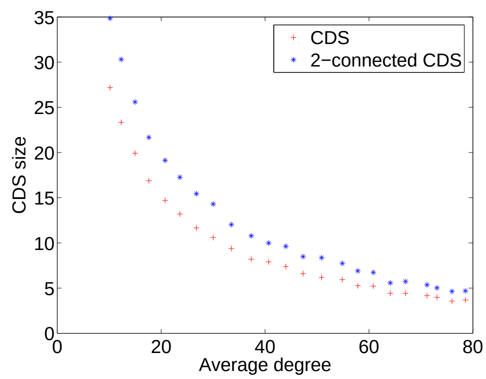
To evaluate the number of nodes that augment CDS into 2-CDS, which is the overhead for constructing a fault tolerant virtual backbone, we show the ratio of 2-CDS over CDS in Fig. 7.(b). As can be seen, the performance of CDSA algorithm is consistent under different network topologies since the ratio of 2-CDS over CDS is constantly around 1.3. In other words, using CDSA, the overhead is predictable because it is always around 0.3 of the CDS size. This is interesting because in theoretical analysis, we prove that in worst case, $8*|CDS|$ nodes are added to augment a CDS into a



(a) backbone size for different transmission range



(b) ratio of 2cds over cds for different transmission range



(c) backbone size for different node degree

Fig. 7. The effect of transmission range and node density to the backbone size

2-CDS, while the simulation shows under average case, actually only $0.3 * |CDS|$ augmenting nodes are needed. In fact, at each step to augment a leaf block to other blocks, in 99% times only one or two interconnecting nodes are needed.

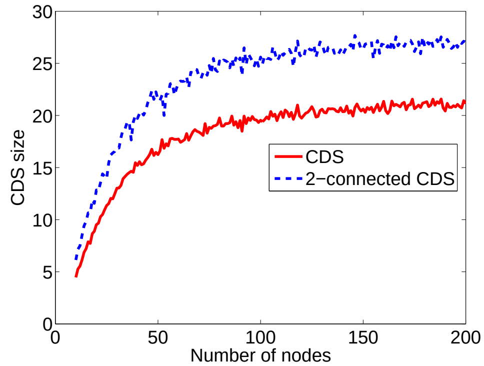
To further understand the effect of node density, we calculate the average node degree for each transmission range and illustrate the backbone size for different average node degrees in Fig. 7.(c). It is shown that 20% of all nodes are selected into the 2-connected virtual backbone when the average node degree is 20, which is only 5% higher than a 1-connected virtual backbone. If the average node degree is 40, our algorithm selects only 10% of the nodes into 2-connected virtual backbone.

B. Impact of Node Size

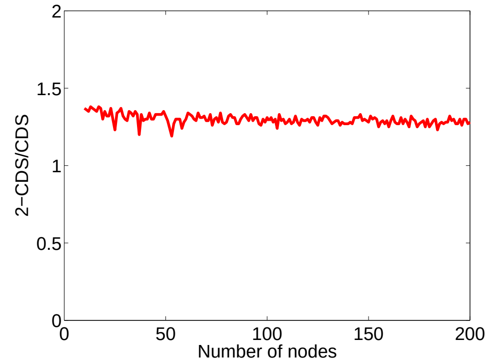
In this simulation, we fix the transmission range at $250m$, which is a quarter to the area edge ($1000 \times 1000m^2$). Node size varies from 10 to 200.

Fig. 8 shows the impact of node density on the size backbone. In Fig. 8.(a), x-axis is the number of nodes in the network and y-axis is the backbone size. When the number of nodes in the network increases from 10 to 50, CDS and 2-CDS size increase significantly. However, the CDS and 2-CDS size increase much slower when the number of nodes in the network increases from 50 to 200 and the backbone size keeps almost the same when node size is from 150 to 200. This implies that when node number is greater than 150, in average, around 20 nodes with transmission range at 250 can always dominate this $1000 \times 1000m^2$ area and 26 nodes are enough to construct a 2-connected virtual backbone. This shows that our algorithm has good scalability.

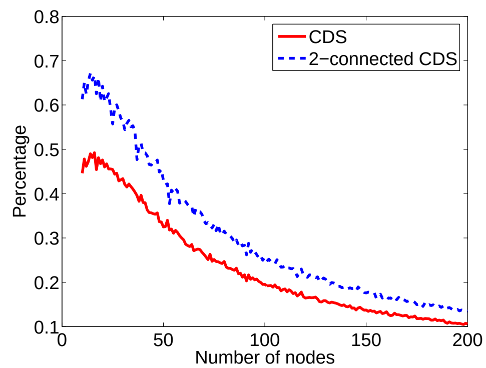
Fig. 8.(b) shows the ratio of CDS over 2-CDS for different node numbers. This figure is consistent with Fig. 7.(b) in that the ratio is constant at around 1.3. This confirms that the CDS size and the number of nodes that



(a) backbone size



(b) ratio of 2-CDS over CDS



(c) percentage of nodes that are selected into the backbone

Fig. 8. The impact of node size on the backbone size

are chosen by CDSA for augmenting a CDS to a 2-CDS tend to be correlated.

Fig. 8.(c) shows the percentage of the nodes in the network that are chosen in the backbones. We find that although the absolute value of nodes in CDS and 2-CDS increase as the number of nodes in the network increases, the percentages of nodes selected in CDS and 2-CDS decrease. For example, when deploying 100 nodes with a transmission range of 250 in the $1000 \times 1000m^2$ region, 20% of the nodes are chosen into CDS, and 25% of the nodes are chosen into 2-CDS. While when deploy 200 nodes with the same transmission range in the same region, only 10% of the nodes are chosen into CDS and 15% of the nodes are chosen into 2-CDS. This is reasonable because the more nodes deployed in a region, the higher the node density, therefore the smaller percentage of nodes are selected into CDS and 2-CDS.

In summary, CDSA can improve the fault tolerance of virtual backbone with only marginal extra overhead. The performance of CDSA algorithm is consistent under different network topologies and the overhead is predictable and usually is 0.3 of the CDS size.

VI. Conclusion

In this paper, we studied the k-connected m-dominating problem and propose a new algorithm called Connecting Dominating Set Augmentation (CDSA) to construct a 2-connected virtual backbone. We prove that CDSA has constant approximation ratio, thus has guaranteed quality. Through extensive simulations, we demonstrate that CDSA can improve the fault tolerance of virtual backbone with only marginal extra overhead.

Our future work will focus on two directions: i) propose distributed and localized algorithm for 2-connected virtual backbone, ii) propose general algorithm for any k and m.

REFERENCES

[1] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proceedings of MOBICOM*, 1999.

[2] P. Sinha, R. Sivakumar, and V. Bharghavan, "Enhancing ad hoc routing with dynamic virtual infrastructures,," in *Proceedings of Infocom*, 2001.

[3] F. Dai and J. Wu, "On constructing k-connected k-dominating set in wireless networks," in *IEEE International Parallel & Distributed Processing Symposium*, 2005.

[4] J. L. Bredin, E. D. Demaine, M. Hajiaghayi, and D. Rus, "Deploying sensor networks with guaranteed capacity and fault tolerance," in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)*, 2005, pp. 309–319.

[5] H. Koskinen, J. Karvo, and O. Apilo, "On improving connectivity of static ad-hoc networks by adding nodes," in *Mediterranean Ad Hoc Networks*, 2005.

[6] D.Chen, D.Z.Du, X.D.Hu, G.H.Lin, L.Wang, and G.Xue, "Approximations for steiner trees with minimum number of steiner points," *Journal of Global Optimization*, vol. 18, no. 1, pp. 17–33, 2000.

[7] D.Du, L.Wang, and B.Xu, "The euclidean bottleneck steiner tree and steiner tree with minimum number of steiner points," *Lecture Notes in Computer Science*, vol. 2108, pp. 509–518, 1991.

[8] A. Agrawal, P. Klein, and R. Ravi, "When trees collide: an approximation algorithm for the generalized steiner problem on networks," in *Technical Report CS-90-32, Brown University*, 1991.

[9] S. Khuller and U. Vishkin, "Biconnectivity approximations and graph carvings," *Journal of the ACM*, pp. 759–770, 1994.

[10] S. Khuller and B. Raghavachari, "Improved approximation algorithms for uniform connectivity problems," in *Proceedings of the 27th annual ACM symposium on Theory of computing (STOC)*, 1995.

[11] M. Min, F. Wang, D.-Z. Du, and P. M. Pardalos, "A reliable virtual backbone scheme in mobile ad hoc networks," in *Proceedings of IEEE International conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2004.

[12] P.-J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," in *Proceedings of 21th Annual Joint Conference of the IEEE Computer and Communication Societies(InfoCom)*, 2002.

[13] R. Tarjan, "Depth first search and linear graph algorithms," *SIAM Journal of Computing*, vol. 1, no. 2, pp. 146–160, 1972.

[14] D. B. West, "Introduction to graph theroy, second edition," 2001.

[15] S. Funke, A. Kesselman, U. Meyer, and M. Segal, "A simple improved distributed algorithm for minimum cds in unit disk graphs," in *Proceedings of 1st IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2005.