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

Feng Wang, *Member, IEEE,* My T. Thai, *Member, IEEE,* and Ding-Zhu Du, *Member, IEEE*

***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 show that CDSA has guaranteed quality by proving that 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 in practice, CDSA can build a 2-connected virtual backbone with only small overhead.**

***Index Terms*—Virtual backbone, fault tolerance, k-connectivity, dominating set.**

## 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 *flood* 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 a wireless network with the widely used Unit Disk Graph (UDG), assuming that each node has the same transmission range. [1] 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) $B$ is $k$-node-connected, ii) each node not in $B$ is dominated by at least $m$ nodes in $B$. The nodes in $B$ are called backbone nodes. In this paper, we study a special case of this problem for $k = 2$ and $m = 1$, 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 [2]

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 72, thus the quality of CDSA is guaranteed.

Through extensive simulations, we demonstrate that our algorithm can construct a 2-connected virtual backbone with small 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 2 describes the related work. In section 3, we present CDSA algorithm and prove its correctness and analyze its 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 literature, most of the virtual backbone construction algorithms [4][5][6][7][8][9] are special cases of the k-connected m-dominating virtual backbone problem, addressing the case of k=1 and m=1.

In [10], 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

Manuscript received February 2006; revised May 17, 2007; accepted October 15, 2007. The associate editor coordinating the review of this paper and approving it for publication was Q. Zhang.

F. Wang is with the Mathematical and Natural Sciences Division, Arizona State University, PO Box 37100, Mail Code: 2352, Phoenix, AZ, 85069 (e-mail: fwang25@asu.edu).

M. T. Thai is with the Computer and Information Science and Engineering Department, University of Florida, Gainesville, FL 32611.

D.-Z. Du is with the Department of Computer Science and Engineering, University of Texas at Dallas, Richardson, TX 75083.

Digital Object Identifier 10.1109/TWC.2009.051053

[1]This is a simplified but widely used model in most of the approximation algorithms for virtual backbone. Recently, [3] proposed the first constant approximation algorithm for constructing CDS in Disk Graph with Bidirectional Links.

[2]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.

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, has no upper bound on the size of resultant backbone. 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 [10] 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 [11][12] 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 [13][14]. 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 [15], 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, [16] [17] 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 introducing 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 and contains one cut-vertex of $G$.

### 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 [7] instead of the well-known algorithm in [4] because [7] interleaves the process of finding Maximum Independent Set (MIS) and the process of connecting MIS. It is better than [4] in terms of CDS size.
2) Compute all the blocks in $C$ using the standard algorithm in [18] 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

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

Fig. 1 gives an example of the original network topology and the constructed 2-connected virtual backbone by CDSA.

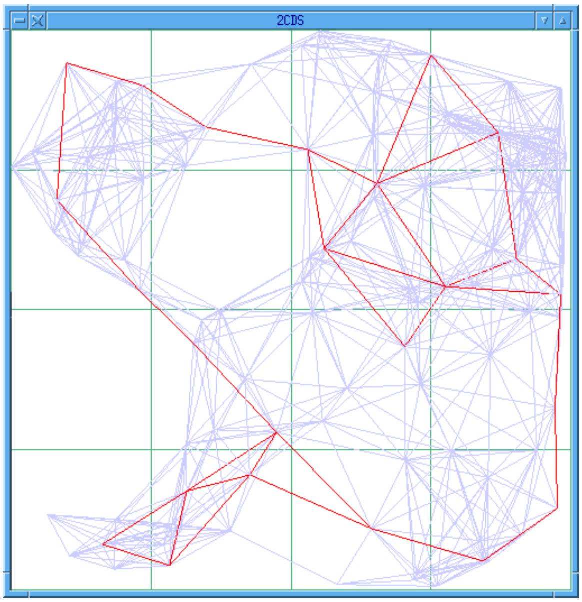



Fig. 1. An example of 2-Connected virtual backbone.

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.

### 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 15, a path $P_{ij}$ always exists, and iii) the while loop from line 5 to line 18 will run in bounded time.

For i), it has been proved in [19] 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 exists 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[3] 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.

### D. Time Complexity

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

[3]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

*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 72.

*Lemma 1:* At most 8 new nodes are added into the backbone at each augmenting step (step 3 in the description of CDSA).

*Proof.*

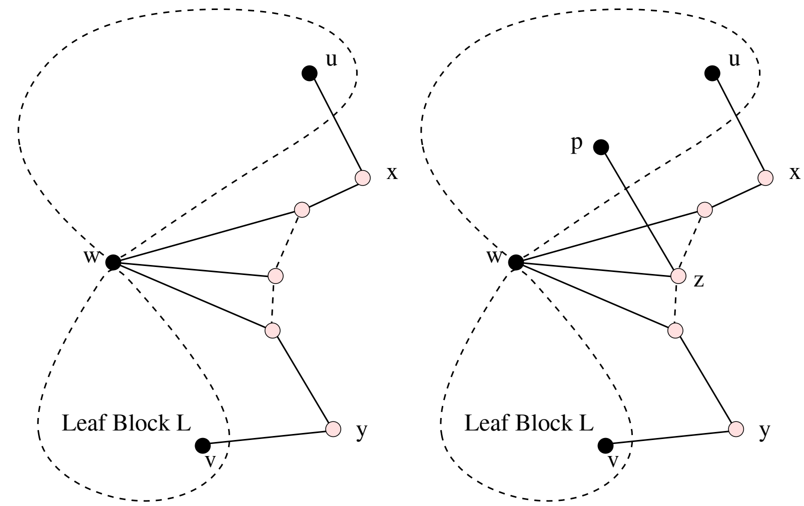
Suppose we mark the backbone nodes with BLACK and the remaining nodes with GRAY. Suppose $L$ is a leaf block of $CDS$ and $w$ is the cut-vertex. Suppose nodes $u$ and $v$, where $u \in L$ and $v \in V_{Backbone} - L$, are the two black nodes connected by the shortest possible path without any black nodes[4], 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 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

[4]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.

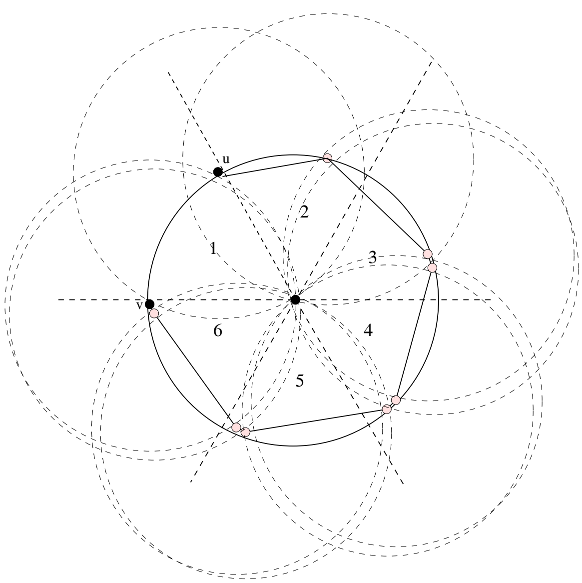


Fig. 3. $w$ is the cut-vertex, node $u$ and $v$ are both neighbors of node $w$.

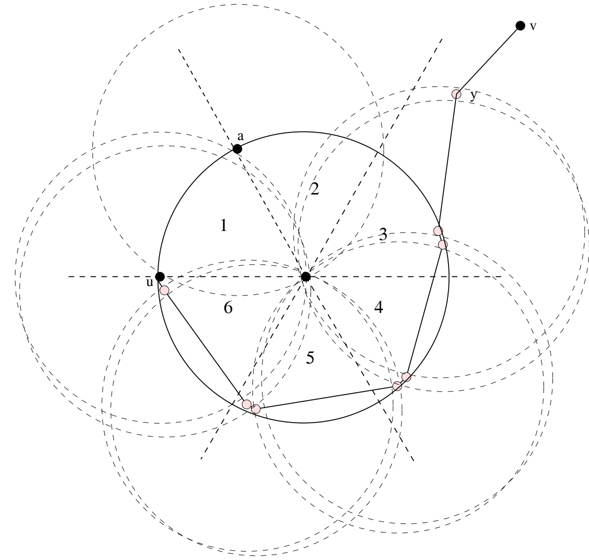


Fig. 4. $w$ is the cut-vertex, $u$ is in its leaf block is a neighbor of $w$, while $v$ is not in the leaf block and $v$ is not a neighbor of $w$.

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. □

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

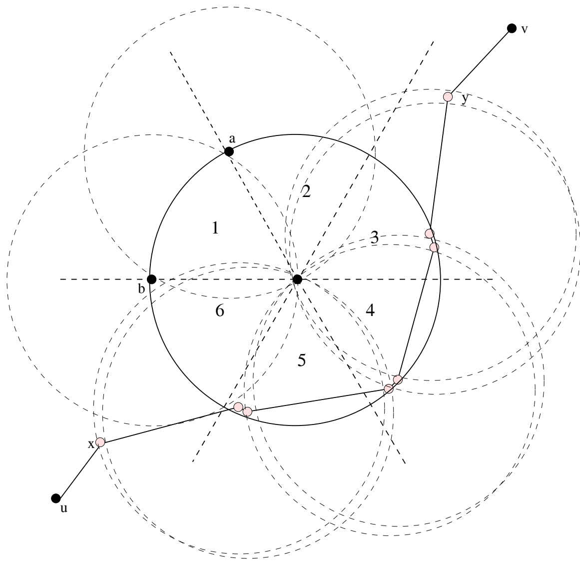



Fig. 5. $w$ is the cut-vertex, $u$ is in its leaf block while $v$ is not in its leaf block and neither $u$ nor $v$ are neighbors of $w$.

dominating set resulting from CDSA. We have following lemmas and theory.

*Lemma 2:* [7] $|CDS| \leq 8|MCDS| + 1$.

*Lemma 3:* $|MCDS| \leq |OPT|$.

*Proof.* It is straightforward that $|MCDS| \leq |OPT|$ because a 2-CDS is also a CDS. □

*Theorem 2:* CDSA has a constant approximation ratio of 72.

*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, $|2CDS| \leq |CDS| + 8 * (|CDS| - 1) = 9|CDS| - 8$. From Lemmas 2, 3, we have $|CDS| \leq 8|MCDS| + 1 \leq 8|OPT| + 1$. Thus $|2CDS| \leq 9(8|OPT| + 1) - 8 = 72|OPT| + 1$. □

[20] 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. 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 four sets 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 these two sets 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.

In the third set, scalability is evaluated for different networks with similar network density. In the fourth set, we present the number of 2-dominated non-backbone nodes to measure the domination of the virtual backbone constructed by CDSA.

### A. Impact of Transmission Range and Node Density to Backbone Size

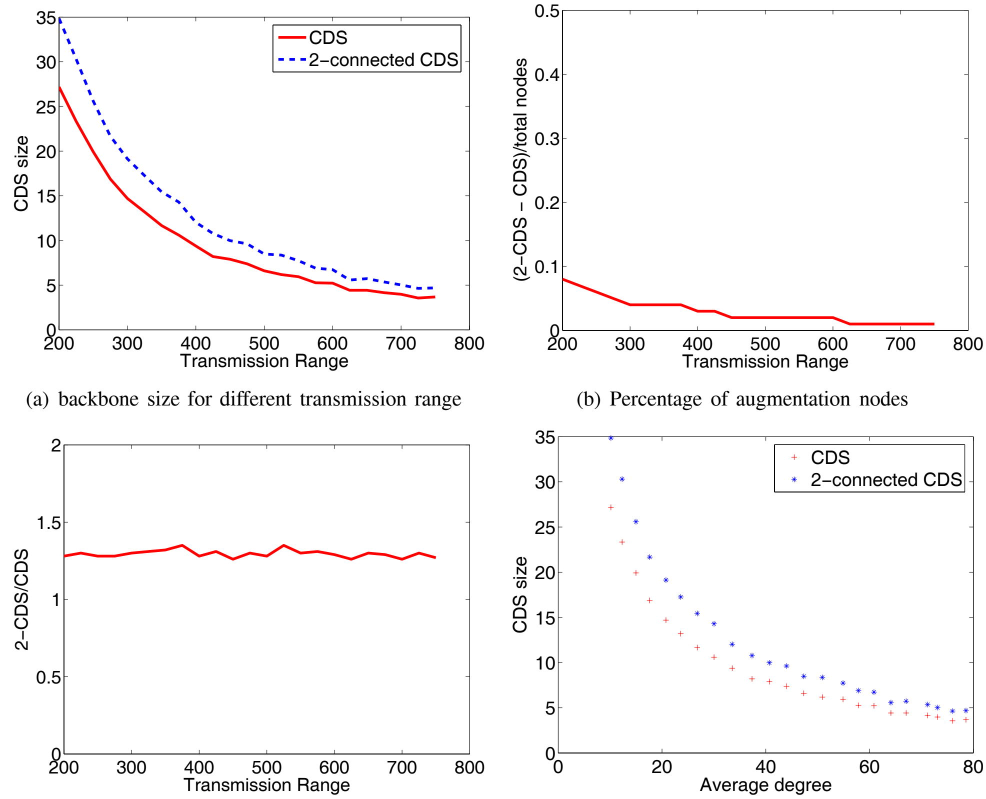
In this simulation, we randomly place 100 nodes in an $1000 \times 1000m^2$ region. The node transmission range varies from $200m$ to $750m$.

Fig. 6 shows the impact of transmission range and node degree on the backbone size. In Fig. 6.(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 2-connected virtual backbone, we define the percentage of augmentation nodes as $\frac{2CDSsize - CDSsize}{totalnodes}$ and show the percentage of augmentation nodes in Fig. 6.(b). It is clear that the number of augmentation nodes is very small. For example, only 8 new nodes are needed to augment a CDS to a 2-CDS for sparse network when transmission range is $200m$ (average node degree is around 10), and only 2 nodes are needed for dense network when transmission range is greather than $450m$ (average node degree is above 40). This shows that the overhead of CDSA to construct 2-connected virtual backbone is very small.

We also measure the overhead in terms of the ratio of 2-CDS over CDS in Fig. 6.(c). 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 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. 6.(d). 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,




(a) backbone size for different transmission range

(b) Percentage of augmentation nodes

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

(d) backbone size for different node degree

Fig. 6. The effect of transmission range and node density to the backbone size.

our algorithm selects only $10\%$ of the nodes into 2-connected virtual backbone.

### *B. Impact of Node Size to Backbone 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.

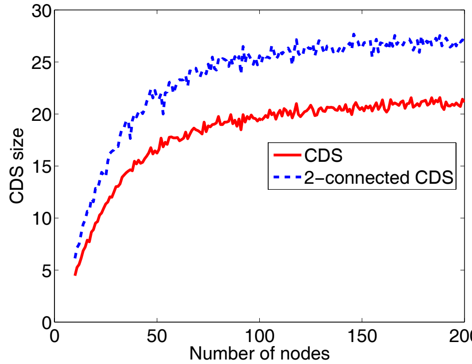
Fig. 7 shows the impact of node density on the size backbone. In Fig. 7.(a), x-axis is the number of nodes in the network and y-axis is the backbone size. When the number of nodes in the network increases from 10 to 50, CDS and 2-CDS size increase significantly. However, the CDS and 2-CDS size increase 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. 7.(b) shows the ratio of CDS over 2-CDS for different node numbers. This figure is consistent with Fig. 6.(c) in that the ratio is constant at around 1.3. This confirms that the CDS size and the number of nodes that are chosen by CDSA for augmenting a CDS to a 2-CDS tend to be correlated.

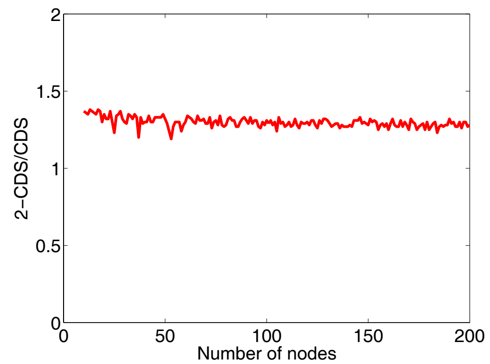
Fig. 7.(c) shows the percentage of the nodes in the network that are chosen in the backbones. We find that although the absolute value of nodes in CDS and 2-CDS increase as the number of nodes in the network increases, the percentages of nodes selected in CDS and 2-CDS decrease. For example, when deploying 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.

### *C. Scalability*

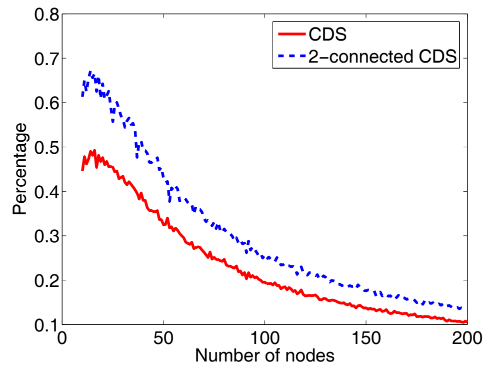
In this simulation, we show the scalability of CDSA algorithm for a certain network density. Since we randomly place nodes in a region, there is no guarantee that the generated network has a fixed network density. To evaluate the scalability of our scheme, we fix the transmission range to $250m$, and enlarge the area edge when the number of nodes in the area increases. The relation between the number of nodes (N) and area edge (E) is: $E = \sqrt{N} * 100$. For example, if the number of nodes is 100, then the area edge is $1000m$. If the number of nodes is 400, then the area edge is $2000m$. Our simulation results show that by varying the number of nodes from 100 to 400, and adjusting the area edge accordingly, we can generate random networks with average node degrees varying between 14.9 and 17.4. Fig. 8 shows the percentage of 2-CDS nodes for different networks with similar network density. x-axis is the number of nodes in the network and y-axis is the percentage of 2-CDS nodes. For node number 100(area edge $1000m$),




(a) backbone size




(b) ratio of 2-CDS over CDS




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

Fig. 7. The impact of node size on the backbone size.

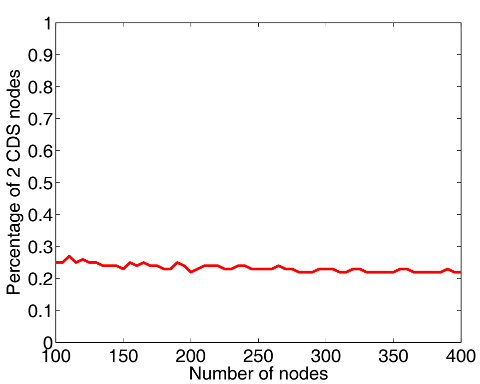



Fig. 8. The scalability of CDSA algorithm.

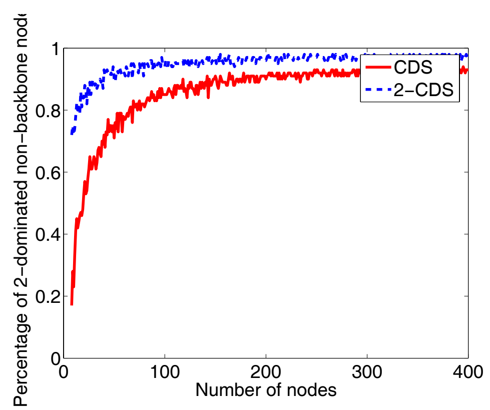



Fig. 9. The measurement of 2-dominated non-backbone nodes.

$25\%$ nodes are selected in the 2-CDS, and for node number $400$(area edge $1000m$), $22\%$ nodes are selected. This means that as long as the network density is stable, the percentage of 2-CDS nodes remains stable as the total number of nodes in the network increases, i.e., CDSA has good scalability.

### *D. Measurement of Fault Tolerance*

We measure the fault tolerance from two aspects: first, connectivity of the constructed virtual backbone. CDSA guarantees the 2-connectivity of the constructed virtual backbone, which means the virtual backbone is guaranteed to be connected under the failure of one backbone node. This is very important since virtual backbone nodes carry other nodes traffic. If the virtual backbone is broken, the whole network is broken.

Second, domination of the constructed virtual backbone. 2-domination is a desirable feature of a virtual backbone, since the broken of one backbone node does not separate its dominated non-backbone nodes from the rest of the network. Although our algorithm cannot guarantee 2-domination of the non-backbone nodes, it is still interesting to show how many non-backbone nodes are 2-dominated by the virtual backbone constructed by CDSA. This is illustrated in Fig. 9. The first observation is that most non-backbone nodes are actually 2-dominated by the 2-CDS nodes, and the percentage of 2-dominated non-backbone nodes increases as node density increases. For example, when there are more than $40$ nodes (average degree is above 6) in the network, more than 90% non-backbone nodes are 2-dominated by backbone nodes. When there are more than 100 nodes (average degree is 15) in the network, more than 95% non-backbone nodes are 2-dominated by backbone nodes. Another observation is that 2-CDS has better domination than CDS. The gap between 2-CDS and CDS is fairly large for very sparse network, for example, when there are only 10 nodes in the area (average degree is 3.5), only 23% of the non-backbone nodes of CDS are 2-dominated while 71% of the non-backbone nodes of 2-CDS are 2-dominated. For relatively dense network, the percentage of 2-dominated non-backbone nodes of 2-CDS is constantly about 6% higher than that of CDS.

In summary, CDSA can generate a 2-connected virtual backbone with small backbone size. 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 construct a 2-connected virtual backbone with small 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 *Proc. MOBICOM*, 1999.

[2] P. Sinha, R. Sivakumar, and V. Bharghavan, "Enhancing ad hoc routing with dynamic virtual infrastructures,," in *Proc. 20th Annual Joint Conference of the IEEE Computer and Communication Societies(InfoCom)*, 2001.

[3] M. Thai, F. Wang, D. Liu, S. Zhu, and D.-Z. Du, "Connected dominating sets in wireless networks with different transmission ranges," *IEEE Trans. Mobile Computing*, vol. 6, no. 7, pp. 721–730, 2007.

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

[5] K. M. Alzoubi, P.-J. Wan, and O. Frieder, "Message-optimal connected dominating sets in mobile ad hoc networks," in *Proc. 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2002.

[6] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *Proc. 3rd ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, 1999.

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

[8] Y. Li, M. T. Thai, F. Wang, C.-W. Yi, P. Wan, and D.-Z. Du, "On greedy construction of connected dominating sets in wireless networks," *J. Wireless Commun. and Mobile Computing*, Special Issue on Ad Hoc Networks, vol. 5, no. 88, pp. 927–932.

[9] F. Wang, M. Min, Y. Li, and D.-Z. Du, "On the construction of stable virtual backbones in mobile ad-hoc networks," in *Proc. IEEE International Performance Computing and Communications Conference (IPCCC)*, 2005.

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

[11] J. L. Bredin, E. D. Demaine, M. Hajiaghayi, and D. Rus, "Deploying sensor networks with guaranteed capacity and fault tolerance," in *Proc. 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2005, pp. 309–319.

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

[13] 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," *J. Global Optimization*, vol. 18, no. 1, pp. 17–33, 2000.

[14] 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.

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

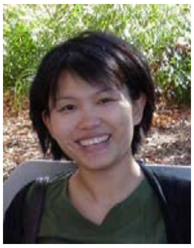
[16] S. Khuller and U. Vishkin, "Biconnectivity approximations and graph carvings," *J. of the ACM*, pp. 759–770, 1994.

[17] S. Khuller and B. Raghavachari, "Improved approximation algorithms for uniform connectivity problems," in *Proc. 27th Annual ACM Symposium on Theory of Computing (STOC)*, 1995.

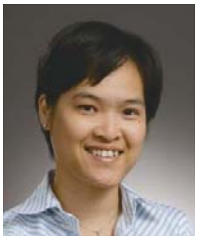
[18] R. Tarjan, "Depth first search and linear graph algorithms," *SIAM J. Computing*, vol. 1, no. 2, pp. 146–160, 1972.

[19] D. B. West, *Introduction to Graph Theory*, 2nd ed., 2001.

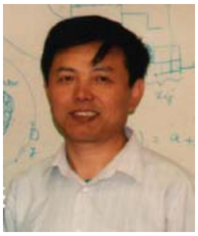
[20] S. Funke, A. Kesselman, U. Meyer, and M. Segal, "A simple improved distributed algorithm for minimum cds in unit disk graphs," in *Proc. 1st IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2005.

**Feng Wang** received the BS degree from Wuhan University in 1996, MS degree from Beijing University in 1999, and PhD degree from the University of Minnesota, Twin Cities in 2005, all in computer science. She is currently an assistant professor in the Mathematical and Natural Sciences Division at Arizona State University. Her research interests include wireless networks, network security, social network, computational biology, and combinatorial optimization. She is a member of the IEEE.



**My T. Thai** received a BS degree in computer science and a BS degree in mathematics from Iowa State University in 1999. She obtained her Ph.D. degree in computer science from the University of Minnesota, Twin Cities, in 2005. She is now an assistant professor in the Department of Computer and Information Sciences and Engineering at the University of Florida. Her main research interests include combinatorics, algorithms, wireless networks, and computational biology. Her work has covered many areas of wireless networks and computational biology, including routing protocols, coverage in sensor networks, broadcast tree, virtual backbone, group testing, and non-unique probe selection. She is a member of the IEEE.



**Ding-Zhu Du** received the MS degree from the Chinese Academy of Sciences in 1982 and the PhD degree from the University of California at Santa Barbara under the supervision of Professor Ronald V. Book in 1985. Before settling at the University of Texas at Dallas, he worked as a professor in the Department of Computer Science and Engineering at the University of Minnesota. He also worked at the Mathematical Sciences Research Institute in Berkeley for one year, in the Department of Mathematics at the Massachusetts Institute of Technology for one year, and in the Department of Computer Science at Princeton University for one and a half years. Currently, he has published more than 140 journal papers and several books. He is the editor-in-chief of the Journal of Combinatorial Optimization and is also on the editorial boards for several other journals. Thirty Ph.D. students have graduated under his supervision. He is a member of the IEEE.