A Localized Backbone Renovating Algorithm for Wireless Ad Hoc and Sensor Networks

Kai Xing*, Shuo Zhang*, Lei Shi ‡, Haojin Zhu †, Yuepeng Wang*,
*University of Science and Technology of China, Anhui, China 230027
Email: {kxing}@ustc.edu.cn, {zshuo, wangyuep, pfhu, tbgu, weihao, fjzhang, yiliang}@mail.ustc.edu.cn
†Shanghai Jiao Tong University, Shanghai, China
Email: zhu-hj@sjtu.edu.cn
‡Hefei University of Technology, Anhui, China
Email: thunder10@163.com

***Abstract*—In this paper we propose and analyze a localized backbone renovating algorithm (LBR) to renovate a broken backbone in the network. This research is motivated by the problem of virtual backbone maintenance in wireless ad hoc and sensor networks, where the coverage area of nodes are disks with identical radii. According to our theoretical analysis, the proposed algorithm has the ability to renovate the backbone in a purely localized manner with a guaranteed connectivity of the network, while keeping the backbone size within a constant factor from that of the minimum CDS. Both the communication overhead and computation overhead of the LBR algorithm are $O(k)$, where k is the number of nodes broken or added. We also conduct extensive simulation study on connectivity, backbone size, and the communicaiton/computation overhead. The simulation results show that the proposed algorithm can always keep the renovated backbone being connected at low communication/computation overhead with a relatively small backbone, compared with other existing schemes. Furthermore, the LBR algorithm has the ability to deal with arbitrary number of node failures and additions in the network.**

***Index Terms*—maximal independent set, backbone renovating**

I. Introduction

Virtual backbone is an important issue in wireless ad hoc and sensor networks and has been widely applied in various research domains such as routing, coverage, interference management, energy saving, etc., e.g., CDS as a virtual backbone [1]–[7], or for coverage [8], [9] and network topology control [10]–[15] for saving energy, reducing signal interference, etc. In general, most of these approaches end up of forming a dominating set as a backbone, through which each node in the network either is on the backbone or has at least a backbone node as its neighbors.

However, in wireless ad hoc and sensor networks the network topology keeps changing all over the time due to node failures, additions, or periodically switch on/off. It is very likely that the constructed backbone quickly becomes defective. The dynamism of the network poses a great challenge for backbone management/maintenance. Therefore, it is imperative to provide an effective solution for backbone maintenance.

In maintaining a backbone in ad hoc and sensor networks, the localized approach is most favorable due to its efficiency and its support to scalable design and network dynamism. In this paper, we propose and analyze a localized backbone renovating (LBR) algorithm. This algorithm explores the geometric properties of unit-disk graphs and renovates a backbone at an ultra low $O(k)$ computation overhead and $O(k)$ communication overhead, where k is the number of nodes broken or added.

The major contributions of this paper are identified below:

- In this paper, we proposes a purely localized backbone renovating algorithm (LBR) with ultra low communication and computation overhead.
- The proposed LBR algorithm has the capability of providing a renovated backbone with guaranteed connectivity of the network. It is proved that unless the network is no longer connected, the proposed algorithm can always keep the renovated backbone being connected.
- The proposed LBR algorithm has the ability to deal with arbitrary number of node failures and additions in the networks in a purely localized manner.
- We have conducted extensive simulation study under various scenarios. The results show that the LBR algorithm can effectively repair the backbone in an efficient manner compared with other existing centralized and localized approaches.

The rest of the paper is organized as follows: Section II presents the related works. The preliminaries, models and assumptions are introduced in Section III. Section IV further derives some geometric properties of unit-disk graphs that serve as the basis of the localized backbone renovating algorithm. Section V is devoted to the localized backbone renovating algorithm design. Section VI provides our theoretical analysis on LBR. Section VII reports our simulation study and comparison results, followed by the conclusions in Section VIII.

II. Related Work

In the following we briefly overview the related works of backbone construction and maintenance in unit-disk graph and summarize the most related research.

Finding a CDS in the network is a popular approach for backbone construction. The study of NP-Completeness of

978-1-4673-5946-7/13/$31.00 ©2013 IEEE

finding an MCDS in general graphs is proposed in [16]. This problem remains NP-hard in unit-disk graphs [17]. For a detailed literature survey, we refer the readers to [18] and the references therein.

Wan proposes the first MIS based CDS construction algorithm [19]–[21]. A similar approach is proposed in [22] to construct and connect an MIS simultaneously. A PTAS for MCDS in unit-disk graphs is proposed in [23]. [24] proposed a distributed algorithm for producing a tree-like backbone with $O(n)$ computation complexity and $O(nlogn)$ communication overhead. [25] selects the nodes with wider communication range, more energy, etc., then uses a steiner tree to connect the dominating set. [26] constructs the backbone via algebraic connectivity and introduce a new metric, connectivity efficiency, as a benchmark when constructing the backbone. In [9], by setting a timer at each node, the nodes with higher node degree have higher probability to be included in the backbone, which finally produces a spanning tree. [27] aims to construct a backbone with the longest lifetime based on a weight matrix of energy efficiency.

In unit-disk graphs and general graphs, the size relationship between MCDS and MIS has been well studied, e.g., [12], [28]–[35]. [36] uses local neighbors information and takes node priority into consideration to construct a CDS, and used an iterative application of a selected local solution to maintain the CDS when the topology changed. In [37], a connected dominating set is built directly without calculating MIS. In [38], an MIS is constructed at first, and then the CDS constructed with gateway nodes. [39] proposed a protocol that is called Distributed Clustering Algorithm (DCA), which can produce a maximal independent that is also a minimal dominating set. [40] proposed a distributed algorithm for calculating a minimal dominating set by a sequential, centralized greedy way, whose execution time is polynomial, which is associated with the size of network. [34]presents a distributed algorithm which constructs a CDS D of size at most $\alpha \cdot opt$ for some fixed constant α in a polynomial time. Compared with [34], [35] proposes a polynomial-time constant-approximation algorithm, GOC-MCDS-C, that produces a CDS D whose size $|D|$ is within a constant factor from that of the minimum CDS. [41] proposed a local randomized greedy (LRG) algorithm, which calculates a minimal dominating set in poly-logarithmic time. However, it can‘t guarantee connectivity.

Another kind of approach focuses on cluster based topology and produces a independent set with the cluster heads. [42] selects the cluster head based on node degree, while in [43] the cluster head is selected based on the normalized link failure frequency and the mobility of the nodes. These algorithms usually start from a single-leader, whose election costs $O(n \log n)$ in message complexity [44]. To improve this, multiple-leader based algorithms are proposed in [12], [45], [46]. To connect all nodes in MIS, [45] requires that each node u in the MIS computes a shortest path to all independent neighbors (the nodes in I whose distance to u is either two or three hops) with a higher id. This connection algorithm results in a CDS with size at most $192 \cdot opt + 48$. By further exploring the geometric properties of neighboring independent nodes, [12] proposes a connection algorithm to generate a CDS with size at most $147 \cdot opt + 33$.

Note that [12], [45], [46] are the most related work since both propose to connect an MIS in a localized fashion. There exist other distributed or centralized algorithms to connect an MIS. For example, a distributed spanning tree can be constructed to connect all nodes in an MIS [47]; or a Steiner tree with minimum number of Steiner points can be applied to connect an MIS [48].

III. Preliminaries, Models and Assumptions

A. Preliminaries

- *dominating set*: Given a graph $G(V, E)$, a dominating set D of $G(V, E)$ is a subset of V such that for $\forall u \in V - D$, there exists a $v \in D$ satisfying $uv \in E$.
- *connected dominating set*: If all nodes in D induce a connected graph, D is a *connected dominating set*.
- *minimum (connected) dominating set*: Among all (connected) dominating sets of V, the one with the smallest cardinality is called the *minimum (connected) dominating set*.
- *independent set*: An *independent set* I of V is a subset of V such that $\forall u, v \in S$, $uv \notin E$.
- *maximal independent set (MIS)*: If adding any node $w \in V$ to I breaks the independent property, I is a *maximal independent set (MIS)*.

For any vertex u in a maximal independent set I, the length of the shortest path from u to its closest vertex in I is either two hops or three hops.

B. Network Model

In this paper, we model the ad hoc and sensor network as a unit-disk graph $G(V, E)$, a widely adopted model for wireless ad hoc and sensor networks in which nodes can communicate with each other if their distance is at most 1 unit. Specifically, V represents the set of sensors and E represents the set of edges. An edge $uv \in E$ if and only if $u, v \in V$ and the Euclidean distance between u and v is no larger than 1 unit. This assumption is reasonable as in ad hoc and sensor networks the topology is determined by the transmission range, which is usually fixed.

We assume that in the network there already exists an MIS and a corresponding backbone that are generated by any approach available. For example, the algorithms proposed in [12], [45], [46] can be applied here. Let u be any vertex in MIS, N_u be the node set of one-hop neighbors of u, I denote the node set of MIS, and C denote the set of nodes that are on the backbone but not in the MIS I (i.e., C is the set of nodes that connect the MIS nodes on the backbone). Let $N_u^{(I)}$ $\subset$ MIS denote the set of nodes in MIS that are two hops or three hops away from u, and I_u denote the set of MIS nodes within three hops of u. We assume $N_u^{(I)}$ and I_u are available to u.

When a node v fails or is added to the network, we assume there is a message broadcasted to v's neighbors in three-hop distance.

IV. Geometric Properties of Unit-Disk Graphs

Based on the definition, an edge in a unit-disk graph exists between two nodes if and only if their Euclidean distance is at most 1. We have identified the following properties:

Lemma 4.1: Let uv and st be two crossing edges in a unit-disk graph $G(V, E)$, as shown in Fig 1. Then at least one of u, v, s, t has direct edges to the other three vertices in G.

Proof: Due to the page limit, we omit the proof here. For more details, please refer to [49]. ■

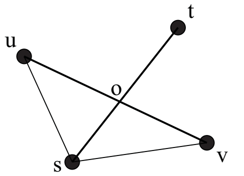


Fig. 1. uv and st are two crossing edges in a unit-disk graph G. Then at least one of u, v, s, and t can reach the other three vertices directly in G.

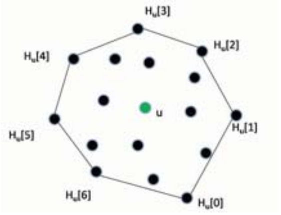


Fig. 2. The array H_u, in which elements are sorted in counter-clockwise.

Lemma 4.2: Let u, v, s, t be four vertices in any MIS of a unit-disk graph G such that there exists a path P_{uv} with length at most three hops to connect u and v and a path P_{st} with length at most three hops to connect s and t. Let P be the set of intersecting nodes in P_{uv} and P_{st}. Then u, v, s, t can reach each other by traversing only vertices in P.

Proof: Due to the page limit, we omit the proof here. For more details, please refer to [49]. ■

Note that the path length constraint of this Lemma can be relaxed. Actually in a unit-disk graph G, every pair of nodes in two crossing paths can reach each other by traversing only vertices in these two paths.

Lemma 4.3: Let u, v be two vertices in any MIS of a unit-disk graph G such that there exists a path P_{uv} with length at most three hops to connect u and v. Considering the straight line segment $\overline{uv}$, every point on $\overline{uv}$ is covered in the transmission range of the nodes in P_{uv}.

Proof: Due to the page limit, we omit the proof here. For more details, please refer to [49]. ■

Corollary 4.1: All the points within the convex polygon $uvst$ are covered by the transmission range of the nodes in P_{uv}.

Lemma 4.4: Let u, v, s, t be four vertices in any MIS of a unit-disk graph G such that u and v are within at most three hops, and s and t are within at most three hops. If the line segment $\overline{uv}$ crosses the line segment $\overline{st}$, given any arbitrary path P_{st} with length at most three hops to connect u and v and path P_{uv} with length at most three hops to connect s and t, they must be connected.

V. Localized Backbone Renovating Algorithm

In this section, a backbone expansion procedure is proposed first. Then we introduce our localized backbone renovating (LBR) algorithm. Specifically, during backbone renovating there are two scenarios to consider: 1. node failure; 2. node addition. In both scenarios we choose to update either I or C, or both. The detailed design of LBR algorithm is elaborated in the following sections.

A. Backbone Expansion with Convex-hull

Given a node u in MIS I, let H_u denote the convex hull of the nodes in $u \cup N_u^{(I)}$, where H_u is an array that records the nodes on the boundary of the convex hull of u, as shown in Fig.2. The convex hull can be easily calculated by Graham's Scan algorithm. Note that the number of nodes on the boundary of convex hull H_u is limited by a constant number 18 [32], the execution of Graham's Scan algorithm costs a constant time for the computation of convex hull H_u.

Let node $u \in V$ compute the shortest path to connect u and the nodes of $N_u^{(I)}$ on the boundary of convex-hull H_u. All the intermediate nodes that connect u and the nodes of $N_u^{(I)}$ on the convex-hull H_u form a set C_u, as shown in Fig.2. $\forall u \in V$ $\cap C^u$ form the set C. It is worth pointing out that $C \cup I$ is proved to be a backbone in [32]. In the following the word *backbone* refers to the expanded backbone.

B. Localized Backbone Renovation with Node Failure

Given an arbitrary node v fails in the network, there are three cases: (a). $v \in I$, namely v belongs to the MIS; (b). $v \in C$, namely v belongs to the backbone but v is not in the MIS; (c). v belongs to neither I nor C, namely v does not belong to the backbone. In the following we sketch the basic idea of our algorithm to deal with the three cases.

1) Case 1: $v \in I$, namely v belongs to the MIS: In this case, there are four steps to renovate the backbone:

- Step 1. The MIS is renovated by $I' = I \cup MDS(S)$, where S denotes the set of v's one-hop neighbors that are not adjacent to any node in I, and $MDS(S)$ denotes the minimum dominating set of S.

  Remark 5.1: Note that the local topology information (e.g., N_v^I) is available to $\forall u \in S$, where $|S|$ is no greater than the node degree of v (usually a small constant number), node $u \in S$ could easily compute a uniquely determined $MDS(S)$. Then the nodes $u \in N_v^I$ update their I_u. We have $I' = I \cup MDS(S)$.

  Remark 5.2: Note that though we use $I' = I \cup MDS(S)$, this update is not necessarily taken all over the network. Instead, it is only taken by nodes within three hops of v, i.e., only the nodes within v's three hops update their I_u with $I'_u = I_u \cup MDS(S) \cap N_u^I$.
- Step 2. The node $u \in I$ where $v \in H_u$ (i.e., v is on the boundary of the convex hull of u) renovates its convex hull based on $N_u^{I'} \setminus v$. Specifically, node u launches Graham Scan algorithm with $N_u^{I'} \setminus v$ to update its convex

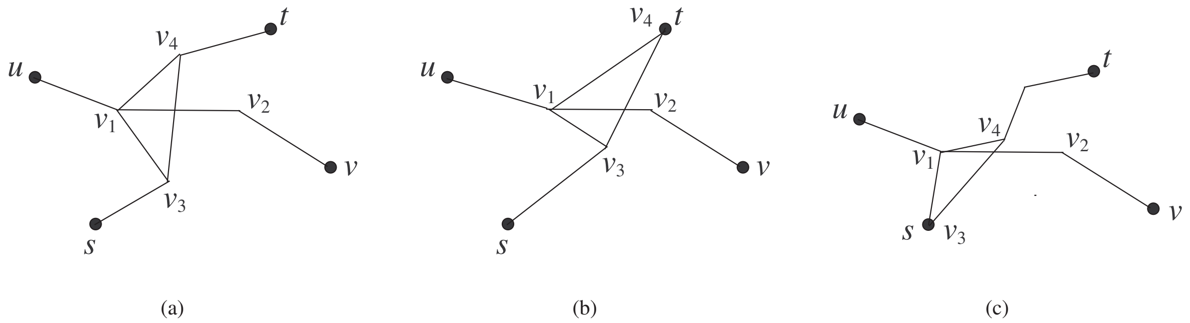
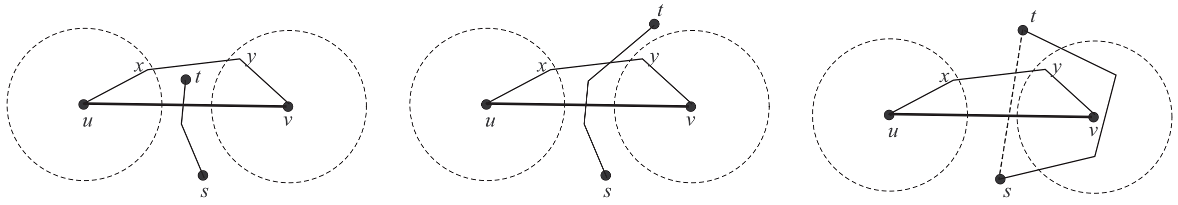


Fig. 3. Case study for Lemma 4.2. P_{uv} and P_{st} are two crossing paths in a unit-disk graph G. v_1, v_2, v_3 and v_4 are the four vertices of the two crossing edges. Then u, v, s, t connect to each other by traversing only nodes in P_{uv} and P_{st}.



(a) Case 1: t is within the convex polygon $uvyx$.

(b) Case 2: P_{st} crosses the line segment $\overline{uv}$.

(c) Case 3: P_{st} does not cross the line segment $\overline{uv}$.

Fig. 4. Case study for Lemma 4.4

hull and the shortest paths to the nodes on convex hull, denoted as H'_u and C'_u.

Remark 5.3: It is worth pointing out that all the nodes in H_u except v will remain in H'_u. The detailed proof is given in Lemma 6.1.

- Step 3. Every newly added node $u \in MDS(S)$ computes its convex hull H_u and the corresponding C_u to connect to the nodes in H_u. According to Step 1 and Step 2, the set C is renovated by $C' = (\bigcup_{u \in I' \setminus I} C_u) \cup C \setminus C_v$.
- Step 4. $I' \cup C'$ contributes the renovated backbone. It is worth pointing out that $I' \cup C'$ is a CDS. The detailed proof will be given in Lemma 6.4.

2) Case 2: $v \in C$, namely v is on the backbone but v does not belong to MIS: Let $u, w \in I$ denote two nodes in the MIS that are connected through node v on the backbone, where $u \in N^I_w$ and $w \in N^I_u$. If the alternate shortest path between u and w is not greater than 3, u and w will be connected with this alternate shortest path. The corresponding intermediate nodes on the path between u and w will be updated in C'.

If the alternate shortest path between u and w is greater than 3, i.e., $u \notin N^I_w$ and $w \notin N^I_u$, u and w begin to update their convex hulls based on Section V-A. All the nodes in H_u except w will remain in H'_u, and vice versa. The detailed proof is given in Lemma 6.1.

3) Case 3: $v \notin I \cup C$, namely v does not belong to the backbone: Since the backbone is a connected dominating set, for the nodes that do not belong to the backbone, they must be dominated by the backbone. In other words, they are all one-hop neighbors adjacent to the backbone nodes. Therefore, it doesn't need to take any action when node v fails in this case.

According to Lemma 6.9, $I' \cup C'$ is the renovated backbone and connects all the nodes in the network.

C. Localized Backbone Renovation with Node Addition

Given an arbitrary node v added into the network, there are two cases: (a). $v \in N_u$, where $u \in I$, namely v has a neighbor u in MIS; (b). $\nexists u \in I$ such that $v \in N_u$, namely v is not adjacent to any node in MIS. In the following we sketch the basic idea of our algorithm to deal with the two cases.

1) Case 1: $\nexists u \in I$ such that $v \in N_u$, namely v is not adjacent to any node in MIS:

- Step 1. The MIS I is renovated first by adding v to I. Let I' denote the renovated MIS, we have $I' = I \cup v$. Specifically, v selects itself as a new MIS node by broadcasting this notification to its neighbors within three hops and then collecting the local topology information from these nodes.

  Remark 5.4: Note that though we have $I' = I \cup v$, this information is not necessary to be broadcast over the whole network. Instead, every node only needs to know the topology changes within its three hops. Therefore, only the nodes u within v's three hops update their local MIS information I'_u.
- Step 2. The set C is renovated. Let C' denote the renovated set C, we have $C' = C \cup C_v$. Specifically, v computes H_v with N^I_v based on Graham Scan algorithm

and connects to the nodes on the boundary of convex-hull H_v via the shortest path. Then all the intermediate nodes that connect v and the nodes on the boundary of convex-hull H_v form the set C_v. $I' \cup C'$ contributes the renovated backbone.

- Step 3. $I' \cup C'$ contributes the renovated backbone.

2) Case 2: $v \in N_u$, where $u \in I$, namely v has a neighbor u in MIS: Since v has a neighbor u in MIS, it must be dominated by u on the backbone. Therefore, it doesn't need to take any action when node v is added into the network in this case.

According to Lemma 6.9, $I' \cup C'$ is the renovated backbone and connects all the nodes in the network.

D. LBR Algorithm

This section provides the pseudo code of LBR algorithm.

VI. Performance Analysis

Lemma 6.1: Given an arbitrary node v on the boundary of node u's convex hull H_u that fails, all the other boundary nodes in H_u will remain in the renovated H'_u.

Proof: It's obvious that the coverage area of convex-hull H_u will shrink. According to the property of convex hull and the execution procedure of Graham Scan algorithm, all the nodes in H_u except v will remain in H'_u. ■

Corollary 6.1: Given an arbitrary node $v \in I$ fails, during the convex hull renovation executed at the MIS node $u \in I$ that $v \in H_u$, all the nodes in H_u will remain in the renovated H'_u.

Lemma 6.2: Given an arbitrary node v in the network fails or is added in the network, LBR terminates locally in a constant time.

Proof: Due to the page limit, we omit the proof here. For more details, please refer to [49]. ■

Corollary 6.2: Given that an arbitrary node in the network fails or is added, the communication overhead of LBR is O(k), where k is the number of nodes broken or added.

Corollary 6.3: Given that an arbitrary node in the network fails or is added, the computation overhead of LBR is O(k), where k is the number of nodes broken or added.

Corollary 6.4: The computation complexity of LBR is $O(n)$, where n is the number of nodes broken or added.

Lemma 6.3: Given an arbitrary node $v \in I$ fails while the network is still connected, all the other nodes in I will remain on the renovated backbone.

Proof: According to Section V-B, during backbone renovating procedure, the only node that is removed from the backbone is v itself. All the other nodes in I remain in the newly renovated MIS I'. Since the backbone is $I' \cup C'$, all the other nodes in I remain on the renovated backbone. ■

Lemma 6.4: Given an arbitrary node $v \in I$ fails, while the network is still connected, the renovated backbone is connected.

Proof: Due to the page limit, we omit the proof here. For more details, please refer to [49]. ■

Algorithm 1 Localized Backbone Renovating Algorithm

Input: v, I, C, $G(V,E)$
Output: The renovated backbone $I' \cup C'$.

```
function LBR($v$, $I$, $C$, $G(V,E)$)
    Case 1: $v$ fails
    if $v \in I$ then    ▷ $v$ is an MIS node
        Step 1: $I' = I \cup MDS(S)$    ▷ The nodes within $v$'s three-hop distance update their $I_u$s with $I'_u = I_u \cup MDS(S) \cap N^I_u$, where $S$ denotes the set of $v$'s one-hop neighbors that are not adjacent to any node in $I$, and $MDS(S)$ denotes the minimum dominating set of $S$.
        Step 2: Renovate $H'_u \leftarrow H_u$ and $C'_u \leftarrow C_u$, $\forall u \in I$ where $v \in H_u$    ▷ Each MIS node $u \in I$ where $v \in H_u$ renovates its convex hull and corresponding $C_u$ based on $N^{I'}_u \setminus v$.
        Step 3: $C' = (\bigcup_{u \in I' \setminus I} C_u) \cup C \setminus v$    ▷ Every newly added node $u \in MDS(S)$ recomputes its convex hull $H_u$ and the corresponding $C_u$. The renovated $C'$ is updated by the newly generated $C_u$ according to Step 1 and Step 2.
    end if
    if $v \in C$ then    ▷ $v$ is on the backbone but is not an MIS node.
        Step 1. Compute the shortest path $SP(u,w)$ between $u$ and $w$    ▷ $u, w \in I$ denote two nodes in the MIS that are connected through node $v$ on the backbone, where $u \in N^I_w$ and $w \in N^I_u$
        Step 2-1. If $|SP(u,w)| \leq 3$, update $C'$ with $SP(u,w)$    ▷ If the alternate shortest path between $u$ and $w$ is no greater than 3, update the set $C'$ with this shortest path $SP(u,w)$.
        Step 2-2. If $|SP(u,w)| > 3$, recompute $H'_u, H'_w$ and $C'_u, C'_w$ and update $C$ with $C'_u$ and $C'_w$    ▷ If the alternate shortest path between $u$ and $w$ is greater than 3, i.e., $u \notin N^I_w$ and $w \notin N^I_u$, $u$ and $w$ begin to update their convex hulls and then $C_u$ and $C_w$.
    end if
    if $v \notin I \cup C$ then
        No action is needed.
    end if


    Case 2: $v$ is a newly added node
    if $\nexists u \in I$ such that $v \in N_u$ then    ▷ $v$ is not adjacent to any node in MIS
        Step 1: $I' = I \cup v$    ▷ The MIS $I$ is renovated by adding $v$ to $I$
        Step 2: Compute $H_v, C_v$    ▷ The set $C$ is renovated
    else
        No action is needed.
    end if
    Nodes in $I' \cup C'$ contribute the renovated backbone. Return.
end function
```

Lemma 6.5: Given an arbitrary node $v \in I$ is added into the network, while the network is still connected, the renovated backbone is connected.

Proof: According to Section V-C1, node v indirectly connects to the backbone via its convex hull H_v. Therefore the renovated backbone is still connected. ■

Corollary 6.5: Given an arbitrary node $v \in I$ fails or is added into the network, while the network is still connected, the renovated backbone is connected.

Lemma 6.6: Given an arbitrary node v in C fails, while the network is still connected, the renovated backbone is connected.

Proof: Due to the page limit, we omit the proof here. For more details, please refer to [49]. ■

Lemma 6.7: Given an arbitrary node $v \in N_u$ where $u \in I$ is added, while the network is still connected, the renovated backbone is connected.

Proof: According to Section V-C2, node v directly connects itself to the backbone (or more specifically node u). Therefore the renovated backbone is still connected. ■

Lemma 6.8: Given an arbitrary node on the backbone that fails, while the network is still connected, the renovated backbone is connected.

Proof: According to Corollary 6.5, Lemma 6.6 and Lemma 6.7, given an arbitrary node on the backbone fails, the renovated backbone provided by LBR is connected. ■

Lemma 6.9: Given an arbitrary node in the network fails or is added, while the network is still connected, the renovated backbone is connected.

Proof: According to Lemma 6.8, given an arbitrary node on the backbone fails or is added, the renovated backbone provided by LBR is connected. ■

Corollary 6.6: The renovated backbone is always connected if the network is connected.

Lemma 6.10: Let h denote the cardinality of the convex hull H_u, the cardinality of the renovated backbone is at most $2h \cdot |I|$.

Proof: Due to the page limit, we omit the proof here. For more details, please refer to [49]. ■

Note that a maximal independent set of V is also a dominating set of V. Multiple works (e.g., [21]) have proved the following result that relates the size of any MIS of a unit-disk graph G to that of its MCDS.

Lemma 6.11: Let I be any maximal independent set and opt be any MCDS of a unit-disk graph G. Then $|I| \leq k \cdot |opt| + 1$ for $|opt| > 1$, $k \leq 4$.

Lemma 6.12: Let h denote the cardinality of the convex hull H_u, which is usually a small constant. The size of the connected dominating set renovated by LBR is less than $8h \cdot opt + h + 1$, where opt is the size of a MCDS.

Proof: This lemma follows from Lemma 6.11 and Lemma 6.10. ■

VII. Simulation

In this section, We compare the performance of three different backbone maintenance algorithms with LBR, OST, a centralized algorithm that keeps a minimum spanning tree in entire network; AST, another centralized algorithm that keeps the minimum spanning trees computed with every MIS node as a root in the network; BF, a localized best-effort algorithm that tries to reconnect every broken part on the backbone with shortest path within three hops. It is worth pointing out that BF may fail to renovate the backbone sometimes and cannot guarantee network connectivity after maintenance.

The metrics we used to evaluate the performance of LBR and other algorithms are the size of maintained backbone and the success rate that the renovated backbone is connected while the network is connected.

A. Settings

In the simulation, nodes are randomly distributed in an area of $500m \times 500m$ and the results are averaged over 100 runs. The communication radius of each node is chosen from $[30m, 40m]$. According to our simulation settings, the radius of $30m$ indicates that the initial network is sparse (i.e., average node degree is about 6) and the radius of $40m$ indicates that the initial network is relatively dense (i.e., average node degree is about 10). Let cn denote the number of changed nodes in the network, where $cn = [50, 100, 150, 200, 250, 300, 350, 400]$ and 0 represents the initial topology, fr denotes the percentage of the number of failed nodes in cn, ar denotes the percentage of the number of added nodes in cn. Obviously, fr + ar = 100%.

The network topology changes over time in three ways. In the first topology changing situation, the initial number of nodes in the network is 300, $fr = 10\%$, and $ar = 90\%$. This setting can show the performance of these algorithms when the number of nodes in the network increases (i.e., the network becomes denser). In the second topology situation, the initial number of nodes is 500, $fr = 50\%$, and $ar = 50\%$. This setting can show the performance when the size of the network slightly changes. In the third situation, the initial number of nodes is 500, $fr = 90\%$, and $ar = 10\%$. This setting can show the performance when the number of nodes in the network decreases (i.e., the network becomes sparser). The failed nodes (newly added nodes) are randomly selected (deployed) in the network.

B. Simulation Results

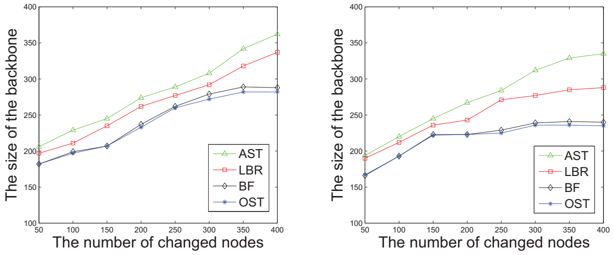
1) Simulation Study on Backbone Size: Fig.5, Fig.6, and Fig.7 illustrate the relationship between the size of the backbone and the number of changed nodes given $[ar = 90\%, fr = 10\%]$, $[ar = 50\%, fr = 50\%]$, and $[ar = 10\%, fr = 90\%]$, respectively, under different communication radii $30m$ and $40m$.

Both Fig.5 and Fig.7 show that as the number of nodes in the network increases (decreases), the size of the backbone increases (decreases) linearly in all the algorithms OST, AST, BF, and LBR. OST leads to the slowest backbone size increase (decrease) as it uses only one minimum spanning tree in the network. BF leads to the second slowest backbone size increase (decrease) as it repairs a spanning tree locally and thus leads to limited size increase (decrease) in the network. AST leads to the fastest backbone size increase (decrease) as it uses all possible minimum spanning trees rooted at the nodes in MIS. LBR leads to the medium increase (decrease) of backbone size among the three algorithms, as it repairs the backbone locally and terminates locally at a constant time, as shown in Lemma 6.2. LBR does not necessarily provide a minimum spanning tree or contribute the combination of all possible minimum spanning trees. Thus LBR leads to the medium increase (decrease) in both Fig.5 and Fig.7.

Fig.6 illustrates the relationship between the size of the backbone and the number of changed nodes given $ar = 50\%$ and $fr = 50\%$ under the communication radii $30m$ and $40m$, respectively. Both Fig.6(a) and Fig.6(b) show that as the number of nodes in the network remains stable, the size of the backbone remains stable in all algorithms OST, AST, BF, and LBR.

According to Fig.5, Fig.6, and Fig.7, OST and BF lead to the smallest backbone all the time as they use only one minimum spanning tree in the network. AST leads to the largest backbone as it uses all possible minimum spanning trees rooted at the nodes in MIS. LBR leads to a medium-size backbone among the three algorithms, as it repairs the backbone locally and terminates locally at a constant time, as shown in Lemma 6.2.

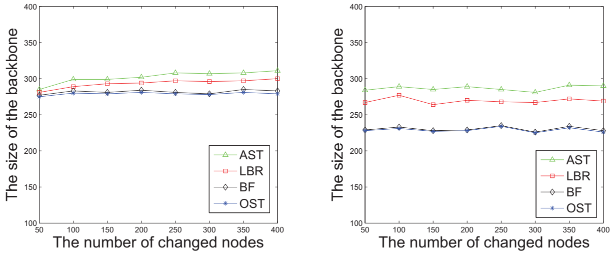
From Fig.5, Fig.6, and Fig.7, we can also find that the larger the communication radius, the denser the network, the smaller the renovated backbone, and vice versa. It is also interesting to observe that when the network becomes denser, the backbone size of LBR generally follows the trend of that of OST; when the network becomes sparser, the backbone size of LBR generally follows the trend of AST.



(a) Communication radius = 30m (b) Communication radius = 40m

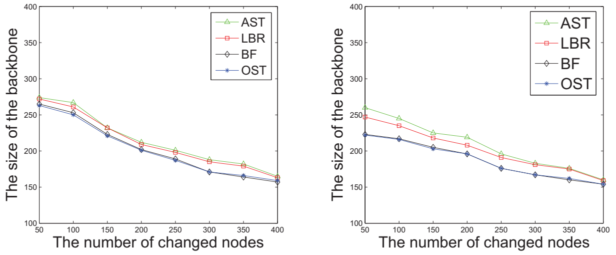
Fig. 5. The size of backbone $vs.$ the number of changed nodes under the topology changing setting $ar = 90\%$ and $fr = 10\%$ under different communication radii [30m, 40m].

2) Simulation Study on Success Rate: It is worth pointing out that the success rates of AST, OST are always 1 when the network is connected. This result is reasonable since both AST and OST are centralized algorithms and thus can always guarantee connectivity when the network is connected. Therefore, we simple compare the success rate of LBR and BF.



(a) Communication radius = 30m (b) Communication radius = 40m

Fig. 6. The size of backbone $vs.$ the number of changed nodes under the topology changing setting $ar = 50\%$ and $fr = 50\%$ under different communication radii [30m, 40m].



(a) Communication radius = 30m (b) Communication radius = 40m

Fig. 7. The size of backbone $vs.$ the number of changed nodes under the topology changing setting $ar = 10\%$ and $fr = 90\%$ under different communication radii [30m, 40m].

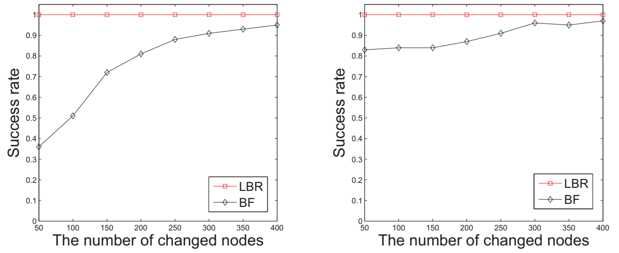
Fig.8, Fig.9, and Fig.10 illustrate the relationship between the success rate and the number of changed nodes given $[ar = 90\%, fr = 10\%]$, $[ar = 50\%, fr = 50\%]$, and $[ar = 10\%, fr = 90\%]$, respectively, under different communication radii $30m$ and $40m$.

Both Fig.8 and Fig.10 show that as the number of nodes in the network increases (decreases), the success rate that the backbone renovated by BF is connected increases (decreases) in BF. In Fig.9, it is also interesting to observe that when the number of nodes in the network remains stable, the success rate that the backbone renovated by BF is connected is not stable and slowly decreases as the number of changed nodes increases. This indicates that BF is not robust for backbone maintenance.

From Fig.8, Fig.9, and Fig.10, we can easily find that the backbone renovated by LBR is always connected if the network is connected. This is also proved in Lemma 6.9. In these figures, we can also find that the larger the communication radius, the denser the network, the higher the success rate that the backbone renovated by BF is connected, and vice versa.

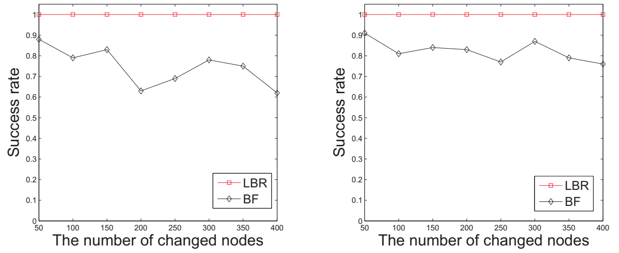
3) Communication and Computation Overhead: This section studies the relationship between the size of the network and the communication/computation overhead of the four algorithms. Specifically, we set the initial number of nodes in the network to $[500, 1000]$. The number of changed nodes is set to 500, given the topology changing setting $ar = 50\%$ and $fr = 50\%$ with a communication radius 40m.

Fig.11(a) shows that as the size of the network increases,



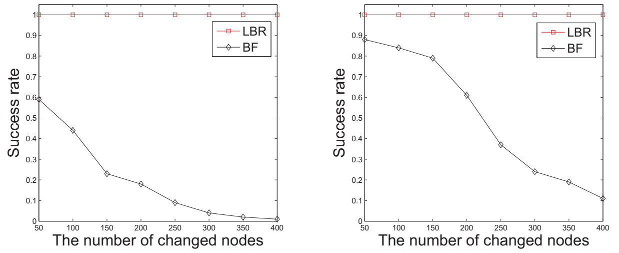
(a) Communication radius = 30m (b) Communication radius = 40m

Fig. 8. The success rate of maintaining the network connectivity via the renovated backbone $vs.$ the number of changed nodes under the topology changing setting $ar = 90\%$ and $fr = 10\%$ under different communication radii [30m, 40m].



(a) Communication radius = 30m (b) Communication radius = 40m

Fig. 9. The success rate of maintaining the network connectivity via the renovated backbone $vs.$ the number of changed nodes under the topology changing setting $ar = 50\%$ and $fr = 50\%$ under different communication radii [30m, 40m].

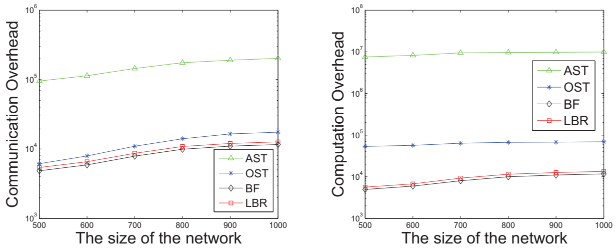


(a) Communication radius = 30m (b) Communication radius = 40m

Fig. 10. The success rate of maintaining the network connectivity via the renovated backbone $vs.$ the number of changed nodes under the topology changing setting $ar = 10\%$ and $fr = 90\%$ under different communication radii [30m, 40m].

the communication overhead of the four algorithms increases. As shown in the graph, since AST and OST are centralized algorithm, they need to collect the global topology information, and thus have the highest communication overhead. OST has smaller communication compared with AST, since OST computes much less number of minimum spanning trees than AST. LBR and BF are localized algorithms and thus have much less communication overhead compared with AST and OST. However, LBR has slightly larger communication overhead than that of BF, because it needs to repair the topology within three hops instead of repairing only one path.

Fig.11(b) shows that as the size of the network increases, the computation overhead of centralized algorithms increases to some extents, because the backbone size of OST and AST, which determines their computation overhead, becomes stable when the size of the network (or more specifically, the network density) increases to some extent. We can also find that the computation overhead of LBR and BF slowly increases, because as the size of the network increases the number of nodes needed to repair increases.



(a) Communication overhead (b) Computation overhead

Fig. 11. The communication/computation overhead $vs.$ the number of nodes in the network under the topology changing setting $ar = 50\%$ and $fr = 50\%$ with communication radius 40m.

VIII. Conclusion

In this paper we propose a localized backbone renovating algorithm (LBR) for backbone maintenance in wireless ad hoc and sensor networks. Our theoretical analysis shows that the LBR algorithm could renovate the backbone in a purely localized manner with guaranteed connectivity while keeping the backbone size within a constant factor from that of the minimum CDS. Unless the network is no longer connected, LBR can always keep the renovated backbone connected. Both theoretical analysis and simulation study also show that LBR has ultra low communication and computation overhead. Besides, LBR can deal with arbitrary number of node failures and additions, which provides good scalability to network management.

IX. Acknowledgments

This research is supported by NSFC under grant 61170267, 61003218, 61272444, Jiangsu NSF under grant BK2011358, RFDP under grant 20113402120008, and National 973 Program under grant 2011CB302905.

References

[1] I. Stojmenovic, *Handbook of sensor networks: algorithms and architectures*. Wiley-Interscience, 2005, vol. 49.

[2] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *DIALM '99: Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*, 1999, pp. 7–14.

[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, pp. 721–730, 2007.

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

[5] Y. Xu, J. Heidemann, and D. Estrin, “Geography-informed energy conservation for ad hoc routing,” in *MobiCom ’01: Proceedings of the 7th annual international conference on Mobile computing and networking*, 2001, pp. 70–84.

[6] R. Sivakumar, P. Sinha, and V. Bharghavan, “Cedar: a core-extraction distributed ad hoc routing algorithm,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1454 –1465, 1999.

[7] S. Basagni, M. Mastrogiovanni, A. Panconesi, and C. Petrioli, “Localized protocols for ad hoc clustering and backbone formation: A performance comparison,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 17, no. 4, pp. 292–306, 2006.

[8] J. Carle and D. Simplot-Ryl, “Energy-efficient area monitoring for sensor networks,” *Computer*, vol. 37, no. 2, pp. 40–46, 2004.

[9] K. Sakai, S. Huang, W. Ku, M. Sun, and X. Cheng, “Timer-based cds construction in wireless ad hoc networks,” *Mobile Computing, IEEE Transactions on*, vol. 10, no. 10, pp. 1388–1402, 2011.

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

[11] M. D. J. Blum and X. Cheng, *Handbook of Combinatorial Optimization*. Kluwer Academic Publisher, 2004, ch. Applications of Connected Dominating Sets in Wireless Networks, pp. 329–369.

[12] X. Cheng, M. Ding, D. H. Du, and X. Jia, “Virtual backbone construction in multihop ad hoc wireless networks,” *Wireless Communications and Mobile Computing*, vol. 6, pp. 183–190, 2006.

[13] I. Stojmenovic, “Comments and corrections to ”dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks”,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 15, no. 11, pp. 1054 –1055, 2004.

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

[15] I. Stojmenovic and J. Wu, “Broadcasting and activity scheduling in ad hoc networks,” *Mobile Ad Hoc Networking*, pp. 205–229, 2004.

[16] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[17] B. N. Clark, C. J. Colbourn, and D. S. Johnson, “Unit disk graphs,” *Discrete Mathematics*, vol. 86, pp. 165–177, 1990.

[18] J. Blum, M. Ding, A. Thaeler, and X. Cheng, “Connected dominating sets in sensor networks and manets,” in *Handbook of Combinatorial Optimization (Eds. D.-Z. Du and P. Pardalos)*, 2004, pp. 329–369.

[19] K. Alzoubi, P.-J. Wan, and O. Frieder, “New distributed algorithm for connected dominating set in wireless ad hoc networks,” in *HICSS ’02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS’02)-Volume 9*, 2002, p. 297.

[20] K. Alzoubi, P. Wan, and O. Frieder, “Distributed heuristics for connected dominating set in wireless ad hoc networks,” in *Journal of Communications and Networks*, vol. 4, no. 1, 2002.

[21] P.-J. Wan, K. M. Alzoubi, and O. Frieder, “Distributed construction of connected dominating set in wireless ad hoc networks,” in *Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, 2002, pp. 1597–1604.

[22] X. Cheng, “Routing issues in ad hoc wireless networks,” in *PhD Thesis, Department of Computer Science, University of Minnesota*, 2002.

[23] X. Cheng, X. Huang, D. Li, W. Wu, and D.-Z. Du, “A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks,” *Networks*, vol. 42, no. 4, pp. 202–208, 2003.

[24] P. Wan, K. Alzoubi, and O. Frieder, “Distributed construction of connected dominating set in wireless ad hoc networks,” in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, 2002, pp. 1597–1604.

[25] H. Guo, Y. Qian, K. Lu, and N. Moayeri, “Backbone construction for heterogeneous wireless ad hoc networks,” in *Communications, 2009. ICC’09. IEEE International Conference on*, 2009, pp. 1–5.

[26] Z. Zhang, Q. Ma, and X. Wang, “Exploiting use of a new performance metric for construction of robust and efficient wireless backbone network,” in *IWQoS’10*, 2010, pp. 1–9.

[27] S. Hussain, M. Shafique, and L. Yang, “Constructing a cds-based network backbone for energy efficiency in industrial wireless sensor network,” in *Proceedings of HPCC*, 2010, pp. 322–328.

[28] W. Wu, H. Du, X. Jia, Y. Li, and S. C.-H. Huang, “Minimum connected dominating sets and maximal independent sets in unit disk graphs,” *Theor. Comput. Sci.*, vol. 352, no. 1, pp. 1–7, 2006.

[29] M. Cardei, M. X. Cheng, X. Cheng, and D.-Z. Du, “Connected domination in ad hoc wireless networks,” in *International Conference on Computer Science and Informatics (CS&I 2002)*, 2002, pp. 251–255.

[30] Y. Li, M. T. Thai, F. Wang, C.-W. Yi, P.-J. Wan, and D.-Z. Du, “On greedy construction of connected dominating sets in wireless networks: Research articles,” *Wirel. Commun. Mob. Comput.*, vol. 5, no. 8, pp. 927–932, 2005.

[31] X. Cheng, X. Huang, D. Li, W. Wu, and D.-Z. Du, “A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks,” *Networks*, vol. 42, no. 4, pp. 202–208, 2003.

[32] D. Chen, X. Mao, X. Fei, K. Xing, F. Liu, and M. Song, “A Convex-Hull based algorithm to connect the maximal independent set in Unit-Disk graphs,” 2006, pp. 363–370.

[33] L. Ding, W. Wu, J. Willson, H. Du, and W. Lee, “Construction of directional virtual backbones with minimum routing cost in wireless networks,” in *IEEE INFOCOM’11*. IEEE, 2011, pp. 1557–1565.

[34] H. Du, Q. Ye, W. Wu, W. Lee, D. Li, D. Du, and S. Howard, “Constant approximation for virtual backbone construction with guaranteed routing cost in wireless sensor networks,” in *IEEE INFOCOM’11*, 2011, pp. 1737–1744.

[35] H. Du, W. Wu, Q. Ye, D. Li, W. Lee, and X. Xu, “Cds-based virtual backbone construction with guaranteed routing cost in wireless sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, 2012.

[36] J. Wu, F. Dai, and S. Yang, “Iterative local solutions for connected dominating set in ad hoc wireless networks,” *Computers, IEEE Transactions on*, vol. 57, no. 5, pp. 702–715, 2008.

[37] B. Das and V. Bharghavan, “Routing in ad-hoc networks using minimum connected dominating sets,” in *IEEE ICC’97*, vol. 1, 1997, pp. 376–380.

[38] S. Basagni, D. Turgut, and S. Das, “Mobility-adaptive protocols for managing large ad hoc networks,” in *IEEE ICC’01*, vol. 5, 2001, pp. 1539–1543.

[39] S. Basagni, “Distributed clustering for ad hoc networks,” in *I-SPAN’99*, 1999, pp. 310–315.

[40] B. Liang and Z. Haas, “Virtual backbone generation and maintenance in ad hoc network mobility management,” in *IEEE INFOCOM’00*, vol. 3, 2000, pp. 1293–1302.

[41] L. Jia, R. Rajaraman, and T. Suel, “An efficient distributed algorithm for constructing small dominating sets,” *Distributed Computing*, vol. 15, no. 4, pp. 193–205, 2002.

[42] M. Gerla and J. Tsai, “Multicluster, mobile, multimedia radio network,” *Wireless networks*, vol. 1, no. 3, pp. 255–265, 1995.

[43] U. Kozat, G. Kondylis, B. Ryu, and M. Marina, “Virtual dynamic backbone for mobile ad hoc networks,” in *ICC’01*, vol. 1, 2001, pp. 250–255.

[44] I. Cidon and O. Mokryn, “Propagation and leader election in a multihop broadcast environment,” in *DISC ’98: Proceedings of the 12th International Symposium on Distributed Computing*, 1998, pp. 104–118.

[45] K. M. Alzoubi, P.-J. Wan, and O. Frieder, “Message-optimal connected dominating sets in mobile ad hoc networks,” in *MobiHoc ’02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, 2002, pp. 157–164.

[46] Y. Li, S. Zhu, M. T. Thai, and D.-Z. Du, “Localized construction of connected dominating set in wireless networks,” in *NSF International Workshop on Theoretical Aspects of Wireless Ad Hoc, Sensor and Peer-to-Peer Networks (TAWN04)*, 2004.

[47] S. Guha and S. Khuller, “Approximation algorithms for connected dominating sets,” *Algorithmica*, vol. 20, no. 4, pp. 374–387, 1998.

[48] M. Min, H. Du, X. Jia, C. X. Huang, S. C.-H. Huang, and W. Wu, “Improving construction for connected dominating set with steiner tree in wireless sensor networks,” *J. of Global Optimization*, vol. 35, no. 1, pp. 111–119, 2006.

[49] K. Xing and S. Zhang, “A localized backbone renovating algorithm for wireless ad hoc and sensor networks,” in *Tech Report*, 2012, http://staff.ustc.edu.cn/ kxing/Publications/TechReport/shuo.pdf.