Efficient Weighted Backbone Construction for Wireless Ad Hoc Networks

Xiang-Yang Li, Kousha Moaveni-Nejad, Wen-Zhan Song, Wei-Zhao Wang, Yu Wang
Department of Computer Science, Illinois Institute of Technology
10 West 31st Street, Chicago, IL 60616, USA

Abstract—**Backbone based routing has been extensively studied for wireless ad hoc networks recently. However, no priori method exists that can construct a backbone efficiently when each wireless node has a cost of being in the backbone. In this paper we propose a new efficient method to construct the weighted backbone for efficient routing in wireless ad hoc networks. We prove that the total cost of the constructed backbone is within $\min(4\delta + 1, 18\log d) + 10$ times of the optimum. The total number of messages of our method is $O(n)$ when the geometry information of each wireless node is known; the total number of messages is $O(m)$ when no geometry information is known. We also show that the constructed backbone is efficient for unicast: the total cost (or hop) of the least cost (or hop) path connecting any two nodes using backbone is no more than 3 times of the least cost (or hop) path in the original communication graph.**

Keywords— **Connected dominating set, clustering, localized algorithm, wireless ad hoc networks.**

I. Introduction

Wireless ad hoc networks draw lots of attentions in recent years due to its potential applications in various areas. We consider a wireless ad hoc network consisting of a set V of n wireless nodes distributed in a two-dimensional plane. Each wireless node has an omni-directional antenna. This is attractive for a single transmission of a node can be received by all nodes within its vicinity. By a proper scaling, we assume that all nodes have the maximum transmission range equal to one unit. These wireless nodes define a *unit disk graph* $UDG(V)$ in which there is an link between two nodes if and only if their Euclidean distance is at most one. The number of links in the unit disk graph could be as large as $O(n^2)$, i.e., the square order of the number of network nodes.

The movement of wireless nodes causes the network topology to change constantly, which makes efficient routing in non-static wireless ad hoc networks difficult and challenging. We will assume that the nodes are static or can be viewed as static during a reasonable period of time. The simplest routing method is to flood the message, which not only wastes the rare resources of wireless node, but also diminishes the throughput of the network. One way to avoid flooding is to let each node communicate with only a selected subset of its neighbors [1], [2], [3], [4], or to use a hierarchical structure like Internet. Examples of hierarchical routing are dominating set based routings [5], [6], [7], [8].

Many researchers proposed to use the *connected dominating set* (CDS) as a virtual backbone for hierarchical routing in wireless as hoc networks [6], [9], [7], [10]. Efficient distributed algorithms for constructing connected dominating sets in ad hoc wireless networks were well studied [11], [12], [13], [14], [15], [6], [7], [16]. The notion of cluster organization has been used for wireless ad hoc networks since their early appearance. Baker *et al.* [12], [13] introduced a "fully distributed linked cluster architecture" mainly for hierarchical routing and demonstrated its adaptivity to the network connectivity changes. The notion of the cluster has been revisited by Gerla *et al.* [17], [18] for multimedia communications with the emphasis on the allocation of resources, namely, bandwidth and channel, to support the multimedia traffic in an ad hoc environment. In [19], Gao, *et al.* proposed a randomized algorithm for maintaining the discrete mobile centers, i.e., dominating sets. They showed that it is an $O(1)$ approximation to the optimal solution with very high probability, but the constant approximation ratio is quite large. Recently, Alzoubi, *et al.* [11] proposed a method to approximate *minimum connected dominating set* (MCDS) within 8 whose time complexity is $O(n)$ and message complexity is $O(n \log n)$. Alzoubi [20] continued to propose a localized method that can construct the MCDS using linear number of messages. Existing clustering methods first choose some nodes to act as coordinators of the clustering process, i.e., clusterhead. Then a cluster is formed by associating the clusterhead with some (or all) of its neighbors. Previous methods differ on the criterion for the selection of the clusterhead, which is either based on the lowest (or highest) ID among all unassigned nodes [13], [18], or based on the maximum node degree [17], or based on some generic weight [14] (node with the largest weight will be chosen as clusterhead). Notice that, any maximal independent set is always a dominating set. Several clustering methods essentially compute a maximal independent set as the final clusterheads. In [21], [22], Wang and Li used any method that can approximate the MCDS efficiently and then build the local Delaunay graph [23] on top of the approximated MCDS. They showed that the resulting graph is a hybrid sparse spanner for network topology.

All of above methods try to minimize the number of clusterheads, in other words, the number of nodes in the backbone. However, in wireless ad hoc networks, each nodes may have different costs for being in the constructed backbone, due to various devices, power capacities, and loads. Therefore, in this paper, we assume that each wireless node has a *cost (or weight)* of being in the backbone. We study how to construct a sparse backbone efficiently for a set of weighted wireless nodes such that the total cost of the backbone is minimized and every route in the constructed network topology is efficient. Here a route is *efficient* if its cost or hops is no more than a constant factor of the minimum length or hops needed to connect the source and the destination in the unit disk graph. To the best of our knowledge, this is the first method to generate weighted backbone with a constant approximation ratio while the communication cost of *each* wireless node is bounded by a constant. We prove that the total cost of the constructed backbone is within

$\min(4\delta+1, 18\log d)+10$ times of the optimum. The total number of messages of our method is $O(n)$ when the geometry information of each wireless node is known; the total number of messages is $O(m)$ when no geometry information is known. We also show that the constructed backbone is efficient for unicast: the total cost (or hop) of the least cost (or hop) path connecting any two nodes using backbone is no more than 3 times of the least cost (or hop) path in the original communication graph.

The rest of the paper is organized as follows. In Section II, we provide preliminaries necessary for describing our new algorithms, and briefly review the literature related to backbone construction methods. Section III presents our new weighted backbone formation algorithms. We conclude our paper in Section IV by pointing out some possible future research directions.

II. Preliminaries

In this section, we give some definitions and notations that will be used in our presentation later. We assume that all wireless nodes are given as a set V of n points in a two dimensional space. Each node has some computational power. These nodes induce a *unit disk graph* $UDG(V)$ in which there is an edge between two nodes if and only if their distance is at most one. Hereafter, we always assume that $UDG(V)$ is a connected graph. We call all nodes within a constant k hops of a node u in the unit disk graph $UDG(V)$ as the k*-local nodes* or k*-hop neighbors* of u, denoted by $N_k(u)$, which includes u itself. We always assume that the nodes are almost-static in a reasonable period of time.

A subset of vertices in a graph G is an *independent set* if for any pair of vertices, there is no edge between them. It is a *maximal independent set* if no more vertices can be added to it to generate a larger independent set. It is a *maximum independent set* (MIS) if no other independent set has more vertices.

A subset S of V is a *dominating set* if each node u in V is either in S or is adjacent to some node v in S. Nodes from S are called dominators, while nodes not is S are called dominatees. Clearly, any maximal independent set is a dominating set. A subset C of V is a *connected dominating set* (CDS) if C is a dominating set and C induces a connected subgraph. Consequently, the nodes in C can communicate with each other without using nodes in $V-C$. A dominating set with minimum cardinality is called minimum dominating set, denoted by MDS. A connected dominating set with minimum cardinality is the *minimum connected dominating set*, denoted by MCDS.

In wireless ad hoc networks, every wireless node u may have a cost (or weight) $c(u)$ to being in backbone. Then a connected dominating set C is called *weighted connected dominating set* (WCDS). A subset C of V is a *minimum weighted connected dominating set* (MWCDS) if C is a WCDS with minimum total cost.

III. Efficient Weighted Backbone Formation

In this section, we study how to form a backbone (weighted connected dominating set) for a homogeneous network modelled by UDG by assuming that each wireless node u has a cost $c(u)$ of being on the backbone. We will propose a localized algorithm that can construct a backbone whose total cost is no more than $\min(18\log d, 4\delta+1)+10$ times of the optimum solution. Here d is the maximum degree of all wireless nodes, and $\delta=\max_{i,j\in E} c(i)/c(j)$, where E is the set of communication links in the wireless networks modelled by UDG=(N,E).

Here, we assume that each node knows the IDs (and costs) of all its 1-hop neighbors, which can be achieved by requiring each node to broadcast its ID (and cost) to its 1-hop neighbors initially. This protocol can be easily implemented using synchronous communications as did in [12], [13]. If the number of neighbors of each node is known a priori, then this protocol can also be implemented using asynchronous communications. Here, knowing the number of neighbors ensures that a node does get all updated information of its neighbors so it knows that whether itself has the smallest ID (or cost) among all its neighbors.

Our method has the following two phases: the first phase (clustering phase) is to find a set of wireless nodes as the dominators[1] and the second phase is to find a set of nodes to connect these dominators to form the final backbone of the network. Figure 1 gives an example of the constructed backbone and different roles of nodes.

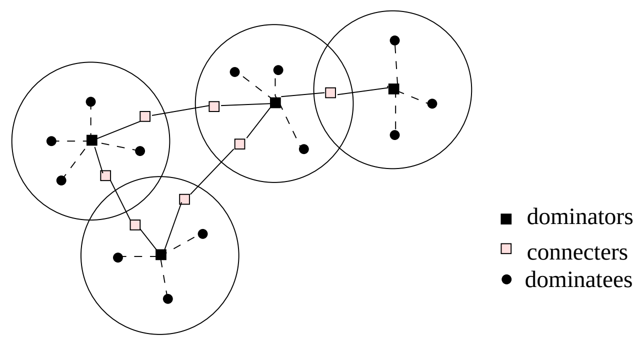

Fig. 1. An example of backbone.

Before describing our method, we first review an important observation of the *dominating set* on UDG, which will play master roles in our proofs later. After clustering, one dominator node can be connected to many dominatees. However, it is well-known that a dominatee node can only be connected to at most *five* independent nodes in the unit disk graph model. Generally, it is well-known that, for each node, there are at most a constant number of independent nodes that are at most k units away. The following lemma which bounds the number of independent nodes within k units from a node v is proved in [21], [22] by using a simple area argument.

Lemma 1: For every node v, the number of independent nodes inside the disk centered at v with radius k-units is bounded by a constant $\ell_k=(2k+1)^2$.

The bounds on ℓ_k can be improved by a tighter analysis. In [24], Li *et al.* gave the detailed proof to show that for unit disk graph the number of independent nodes in 2-hops neighborhood (not including the 1-hop neighbors) is at most 13 while the number of independent nodes in 1-hop neighborhood is at most 5.

[1] We will interchange the terms cluster-head and dominator. The node that is not a cluster-head is also called *ordinary* node or *dominatee*. A node is called *white* node if its status is yet to be decided by the clustering algorithm. Initially, all nodes are white. The status of a node, after the clustering method finishes, could be *dominator* or *dominatee*.

Therefore, there are at most 18 independent nodes inside the disk centered at a node v with radius 2.

A. Finding Dominators

Many methods have been proposed in the literature to find a dominating set for homogeneous networks. We will first show by examples that these methods do not generate a dominating set whose cost is always comparable with ours in the worst case. Since, in this paper, we are interested in localized methods, we will thus mainly discuss the priori localized methods here.

The first method to generate a dominating set would be to generate a maximal independent set as follows [21], [22]. First, assume that all nodes are marked as WHITE originally, which represents that the node is not assigned any role yet. A node u sends a message IamDominator to all its one-hop neighbors if it has the smallest cost (ID is often used if every node has unit cost) among all its WHITE neighbors. Node u also marks itself Dominator. When a node v received a message IamDominator from its one-hop neighbors, node v then marks itself Dominatee. Node v then sends a message IamDominatee to all its one-hop neighbors. Clearly, the nodes marked with Dominator indeed form a dominating set.

We then show by example that the produced dominating set may be arbitrarily larger than the optimum solution. See Figure 2 for an illustration. Assume that $2n$ wireless nodes u_i and v_i ($0 \leq i < n$) are distributed on two circles centered at one node w with radii 1 and 2. The cost of every nodes are assigned as the blue values in the Figure 2. The dominators selected by the first method are nodes w and u_i ($0 \leq i < n$), the total cost of the solution is ∞. However, the optimal solution formed by v_i ($0 \leq i < n$) has total cost n.

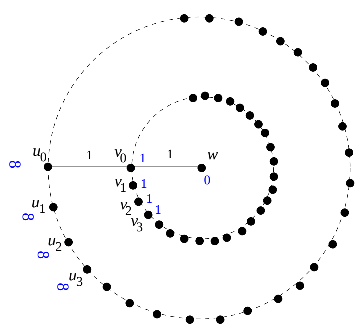

Fig. 2. An example why the first method fails to produce low cost weighted connected dominating set.

The second method of constructing a dominating set is based on minimum weighted set cover. The method can be described in a centralized was as follows: in each round, we select a node i with the minimum ratio $c(i)/d_i$, where d_i is the number of nodes not covered by previously selected dominators. It is well-known that this centralized method produces a dominating set whose total cost is no more than $\log d$ times of the optimum, where d is the maximum original degree of all nodes. In [11], Alzoubi *et al.* gave an example (as in Figure 4) with a family of instances for which the size of the solution computed by the second method is larger than the optimum solution by a logarithm factor. In this example, all nodes have a unit weight. For the detail of this example, see [11]. Moreover, this method is expensive to implement in a distributed way. First, among all nodes, it is expensive to find the node i with the minimum ratio $c(i)/d_i$ among all unchosen nodes. Second, it is also expensive to update the number of neighbors that are not covered by previously selected dominators.

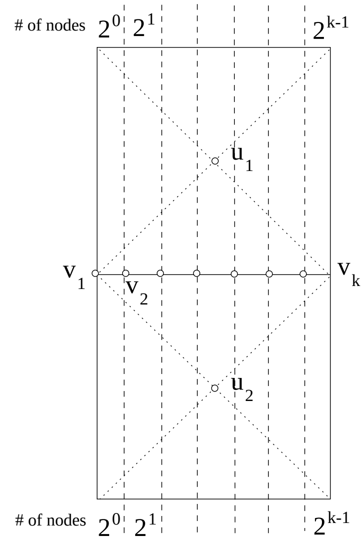

Fig. 3. An example why the second method fails to produce low cost weighted connected dominating set.

The third method to select the dominating set is proposed by Bao and Garcia-Luna-Aceves [25]. A node decides to become a dominator if either one of the following criteria are satisfied: 1) the node has the smallest cost in its one-hop neighborhood; 2) the node has the smallest cost in the one-hop neighborhood of one of its one-hop neighbors. This method can be performed in localized way or rounds based way. However, we can show by an example that the produced dominating set may be arbitrarily larger than the optimum solution. See Figure 4 for an illustration. Again assume that $2n + 1$ wireless nodes are distributed as shown in Figure 4. The cost of every nodes are re-assigned as the blue values in the Figure 4. The dominators selected by the third method are nodes w and v_i ($0 \leq i < n$), the total cost of the solution is $n(1 - \epsilon)$. However, the optimal solution formed by node w and seven nodes from u_i has total cost 7. It is easy to show that seven unit disks centered at 7 nodes among all u_i can cover all u_i.

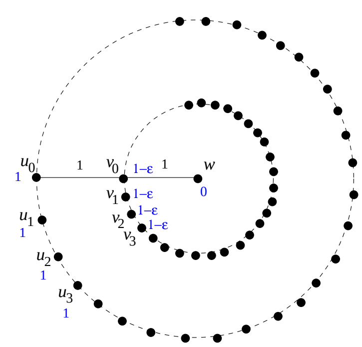

Fig. 4. An example why the third method fails to produce low cost weighted connected dominating set.

We then propose our method of constructing a connected dominating set whose total cost is comparable with the optimum solution. Our method works as follows.

Algorithm 1: Constructing efficient dominating set

1. First assume that all nodes are originally marked WHITE.
2. A node u sends a message ItryDominator to all its one-hop neighbors if it has the smallest cost among all its WHITE neighbors. Node u also marks itself PossibleDominator.
3. When a node v received a message ItryDominator from its one-hop neighbors, node v then marks itself Dominatee. Node v then sends a message IamDominatee to all its one-hop neighbors.
4. When a node w receives a message IamDominatee from its neighbor v, node w removes node v from its list of WHITE neighbors.
5. Each node u marked with PossibleDominator collects the cost and ID of all of its two hop neighbors.
6. Using the greedy method for minimum weighted set cover (like the second method), node u selects a subset of its two hop-neighbors to cover *all* the one-hop neighbors (including u) of node u. If the cost of the selected subset is smaller than the cost of node u, then node u sends a message YouAreDominator to each node in the selected subset. Otherwise, node u just marks itself Dominator.
7. When a node w received a message YouAreDominator, node w marks itself Dominator.

Theorem 2: Algorithm 1 constructs a dominating set whose total cost is no more than $\min(18 \log d, 4\delta + 1)$ times of the optimum.

PROOF. First, we prove the total cost of the maximum independent set MIS formed by all PossibleDominator nodes is no more than $4\delta + 1$ times of the optimum. Assume node u is a node from the optimum OPT. If u is not a PossibleDominator node then there are at most 5 PossibleDominator nodes around u. Let $v_1^u, v_2^u, \cdots, v_5^u$ denote them. The cost of one of these five nodes is smaller than the cost of u, otherwise node u will be selected as a PossibleDominator node. W.l.o.g., let $c(v_1^u) \le c(u)$. We also know that $c(v_i^u) \le \delta c(u)$ for $2 \le i \le 5$. Thus, $\sum_{1 \le i \le 5} c(v_i^u) \le (4\delta + 1)c(u)$. If we summarize the inequations for all nodes in the optimum dominating set OPT, we get

$$\sum_{u \in OPT} \sum_{1 \le i \le 5} c(v_i^u) \le (4\delta{+}1) \sum_{u \in OPT} c(u) = (4\delta{+}1)c(OPT).$$

Since $c(MIS) = \sum_{v \in MIS} c(v) \le \sum_{u \in OPT} \sum_{1 \le i \le 5} c(v_i^u)$, it is obvious that

$$c(MIS) \le (4\delta + 1)c(OPT).$$

Then, we prove the total cost of the nodes selected by the greedy method in Step 6 is no more than $18 \log d$ times of the optimum. Assume that node u runs the greedy algorithm and gets the subset as $GRDY_u$, and the cost of the selected subset $c(GRDY_u)$ is at most $c(u)$. It is well known that the dominating set generated by the greedy algorithm for set cover is no more than $\log f$ times of the optimum if every set has at most f items. Here, we know that every dominator can cover at most d dominatees, thus, $c(GRDY_u) \le \log d \cdot c(LOPT_u)$. Here $LOPT_u$ is the local optimum dominating set in the 2-hops neighborhood of u. Assume that OPT_u is the subset of the optimum solution for $MWCDS$ which falls in the 2-hops neighborhood of u. Clearly we have $c(LOPT_u) \le c(OPT_u)$, since $LOPT_u$ is the local optimum. Thus, $c(GRDY_u) \le \log d \cdot c(LOPT_u) \le \log d \cdot c(OPT_u)$. Consider all nodes in the MIS, we get

$$\begin{aligned} c(GRDY) &\le \sum_{u \in MIS} c(GRDY_u) \\ &\le \log d \cdot \sum_{u \in MIS} c(OPT_u). \end{aligned}$$

Remember that for each node v, the number of independent nodes in the 2-hops neighborhood of v is bounded by 18. Therefore, each dominator is counted by at most 18 times. Thus, $\sum_{u \in MIS} c(OPT_u) \le 18c(OPT)$, since the union $\bigcup_{u \in MIS} OPT_u$ is also a dominating set.

Remember that, for each node u in MIS, we either use u as the dominator or as $GRDY_u$ as the dominators, whichever has a smaller cost. Then, the total weight of the final dominating set is at most

$$\begin{aligned} &\sum_{u \in MIS} \min(c(u), c(GRDY_u)) \\ &\le \min(\sum_{u \in MIS} c(u), \sum_{u \in MIS} c(GRDY_u)) \\ &\le \min(4\delta + 1, 18 \log d) \cdot c(OPT). \end{aligned}$$

This finishes our proof. □

Notice that here the approximation ratio is $\min(18 \log d, 4\delta + 1)$. So if one of $\log d$ and δ is small, the approximation ratio is small.

Theorem 3: Algorithm 1 uses $O(n)$ messages.

PROOF. First, for messages ItryDominator and IamDominatee, every node at most send out once this kind of messages. Thus, the total number of these two messages is $O(n)$.

Second, for each PossibleDominator node, it need to collects the cost and ID of all of its two hop neighbors. This step may cost lots of communications (at most $O(m)$ messages, where m in the number of links in the original UDG). However, recently Calinescu [26] proposed a communication efficient method (using $O(n)$ messages) to collect $N_2(u)$ for every node u when the geometry information is known.

Third, after applying the greedy method node u may send a message YouAreDominator to node v, but since the number of independent nodes u in two hops of v is bounded by a constant, the total number of this message is also $O(n)$.

Consequently, Algorithm 1 uses $O(n)$ messages. □

B. Finding Connectors

The second step of weighted connected dominating set formation is to find some *connectors* (also called *gateways*) among all the dominatees to connect the dominators. Then the connectors and the dominators form a *connected dominating set* (or called backbone) as in Figure 1.

The differences of proposed clustering methods approximating MCDS lie in how to find gateways to connect these clusterheads, and whether providing performance guarantees. For example, the basic algorithm for constructing a CDS proposed in [17] does not guarantee that the constructed clusters are connected. As it agreed, in some cases, it needs *Distributed Gateway* (DG) to connect some clusters that are nonoverlapping. But, how to choose the DGs was not specified. Additionally, no performance guarantee was proved. In [12], [13], they consider in detail how to select the gateway nodes to connect the clusters based on cases of overlapping clusters and nonoverlapping clusters. Here, two clusters (headed by two different clusterheads) are said to be overlapping if there is at least one common dominatee node; they are said to be nonoverlapping if there is two dominatee nodes (one from each cluster) are connected. However, they did not prove the message complexity of their protocols, nor the approximation ratio of the generated connected dominating set. Additionally, as they agreed, it may generate two or perhaps more gateway pairs for some nonoverlapping clusters pair. On the other hand, Alzoubi *et al.* [5], [20], [21], [22] proposed several localized methods to find connectors using total $O(n)$ messages and showed that the constructed CDS is within a constant factor of optimum. However, all of these methods only consider the unweighted scenario. We can show by example that these methods generally do not produce a weighted connected dominating set with good approximation ratio.

Given a dominating set S, let $VirtG$ be the graph connecting all pairs of dominators u and v if there is a path in UDG connecting them with at most 3 hops. It is well-known that the graph $VirtG$ is connected. This observation is a basis of several algorithms [12], [13], [17], [18] for constructing a CDS. It is natural to form a connected dominating set by finding connectors to connect any pair of dominators u and v if they are connected in $VirtG$. This strategy was used in several previous methods, such as [11], [20], [12], [13], [18], [21], [22].

Our new connector selection method for weighted connected dominating set is also based one this observation. First, we define two dominators u and v are *neighboring dominators* if they are at most 3 hops away. Let $\mathsf{LCP}(u,v,G)$ denote the least cost path $uv_1v_2\cdots v_kv$ between vertices u and v on a edge weighted graph G, and $\mathcal{L}(u,v,G)$ denote the cost of nodes on path $\mathsf{LCP}(u,v,G)$ excluding u and v, i.e., $\mathcal{L}(u,v,G) = \sum_{1\le i\le k} c(v_k)$.

Algorithm 2: **Connector Selection**

1. Every dominatee w broadcasts to its 2-hop neighbors the information about its dominators.
2. For any dominator u, it collects all 3 hops paths information for all its neighboring dominators.
3. For each neighboring dominator v of u, find the least cost path $\mathsf{P}(uv)$ among all paths with at most 3 hops. Notice that if u and v can communicate directly (1 hop away), the least cost path $\mathsf{P}(uv)$ is just the link uv and has a cost 0. [2] Mark all nodes on the path as GRAY.
4. Build an *edge weighted* virtual graph $VirtG$ using all dominators as its vertices. Add virtual edge $\widetilde{uv}$ to virtual graph $VirtG$, if dominator u and v are within three hops. The length of $\widetilde{uv}$ is the cost of path $\mathsf{P}(uv)$. Notice that here the cost of end-nodes u and v is not included.
5. Build MST on graph $VirtG$, let $VMST$ denote $MST(VirtG)$.
6. For any virtual edge $e \in VMST$, select each of the dominatees on the path corresponding to e as a connector.

The graph constructed by combining all of dominators and the connectors selected by the above algorithm is called a WCDS graph (or *backbone* of the network). If we also add all edges that connect all dominatees to their dominators, the graph is called extended WCDS, denoted by WCDS'. In Figure 1, we present an example of WCDS', where the solid lines in the graph forms the WCDS graph, the square nodes are dominators or connectors, while the circular nodes are dominatees. The set of dominators and connectors forms a *connected dominating set*.

The following lemma about the relationship between $\mathcal{L}(u,v,UDG)$ and $\mathcal{L}(u,v,VirtG)$ will be used later to prove that our backbone structure is indeed low-cost.

Lemma 4: For any pair of dominator nodes u and v,

$$\mathcal{L}(u,v,VirtG) \le 2\cdot\mathcal{L}(u,v,G).$$

PROOF. Notice the original graph is node weighted while the virtual graph $VirtG$ is edge weighted. We assume that path $uv_1v_2\cdots v_kv$ is the least cost path connecting u and v in the original graph UDG, as shown in Figure 5.

For any dominatee node p in original communication graph, it must be dominated by at least one dominator. Thus, we can assume node u_i are node v_i's dominator as shown in Figure 5. For dominators u_i and u_{i+1}, we argue that the length of $\widetilde{uv}$ is at most the summation of the cost of v_i and v_{i+1}. We discuss by cases:

Case 1: $u_i = u_{i+1}$. In this case, we can consider there is a self loop from u_i with length 0. Thus, the cost $c(v_i)+c(v_{i+1})$ is at most 0, which is greater than the self loop.

Case 2: $u_i = v_i$ and $u_{i+1} = v_{i+1}$. In this case, there is a edge with 0 cost between u_i and u_{i+1}, which is guaranteed smaller than $c(v_i)+c(v_{i+1})$.

Case 3: $u_i = v_i$ and $u_{i+1} \ne v_{i+1}$ (or $u_i \ne v_i$ and $u_{i+1} = v_{i+1}$). In this case, $u_iv_{i+1}u_{i+1}$ is a 2 hop path between u_i and u_{i+1} whose length is equals the cost of v_{i+1}. Thus, the length of $\widetilde{uv}$ is smaller than or equals $c(v_{i+1})$, which is not greater than $c(v_i)+c(v_{i+1})$.

Case 4: $u_i \ne v_i$ and $u_{i+1} \ne v_{i+1}$. In this case, $u_iv_iv_{i+1}u_{i+1}$ is a 3 hops path between u_i and u_{i+1} whose length is equals $c(v_i)+c(v_{i+1})$. Thus, the length of $\widetilde{uv}$ is smaller than or equals $c(v_i)+c(v_{i+1})$.

Thus we have $c(\widetilde{u_iu_{i+1}}) \le c(v_i)+c(v_{i+1})$ for $1\le i\le k-1$. Similarly, we also have $c(\widetilde{uu_1}) \le c(v_1)$ and $c(\widetilde{u_kv}) \le c(v_k)$. Sum all these inequalities, we got

$$\mathcal{L}(u,v,VirtG) \le c(\widetilde{uu_1})+c(\widetilde{v_kv})+\sum_{i=1}^{k-1} c(\widetilde{u_iu_{i+1}}) \le 2\sum_{i=1}^{k} c(v_i).$$

This finishes our proof. ◻

In graph UDG, we set all dominators' cost to 0 to obtain a new graph UDG', assume T_{opt} is the tree with the minimum cost that spanning all dominators selected by Algorithm 1. Following lemma shows that there exists a tree T'_{opt} whose cost equals T_{opt}'s and every non-dominant node u in T'_{opt} has a node degree less than 6.

[2] If several paths have the same length, break the tie by selecting the path with the maximum summation of node's IDs.

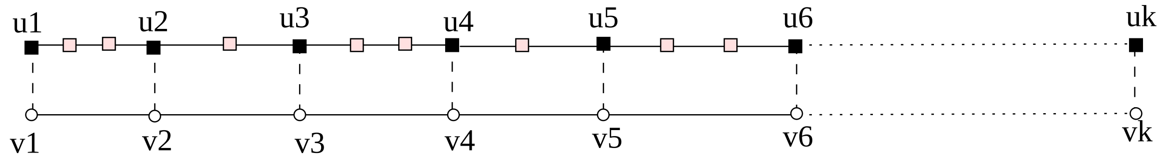

Fig. 5. $\mathcal{L}(u, v, G)$ is no smaller than 2 times the $\mathcal{L}(u, v, VirtG)$.

Lemma 5: There exists a tree T'_{opt} in UDG' spanning all dominators selected in Algorithm 1 and connectors in this tree has degree at most 5.

PROOF. We prove by construction. Consider any optimum cost tree T_{opt} spanning all dominators. In tree T_{opt}, assume there exist some connectors whose degree is greater than 5, we choose any one of them as the root. The depth of a connector is defined as the hops from this connector to the root in tree T_{opt}. We process all connectors u in tree T_{opt} whose degree is greater than 5 in an increasing order of their depths. Assume nodes p, q are u's neighbors in tree T_{opt} such that the $\angle puq$ is minimal. Notice either p or q's depth is greater than u, thus without loss of generality, we can assume p's depth is bigger. By removing edge uq and add edge up, u's degree decreases by 1 while all other connectors whose depth is less than or equals u's remains unchanged. Notice this will result in a new tree spanning all dominators while keep the cost of the tree unchanged. Repeat the above iteration until all connectors' degree is less than 6, the resulting tree is T'_{opt}. □

For tree T'_{opt}, we define its weight $c(T'_{opt})$ as the sum of the cost of all connectors. We also define $c(T) = \sum_{e \in T} c_e$ for an edge weighted tree T.

Theorem 6: Algorithm 2 selects a set of connectors whose total cost is no more than 10 times of the optimum.

PROOF. Let K_{UDG} be another virtual complete graph with vertices from all dominators selected in Algorithm 1 and edge length equals the cost of least cost path between two dominators on original graph UDG. Following we argue the weight of MST on graph K_{UDG} is at most 5 times the weight of tree T'_{opt}.

For spanning tree T'_{opt}, we root it at an arbitrary node and duplicate every link in T'_{opt} (the resulting structure is called DT'_{opt}. Clearly, every node in DT'_{opt} has even degree now. Thus, we can find an Euler circuit, denoted by $EC(DT'_{opt})$, that uses every edge of DT'_{opt} exactly once, which is equivalent to say that every edge in $T'_{opt}(G)$ is used exactly twice. Consequently, we know that every node v_k in $V(T'_{opt})$ is used exactly $deg_{T'_{opt}}(v_k)$ times. Here $deg_G(v)$ denotes the degree of a node v in a graph G. Thus, the total weight of the Euler circuit is at most 5 times of the weight $c(T'_{opt})$, i.e.,

$$c(EC(DT'_{opt})) \leq 5 \cdot c(T'_{opt}).$$

Notice that here if a node v_k appears multiple times in $EC(DT'_{opt})$, its weight is also counted multiple times in $c(EC(DT'_{opt}))$.

If we walk along $EC(DT'_{opt})$, we visit all receivers, and length of any subpath between dominators d_i and d_j is not smaller than $\mathcal{L}(d_i, d_j, G)$. Thus, the cost of $EC(DT'_{opt})$ is at least $c(MST(K_{UDG}))$ since $MST(K_{UDG})$ is the minimum spanning tree spanning all receivers and the cost of the edge $d_i d_j$ in $MST(K_{UDG})$ corresponds the path with the least cost between d_i and d_j. In other words,

$$c(EC(DT'_{opt})) \geq c(MST(K_{UDG})).$$

Consequently, we have

$$c(MST(K_{UDG})) \leq c(EC(DT'_{opt})) \leq 5 \cdot c(T'_{opt}). \quad (1)$$

Now we prove the weight of $MST(VirtG)$ is at most two times the weight of $MST(K_{UDG})$. For any edge $e = d_i d_j \in MST(K_{UDG})$, from Lemma 4, we have

$$c_e \geq \mathcal{L}(d_i, d_j, UDG) \geq \frac{\mathcal{L}(d_i, d_j, VirtG)}{2}.$$

For each edge $e = d_i d_j \in MST(K_{UDG})$, we connect them in graph $VirtG$ using path $\mathsf{LCP}(d_i, d_j, VirtG)$. This constructs a connected subgraph $VirtG'$ on graph $VirtG$ whose cost is not greater than twice the weight of $MST(K_{UDG})$. Thus, we have

$$c(MST(VirtG)) \leq c(VirtG') \leq 2c(MST(K_{UDG})). \quad (2)$$

Combining inequalities (1) and (2) we have

$$c(MST(VirtG)) \leq 2c(MST(K_{UDG})) \leq 10c(T'_{opt}).$$

This finishes the proof. □

Combining Theorem 2 and Theorem 6, we get the following theorem which is the main result of this paper.

Theorem 7: Our algorithm constructs a weighted connected dominating set whose total cost is no more than $\min(18 \log d, 4\delta + 1) + 10$ times of the optimum.

Notice that since the graph $VirtG$ has only linear number of links, we can construct the minimum spanning tree on $VirtG$ using $O(n \log n)$ number of messages. In practice, we may not need construct the minimum spanning tree exactly: a localized approximation of the minimum spanning tree [27] may perform well enough, which has a message complexity only $O(n)$.

C. Routing and Performance

After we construct the backbone WCDS and the induced graph WCDS', if a node u wants to broadcast a message, it follows the following procedure. If node u is not a dominator, then it sends the message to one of its dominators. When the message reaches the backbone, it will be broadcasted along the virtual minimal spanning tree. In previous section, we prove that the total cost of WCDS is no more than a constant times of the optimum.

When considering unicast routing, we can modify our backbone formation algorithm by removing steps 5 and 6 (building $VMST$). Let $UWCDS$ and $UWCDS^{'}$ be the constructed backbone and its induced graph. If a node u wants to unicast a message, it follows the following procedure. If node u is not a dominator and node v is not a neighbor of u, node u sends the message to one of its dominators. Then the dominator will transfer the message to the target or a dominator of the target through the backbone. Now, we prove that the backbone is a spanner for unicast application, i.e., every route in the constructed network topology is efficient. Remember a route is *efficient* if its costs or hops is no more than a constant factor of the minimum length or hops needed to connect the source and the destination in the original communication graph. The constant is called cost or hops stretch factor.

Similar with the proof in [21], [22], we can easily prove the following lemma:

Lemma 8: For any communication graph (not necessarily a UDG model), the hops stretch factor of UWCDS$^{'}$ is bounded by a constant 3.

Then we prove the backbone has a bounded cost stretch factor:

Lemma 9: For any communication graph (not necessarily a UDG model), the cost stretch factor of UWCDS$^{'}$ is bounded by a constant 3.

PROOF. Consider any source node s and target node t that are not connected directly in the original communication graph G. Assume the least cost path $\mathsf{LCP}(s,t,G)$ from s to t in G is $\Pi_{G_h}(s,t) = v_1 v_2 ... v_k$, where $v_1 = s$ and $v_k = t$, as illustrated by Figure 5. We construct another path in UWCDS$^{'}$ from s to t and the total cost of this path is at most 3 times of the cost of the least cost path $\mathsf{LCP}(s,t,G)$.

For any dominatee node p in original communication graph G, we will show that there must exist one dominator q whose cost is not greater than p's cost. First, from our selection procedure of the maximal independent set, node p is not selected to MIS implies that, at some stage, there is a neighbor, say u, with smaller cost selected to MIS, which will be PossibleDominator. Notice that, this PossibleDominator node u may not appear in our final structure. However, this node is not selected only if $c(GRDY_u)$ is smaller than $c(u)$. Notice that clearly, there is at least one node, say v, in $GRDY_u$ that dominates node p since p is a one-hop neighbor of node u and $GRDY_u$ covers all one hop neighbors of u (including u). Clearly, all dominators in $GRDY_u$ has cost no more than $c(u)$ from $c(GRDY_u) \leq c(u)$. If node u is in final structure, we set q as u, otherwise, set q as node v. We call node q as node p's *small dominator*. Notice that q and p can be the same node.

For each node v_i in the path $\mathsf{LCP}(s,t,G)$, let u_i be its small dominator if v_i is not a dominator, else let u_i be v_i itself. Notice that there is a 3-hop path $u_i v_i v_{i+1} u_{i+1}$ in the original communication graph G. Then from Algorithm 2, we know there must exist one or two connectors connecting u_i and u_{i+1}, and also the cost summation of these connectors is at most the cost summation of v_i and v_{i+1}. We define a path, denoted by $\mathsf{LCP}(s,t,UWCDS')$, to connect s and t in UWCDS$'$ as the concatenation of paths $\mathsf{LCP}(u_i,u_{i+1},VirtG)$, for $1 \leq i \leq k-2$, and a least cost path (with $\leq$ two hops) connecting u_{k-1} and t. Remember that the path $\mathsf{LCP}(u_i,u_{i+1},VirtG)$ is only the least cost path among all paths connecting u_i and u_{i+1} using at most 3 hops.

We then show that the path $\mathsf{LCP}(s,t,UWCDS')$ has a cost no more than 3 times of the path $\mathsf{LCP}(s,t,G)$, where $\mathsf{LCP}(s,t,G)$ is the least cost path connecting s and t in the original communication graph G. Clearly, $\sum_{i=1}^{k-2} \mathcal{L}(u_i,u_{i+1},VirtG) \leq c(v_1) + 2\cdot\sum_{i=2}^{k-2} c(v_i) + c(v_{k-1})$. Notice that, in our unicast routing algorithm, when the target node t is within two hops of the dominator node u_{k-1}, node u_{k-1} will not send the data to dominator node u_k. Instead, if target t is one hop neighbor of node u_{k-1}, it will directly send data to node t; otherwise, node u_{k-1} will find a least cost node, say w, to connect to the target node t directly. Obviously, $c(w) \leq c(v_{k-1})$ since node v_{k-1} connects u_{k-1} and target t. Thus, the total cost of the path in the constructed backbone is

$$\begin{aligned}
&\sum_{i=1}^{k-2} \mathcal{L}(u_i,u_{i+1},VirtG) + \mathcal{L}(u_{k-1},t,VirtG) + \sum_{i=1}^{k-1} c(u_i) \\
&\leq c(v_1) + 2\cdot\sum_{i=2}^{k-2} c(v_i) + c(v_{k-1}) + c(v_{k-1}) + \sum_{i=1}^{k-1} c(v_i) \\
&< 3\cdot\sum_{i=1}^{k-1} c(v_i).
\end{aligned}$$

This finishes our proof. ◻

IV. SUMMARY AND FUTURE WORK

In this paper, we present a new algorithm to construct a sparse structure for network backbone in wireless ad hoc networks. A communication efficient distributed algorithm was presented for the construction of a weighted connected dominating set, whose size is guaranteed to be within a constant factor ($\min(18\log d, 4\delta+1)+10$) of the minimum. We also show that WCDS is efficient for both length and hops and has at most $O(n)$ edges. This topology can be constructed locally and is easy to maintain when the nodes move around. All our algorithms have the message complexity $O(n)$ when geometry information is available. Moreover, the number of messages sent by *each* node is bounded by a constant.

There are many interesting open problems left for further study. Remember that, we use the following assumptions on wireless network model: omni-directional antenna, single transmission received by all nodes within the vicinity of the transmitter, nodes being static for a reasonable period of time. To prove that the backbone has low cost, we also assume that all nodes have the same transmission range. Notice that the efficiency property for unicast does not require the communication graph to be a UDG. The problem will become much more complicated if we relax some of these assumptions. Another interesting open problem is to study the dynamic updating of the backbone efficiently when nodes are moving in a reasonable speed. It is interesting to see the practical performance differences of all proposed methods such as methods by Baker *et al.*, Alzoubi *et al.*, and our methods proposed here, in mobile environment. Further future work is to lower the constant bounds given in this paper.

REFERENCES

[1] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *ACM/Kluwer Wireless Networks*, vol. 7, no. 6, 2001.

[2] Gruia Călinescu, Ion Măndoiu, Peng-Jun Wan, and Alexander Zelikovsky, "Selecting forwarding neighbors in wireless ad hoc networks," in *ACM DialM*, 2001.

[3] Brad Karp and H.T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, 2000.

[4] Young-Bae Ko and Nitin H. Vaidya, "Using location information to improve routing in ad hoc networks," Tech. Rep., Department of Computer Science, Texas A&M University, 1997.

[5] Khaled Alzoubi, Peng-Jun Wan, and Ophir Frieder, "Message-optimal connected-dominating-set construction for routing in mobile ad hoc networks," in *3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, 2002.

[6] B. Das and V. Bharghavan, "Routing in ad-hoc networks using minimum connected dominating sets," in *1997 IEEE International Conference on on Communications (ICC'97)*, 1997, vol. 1, pp. 376–380.

[7] Jie Wu and Hailan Li, "A dominating-set-based routing scheme in ad hoc wireless networks," *the special issue on Wireless Networks in the Telecommunication Systems Journal*, vol. 3, pp. 63–84, 2001.

[8] Jie Wu, F. Dai, M. Gao, and I. Stojmenovic, "On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks," *IEEE/KICS Journal of Communication and Networks*, vol. 4, no. 1, pp. 59–70, 2002.

[9] Ben Liang and Zygmunt J. Haas, "Virtual backbone generation and maintenance in ad hoc network mobility management," in *Proc. of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM*, 2000, vol. 3, pp. 1293–1302.

[10] Jie Wu and Hailan Li, "Domination and its applications in ad hoc wireless networks with unidirectional links," in *Proc. of the International Conference on Parallel Processing 2000*, 2000, pp. 189–197.

[11] Khaled M. Alzoubi, Peng-Jun Wan, and Ophir Frieder, "New distributed algorithm for connected dominating set in wireless ad hoc networks," in *HICSS, Hawaii*, 2002.

[12] D. J. Baker and A. Ephremides, "The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm," *IEEE Transactions on Communications*, vol. COM-29, no. 11, pp. 1694–1701, November 1981.

[13] D.J. Baker, A. Ephremides, and J. A. Flynn, "The Design and Simulation of a Mobile Radio Network with Distributed Control," *IEEE Journal on Selected Areas in Communications*, pp. 226–237, 1984.

[14] S. Basagni, "Distributed clustering for ad hoc networks," in *Proceedings of the IEEE International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN)*, 1999, pp. 310–315.

[15] I. Chlamtac and A. Farago, "A new approach to design and analysis of peer to peer mobile networks," *Wireless Networks*, vol. 5, pp. 149–156, 1999.

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

[17] M. Gerla and J. T.-C. Tsai, "Multicluster, mobile, multimedia radio network," *Wireless Networks*, vol. 1, no. 3, pp. 255–265, 1995.

[18] Chunhung Richard Lin and Mario Gerla, "Adaptive clustering for mobile wireless networks," *IEEE Journal of Selected Areas in Communications*, vol. 15, no. 7, pp. 1265–1275, 1997.

[19] J. Gao, L. J. Guibas, J. Hershburger, L. Zhang, and A. Zhu, "Discrete mobile centers," in *Proceedings of the 17th ACM Symposium on Computational Geometry (SoCG 01)*, 2001, pp. 188–196.

[20] Khaled M. Alzoubi, *Virtual Backbone in Wireless Ad Hoc Networks*, Ph.D. thesis, Illinois Institute of Technology, 2002.

[21] Yu Wang and Xiang-Yang Li, "Geometric spanners for wireless ad hoc networks," in *Proc. of 22nd IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2002.

[22] Khaled Alzoubi, Xiang-Yang Li, Yu Wang, Peng-Jun Wan, and Ophir Frieder, "Geometric spanners for wireless ad hoc networks," *IEEE Transactions on Parallel and Distributed Processing*, vol. 14, no. 4, pp. 408–421, 2003, Short version in IEEE ICDCS 2002.

[23] Xiang-Yang Li, G. Calinescu, and Peng-Jun Wan, "Distributed construction of planar spanner and routing for ad hoc wireless networks," in *21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2002, vol. 3.

[24] Peng-Jun Wan and Xiang-Yang Li, "Theoretically good distributed cdma/ovsf code assignment for wireless ad hoc networks," Manuscript, 2003.

[25] Lichun Bao and J. J. Garcia-Luna-Aceves, "Topology management in ad hoc networks," in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. 2003, pp. 129–140, ACM Press.

[26] Gruia Călinescu, "Computing 2-hop neighborhoods in ad hoc wireless networks," in *AdHoc-Now 03*, 2003.

[27] Xiang-Yang Li, Yu Wang, Peng-Jun Wan, Wen-Zhan Song, and Ophir Frieder, "Localized low weight graph and its applications in wireless ad hoc networks," in *IEEE INFOCOM 2004*, 2004.