Geometric Spanners for Wireless Ad Hoc Networks

Yu Wang Xiang-Yang Li
Department of Computer Science
Illinois Institute of Technology
10 W. 31st Street
Chicago, IL 60616, USA
Email: wangyu1@iit.edu,xli@cs.iit.edu

Abstract—**We propose a new geometric spanner for wireless ad hoc networks, which can be constructed efficiently in a distributed manner. It integrates the connected dominating set and the local Delaunay graph to form a backbone of the wireless network. Priori arts showed that both structures can be constructed locally with bounded communication costs. This new spanner has these following attractive properties: (1) the backbone is a planar graph; (2) the node degree of the backbone is bounded from above by a positive constant; (3) it is a spanner for both hops and length; (4) it can be constructed locally and is easy to maintain when the nodes move around; (5) moreover, the computation cost of each node is at most $O(d \log d)$, where d is its 1-hop neighbors in the original unit disk graph, and the communication cost of each node is bounded by a constant. Simulation results are also presented for studying its practical performance.**

Keywords— **Connected dominating set, clustering, Delaunay triangulation, spanner, unit disk graph, localized algorithm, wireless ad hoc networks.**

I. INTRODUCTION

Wireless ad hoc networks [1] 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. In the most common power-attenuation model, the power required to support a link between two nodes separated by distance r is r^α, where α is a real constant between 2 and 5 dependent on the wireless transmission environment. Here we ignore the overhead cost of each node to receive and process the coming signal. 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.

Due to the nodes' limited resource in wireless ad hoc networks, scalability is crucial for network operations. One effective approach is to maintain only a linear number of links using a localized construction method. However, this sparseness of the constructed network topology should not compromise too much on the power consumptions on communications. In this paper, we study how to construct a sparse network topology efficiently for a set of wireless nodes such that every route in the constructed network topology is efficient. Here a route is *efficient* if its length 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.

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. For example, the sensors in the sensor network do not move usually. Notice that, our algorithms do not need to update the network topology when nodes are moving as long as no link used in the final network topology is broken. In other words, although the actual physical deployment of the network topology is no longer a planar graph when nodes are moving, the logical network topology is still a planar graph, which is crucial for some routing algorithm.

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 [2], [3], [4], [5], or to use a hierarchical structure like Internet. Examples of hierarchical routing are dominating set based routings [6], [7], [8], [9]. When each wireless node knows its geometry position and can quickly retrieve[1] the geometry information about the destination node of a routing request, several localized routing methods based on geometrical forwarding [10], [2] are proposed to avoid the flooding. Recently, Karp and Kung [10] proposed a new protocol, *Greedy Perimeter Stateless Routing* (GPSR), which routes the packets on a planar subgraph of UDG and guarantees the delivery of the packet if there exists a path. Bose, *et al.* [2] also proposed a similar method using Gabriel graph as planar subgraph. Relative neighborhood graph is also used in broadcasting [11]. These methods maintain some planar subgraph such as the relative neighborhood graph (RNG) or Gabriel graph (GG) as underlying network topology. The routing is based on geometry forwarding heuristics and the right hand rule is used temporarily when a local minimum occurs. It was known that the RNG and GG are not good spanners for UDG [12], [13]. Recently, Gao, *et al.* [14] proposed a new method to construct sparse spanners. The method combines the node clustering algorithm with a new routing graph, called *Restricted Delaunay Graph* (RDG). Although their clustering algorithm [15] achieves a constant approximation in expectation, the approximation constant is too large for having any practical meaning. Additionally, the method of constructing RDG is not communication effective.

Consequently, we focus on constructing a sparse network

[1] For example, for sensor networks collecting environmental data such as temperature, the data are typically sent to one specific node called sink. In this case, we can assume that the sink node is static and its position is known to all other nodes. The other way to get the location information of a node is to use GPS and location service.

topology, i.e., a subgraph of $UDG(V)$, which has the following desirable features.

Sparseness The topology should be a sparse graph, i.e., the total number of links in this network topology is linear with the total number of wireless nodes. This enables most of algorithms, e.g., routing algorithm based on the shortest path, to run on this topology more efficiently in term of both time and power consumption.

Spanner We want the subgraph to be a spanner of $UDG(V)$ in terms of both length and hops. Here a subgraph $G' \subseteq G$ is a spanner of graph G for length if there is a positive real constant t_1 such that for any two nodes, the length of the shortest path connecting them in G' is at most t_1 times of the length of the shortest path connecting them in G. The constant t_1 is called the *length stretch factor*. In the same way, a subgraph G' is a spanner of a graph G for hops means that there is a positive real constant t_2 such that the number of hops of the shortest hops path in G' is at most t_2 times of the number of hops of the shortest hops path in G between any two nodes. The constant t_2 is called the *hops stretch factor*. Similarly, we can define the *power stretch factor* [12].

Bounded degree Because every wireless node has a limited computational resources, storage, and more importantly, a limited power, the degree of each node in the constructed topology should be bounded by a constant, so that each node only needs to hold and process a constant number of neighbors.

Planarized The topology is a planar graph (i.e., no two edges cross each other in the graph). Some routing algorithms, such as right hand routing and *Greedy Perimeter Stateless Routing* (GPSR) [10], require the topology be planar.

Efficient Localized Construction Due to the limited resources of the wireless nodes, it is preferred that the underlying network topology can be constructed and maintained in a localized manner. Here a distributed algorithm constructing a graph G is a *localized algorithm* if every node u can exactly decide all edges incident on u based only on the information of all nodes within a constant hops of u (plus a constant number of additional nodes' information if necessary). More importantly, we expect that the time complexity of each node running the algorithm to construct the underlying topology is at most $O(d \log d)$, where d is the number of 1-hop neighbors; in addition, the number of messages sent by each node is at most a constant.

A trade-off can be made between the sparseness of the topology and the power efficiency. However, not all sparse subgraphs are good candidates for the underlying network topologies. There are two sets of structures used for wireless networks: flat structures, and hierarchical structures. The flat structures used previously include the relative neighborhood graph, Gabriel graph, Yao structure, and the Delaunay triangulation. On the other hand, the hierarchical structures used typically are based on dominating set, or connected dominating set, or their extensions such as d-dominating set [16].

In [10], Karp and Kung used two planar subgraphs: the *relative neighborhood graph* (RNG), and the *Gabriel graph* (GG). However, Bose, *et al.* [13] proved that the length stretch factors of these two graphs are $\Theta(n)$ and $\Theta(\sqrt{n})$ respectively. Recently, some researchers [17], [18] proposed to construct the wireless network topology based on Yao graph (also called θ-graph). It is known that the length stretch factor and the node out-degree of Yao graph are bounded by some positive constants. But as Li, *et al.* mentioned in [18], all these three graphs cannot guarantee a bounded node degree, e.g., for Yao graph, the node in-degree could be as large as $O(n)$. In [12], [18], Li, *et al.* further proposed to use another sparse topology, *Yao and Sink*, that has both a constant bounded node degree and a constant bounded length stretch factor. It is a spanner for length or power, but not for hops. It is easy to construct a configuration of a set of nodes, for example, n nodes evenly distributed on a unit segment, such that the Yao structure is not a hop spanner. In addition, all these graphs, which are related with Yao graph, are not guaranteed to be planar graphs.

Many researchers proposed to use the *connected domination set* (CDS) as a virtual backbone for hierarchical routing in wireless as hoc networks [7], [19], [8], [20]. Efficient distributed algorithms for constructing connected dominating sets in ad hoc wireless networks were well studied [21], [22], [23], [24], [25], [7], [8], [26]. The notion of cluster organization has been used for wireless ad hoc networks since their early appearance. Baker *et al.* [22], [23] 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.* [27], [28] 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 [15], 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.* [21] 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 [29] 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 [23], [28], or based on the maximum node degree [27], or based on some generic weight [24] (node with the largest weight will be chosen as clusterhead). Notice that, any maximal independent set is always a dominating set. Several clustering method essentially computes a maximal independent set as the final clusterheads. We will use any method that can build the MCDS efficiently, such as those by Alzoubi [29], or by Baker [22], [23], and then build the local Delaunay graph on top of the MCDS. Notice, local Delaunay graph was recently proposed in [30]. We will show that the resulting graph has all properties we listed before. In other words, it is a hybrid sparse spanner for network topology.

The constructed backbone is not always a planar graph, while the planarity is required by several geometry-based localized routing algorithms. Then a *localized Delaunay triangulation* (LDel) of CDS is set as the backbone of the network. We show that the constructed backbone is a planar graph and each node has a bounded degree. All ordinary nodes are connected to their dominators. We show that the constructed subgraph is spanner for both length and hops and has at most $O(n)$ edges. The total communication cost of this method is $O(n)$, which is within a constant factor of the optimum. Moreover, the communication cost of *each* node is bounded by a constant. The computation cost of each node is at most $O(d \log d)$, where d is the number of its 1-hop neighbors. We also conduct experiments to show that this topology is efficient in practice. To the best of our knowledge, this is the first one to generate planar backbone while the communication cost of *each* wireless node is bounded by a constant. This is more attractive since the communications in wireless networks are the most power consuming operations.

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 network topology design issues. Section III presents our new spanner formation algorithms based on CDS and LDel graphs. In addition, we prove some properties of the new spanner. Section IV presents the experimental results. We conclude our paper in Section V by pointing out some possible future research directions.

II. GEOMETRY DEFINITIONS AND NOTATIONS

In this section, we give some geometry 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.

Various proximity subgraphs of the unit disk graph can be used in ad hoc wireless networks [10], [12], [17], [18], such as the *relative neighborhood graph*, the *Gabriel graph* and the *Yao graph*. None of these graphs is hop-spanner. In contrast, we use a *connected dominating set* (CDS) as a virtual backbone of the wireless network, and use *localized Delaunay graph* (LDel) to make the backbone planar. See Figure 1 for an illustration of when an edge is included in a graph defined.

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. 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 denoted by MCDS.

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.

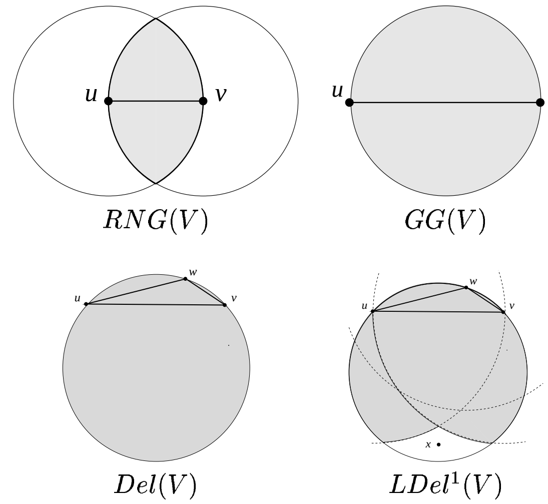


Fig. 1. Definitions of various topologies. The shaded area is empty of nodes inside.

We continue with definition of the Delaunay triangulation. A triangulation of V is a *Delaunay triangulation*, denoted by $Del(V)$, if the circumcircle of each of its triangles does not contain any other vertices of V in its interior. We assume that there are no four vertices of V that are co-circular. A triangle is called the *Delaunay triangle* if its circumcircle is empty of vertices of V. Keil and Gutwin [31], [32] showed the *Delaunay triangulation* is a planar spanner with the length stretch factor as most $\frac{4\sqrt{3}}{9}\pi \approx 2.42$.

However, the main drawback of applying the Delaunay triangulation in the ad hoc wireless environment is that it can not be constructed locally. In [30], Li, *et al.* defined a new geometry structure, called *k-localized Delaunay graph* (LDel^k), and they presented a distributed algorithm to construct it efficiently. A triangle $\triangle uvw$ satisfies *k-localized Delaunay property* if its circumcircle, denoted by $disk(u, v, w)$, does not contain any vertex from $N_k(u) \cup N_k(v) \cup N_k(w)$ inside and all edges of the triangle $\triangle uvw$ have length no more than one unit. Triangle $\triangle uvw$ is called a *k-localized Delaunay triangle*. An edge uv is a Gabriel edge if the disk using uv as diameter does not contain any vertex inside and $\|uv\| \leq 1$. The *k-localized Delaunay graph* over a vertex set V, denoted by $LDel^{(k)}(V)$, has exactly all Gabriel edges and the edges of all k-localized Delaunay triangles. Li, *et al.* [30] proved that local Delaunay triangulation LDel^k is a planar graph for $k \geq 2$ and has thickness 2 if $k = 1$. Here a graph G has *thickness* t if G can be decomposed into t planar graphs but not $t - 1$ planar graphs.

Notice that, the definition of *k-localized Delaunay graph* (LDel^k) by Li, *et al.* [30] is different from the definition of *Restricted Delaunay graph* (RDG) by Gao, *et al.* [14]. Let $UDel(V) = Del(V) \cap UDG(V)$, i.e., the edges in Delaunay triangulation with length at most one unit. Gao, *et al.* [14] called *any* planar graph containing $UDel(V)$ as a RDG. They gave a method to construct a RDG. However, their method is not communication efficient, nor computation efficient. The worst time communication cost is equal to the number of links in the unit disk graph, which could be $O(n^2)$. The worst computation cost of a node is $O(d^3)$, where d is the number of its 1-hop neighbors.

III. New Spanner Formation Algorithms

We begin this section by proposing the localized planar backbone formation algorithms, based on the *connected dominating set* and the *localized Delaunay triangulation*.

A. Formation of Backbone

Several efficient methods [29], [21], [22], [23], [28] for constructing CDS were developed. Previous algorithms for building CDS typically have two phases: clustering and finding connectors (or called gateways). The clustering algorithm basically finds a subset of nodes such that the rest of the nodes are visible to at least one of the cluster-heads. By definition, any algorithm generating a maximal independent set is a clustering method. Various methods can then be used to connect the cluster-heads to form a connected graph. For the completeness of presentation, we will review some priori arts on building CDS, MCDS, and localized Delaunay graph. 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*.

A.1 Clustering

Many algorithms for clustering have been proposed in the literature [21], [29], [22], [23], [24], [7], [28], [25], [16], [33], [34], [8]. All algorithms assume that the nodes have distinctive identities (denoted by ID hereafter). We will typically review the ones by Baker [22], [23] and Alzoubi [29]. For the sake of general description of these priori arts, we will summarize them using our own words. The methods typically use two messages IamDominator and IamDominatee, and have the following procedures: a white node claims itself to be a dominator if it has the smallest ID among all of its white neighbors, if there is any, and broadcasts IamDominator to its 1-hop neighbors. A white node receiving IamDominator message marks itself as dominatee and broadcasts IamDominatee to its 1-hop neighbors. The set of dominators generated by the above method is actually a maximal independent set since no two adjacent nodes will be marked as dominators. Here, we assume that each node knows the IDs of all its 1-hop neighbors, which can be achieved by requiring each node to broadcast its ID to its 1-hop neighbors initially. This protocol can be easily implemented using synchronous communications as did in [22], [23]. 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 largest ID among all white neighbors.

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* dominators in the unit disk graph model. For the completeness of presentation, we include a short proof here.

Lemma 1: For every dominatee node v, it can be connected to at most 5 dominator nodes in unit disk graph model.

PROOF. For the sake of contradiction, assume that a node v has 6 dominator neighbors. We know in the unit disk centered at v there must have 2 dominator neighbors w and u, the angle $\angle wvu$ is at most $\frac{\pi}{3}$. So the distance between w and u must be no more than one unit, which means that there is an edge between w and u in UDG. This is a contradiction with the definition of maximal independent set. ◻

Generally, for each node (dominator or dominatee), there are at most a constant number of dominators that are at most k units away.

Lemma 2: For every node v, the number of dominators inside the disk centered at v with radius k-units is bounded by a constant ℓ_k.

PROOF. Because any two dominators are at least one unit away, the half-unit disks centered at dominators are disjoint with each other. In addition, all such dominators should be in the disk centered at v and with radius k. Then ℓ_k is bounded by how many disjoint half-unit disks we can park in the disk centered at v with radius $k+0.5$. See Figure 2. We have $\ell_k \leq \frac{\pi(k+0.5)^2}{\pi(0.5)^2} = (2k+1)^2$ using an area argument. When $k = 2, 3, 4$, we have $\ell_k \leq 25, 49, 81$.

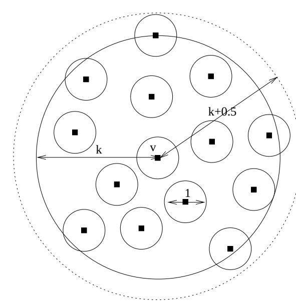


Fig. 2. For every node v, the number of dominators within k units is bounded by a constant ℓ_k.

◻

The bounds on l_k can be improved by a tighter analysis. The above lemma implies that for every node v, the number of dominators within k hops is bounded by a constant ℓ_k.

Almost all clustering methods described before are similar to this synchronous protocol. The difference of these previous methods lyes in how to find gateways to connect these clusterheads. For example, the basic algorithm for constructing CDS proposed in [27] 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 is not specified in the paper. Additionally, no performance guarantee is proved. In [22], [23], they consider in detail of 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 on 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.* [6], [29] proposed a localized method to find connectors using total $O(n)$ messages and showed that the constructed CDS is within a constant factor of optimum. This property enable us to build a planar spanner in total linea number of messages, which is crucial for wireless ad hoc networks since the communication is the most energy-consumption operation. Actually, we will show that the method by Baker *et al.* also has linear number of messages, and the size of the constructed structure is also within a constant factor of optimum. We will discuss in detail of these two methods, which will be the first phase of our hybrid method.

A.2 Finding Connectors

The second step of 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*.

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. Then graph $VirtG$ is connected. This observation is a basis of several algorithms [22], [23], [27], [28] for CDS, although no proof was given in these previous results. 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 [21], [29], [22], [23], [28]. Let $\Pi_{UDG}(u, v)$ be the path connecting two nodes u and v in UDG with the smallest number of hops. Let's first consider how to connect two dominators within 3 hops. The method by Alzoubi *et al.* [21], [29], [35] choosed the connectors as follows: (1) If the path $\Pi_{UDG}(u, v)$ has two hops, then u finds the dominatee node with the smallest ID to connect u and v; (2) If the path $\Pi_{UDG}(u, v)$ has three hops, then u finds the node, say w, with the smallest ID such that w and v are two hops apart. Then node w selects the node with the smallest ID to connect w and v. Thus, basically, node u will decide the next node on the path to connect to another node v. Notice that, it is not obvious how a dominator node u can find such node w as a connector efficiently. In addition to that, using the smallest ID is not efficient because we may have to postpone the selecting of connectors till the node collects the IDs of all its one-hop neighbors. Instead of using the intermediate node with the smallest ID, we can pick any node that comes first to the notice of the node that makes the selection of connectors.

Notice that, the above approach is different from that one adopted by Baker *et al.* [22], [23]. In their protocols, they let the dominatee nodes to decide whether they will serve as the connectors (gateways) or not. For example, if a dominatee node finds that it is dominated by two nonadjacent dominators, say u and v, they it claims itself as a candidate of the connectors for u and v. The node with the highest ID among nodes in the intersection area covered by nodes u and v is chosen as the gateway node for the node pair u and v. In other words, they let the nodes in this intersection area to elect the one with the highest ID, but no detailed protocol is given to do so. For the case of nonoverlapping clusters, a pair of adjacent dominatees (one from each cluster) need to claim them as the candidates for the gateways of these two clusters. They always select the pair of dominatees with the largest sum of identity numbers. In case of a tie, the pair involving the node with the highest ID number is chosen. However, unlike the case of overlapping clusters, here we may end up with two or perhaps more gateway pairs. The existence of one pair may not be known to both partners of the other pair [22]. This cannot be avoided without increasing the communications [22]. We modify the method by Baker *et al.* using the framework by Alzoubi *et al.* and show that it does approximate CDS using linear number of communications. We then discuss in detail the approach to optimize the communication cost and the memory cost. It uses the following primitive messages (some messages are used in forming clusters):

- IamDominator(u): node u tells its 1-hop neighbors that u is a dominator;
- IamDominatee(u, v): node u tells its 1-hop neighbors that u is a dominatee of node v;
- TryConnector(u, w, v, i): node w proposes to its 1-hop neighbors that it could be one of the connectors to connect two dominators u and v. Interger i specifies whether it is the first or the second node on the path to connect two dominators u and v. If uv are two hops apart, then set $i = 0$.
- IamConnector(u, w, v, i): node w tells its 1-hop neighbors that it is the connector to connect two nodes u and v.

Notice that the message IamDominator(u) is only broadcasted at most once by each node; the message IamDominatee(u, v) is only broadcasted at most five times by each node u for all possible dominators v from Lemma 1. From Lemma 2, we know that TryConnector(u, w, v, i) are also broadcasted at most a constant times by each node for all possible dominators u and v.

Lemma 3: Each node has to send out at most a constant number of messages in forming a connected dominating set.

Each node uses the following link lists.

- Dominators: it stores all dominators of u if there is any. Notice that if the node itself is a dominator, no value is assigned for Dominators.
- 2HopDominators: it stores all dominators v that are 2 hops apart from u.

From Lemma 2, for each node u, there are at most ℓ_k number of dominators v that are k hops apart from u. The size of each list is bounded by ℓ_1 and ℓ_2 respectively. Then we are in the position to discuss the distributed algorithm for finding connectors, which are built on the framework of Baker [22], [23]. Assume that a maximal independent set is already constructed by a cluster algorithm.

Algorithm 1: Finding Connectors

1. Every dominatee w broadcasts message IamDominatee(w, v) that w is a dominatee of v.
2. Every node x stores its two-hop away dominators from the messages IamDominatee(w, v) broadcasted by its neighbor w.
3. Every dominatee node w broadcasts to its 1-hop neighbors a message TryConnector($u, w, v, 0$) for every dominators pair u and v (stored at Dominators).
4. If node w has the smallest ID among all its neighbors that sent TryConnector($(u, *, v, 0)$), then node w broadcasts IamConnector($u, w, v, 0$).
5. Every dominatee node w broadcasts to its 1-hop neighbors a message TryConnector($u, w, v, 1$) for its dominator u and the

2-hop away dominator v.
6. Similarly, if node w has the smallest ID among all its neighbors that sent TryConnector($(u, *, v, 1)$), then node w broadcasts IamConnector($u, w, v, 1$).
7. After receiving message IamConnector($u, w, v, 1$), every dominatee node x broadcasts to its 1-hop neighbors a message TryConnector($u, x, v, 2$) for the 2-hop away dominator u and its dominator v.
8. Similarly, if node x has the smallest ID among all its neighbors that sent TryConnector($(u, *, v, 2)$), then node x broadcasts IamConnector($u, x, v, 2$).

Notice that it is possible that, given any two nodes u and v, the path found by node u to connect v is different from the path found by v to connect u. Additionally, there may have multiple paths selected to connect two dominators u and v. We will show that the number of connectors are bounded. This increases the robustness of the backbone.

Notice that, for each two hops away dominators pair u and v, there are at most 2 nodes claiming it to be connectors for them. This is because we can put at most put two nodes inside the lune defined by u and v such that they cannot hear each other directly. Notice, if two nodes can hear each other (i.e., neighbors), then they cannot have the smallest ID among all its neighbors that sent TryConnector($u, *, v, 0$).

For two dominators that are three hops away, it is obvious that there are at most five nodes sent out IamConnector($u, *, v, 1$). Moreover, each such sent message will trigger at most another five nodes to send out message IamConnector($u, *, v, 2$). Thus, there are at most 30 connectors introduced for two domminators. Consequently, the total number of connectors introduced is at most a constant factor of the number of domminators in the graph. Previous reviewed localized clustering method can approximate the MDS with constant 5. So the above method will generate a CDS whose size is within a constant factor of the minimum. Additionally, it is obvious that the number of communications by each node is bounded by a constant: there are constant number of domminator pairs within two hops and for each pair the communications is at most 2. One for claiming itself as connector candidate, and one for claiming itself (if necessary) as connector.

The graph constructed by the above algorithm **FindingConnectors** is called a CDS graph (or *backbone* of the network). If we also add all edges that connect all dominatees to their dominators, the graph is called extended CDS, denoted by CDS'. In Figure 3, we presents an example of CDS', where the solid lines in the graph forms the CDS 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*. Connected dominating set CDS induces a graph: two nodes are connected if and only if their distance is no more than one unit. The induced graph is called induced connected dominating set graph (ICDS). Obviously the CDS is a subgraph of ICDS. If we also add all edges that connect all dominatees to their dominators, the graph is called extended induced CDS, denoted by ICDS'. Later we will prove that both CDS' and ICDS' are the hop and distance spanners; both CDS and ICDS have a bounded node degree. Graphs ICDS and ICDS' can be constructed using only one message each node (to tell its neighbors it is whether dominator, dominatee, or connector node) if CDS is constructed.

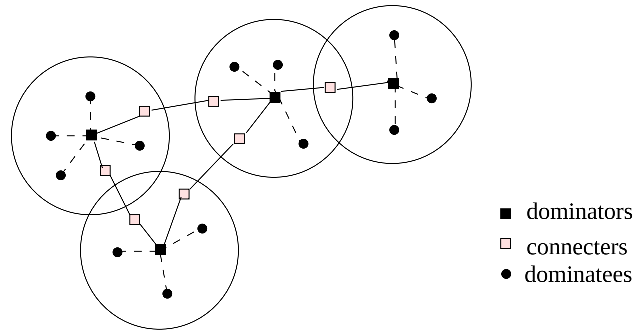


Fig. 3. An example of backbone.

B. *The Properties of Backbone*

This subsection will show that the CDS' graph is a sparse spanner in terms of both hops and length, meanwhile CDS has a bounded node degree.

Lemma 4: The node degree of CDS is bounded by a constant.
PROOF. Consider any node w, there are two cases: w is a dominator node and w is a connector node.

For a dominator w, it can only be connected to some connectors v, which must have some dominators u that are 1-hop or 2 hops away from v. From Lemma 2, we know that the number of this kind of dominators u is bounded by ℓ_3. So the degree of w is also bounded by a constant factor of ℓ_3 since for each such dominator v, there are at most a constant number of connectors introduced and incident on u.

For a connector w, it can be connected to at most 5 dominator nodes and to some connectors. Each of these connectors p should be directly connected to some dominator q, then the number of this kind of dominators q is bounded by ℓ_2. So the degree of w is bounded by a constant. $\square$

The above lemma immediately implies that CDS is a sparse graph, i.e., the total number of edges is $O(k)$, where k is the number of dominators. Moreover, the graph CDS' is also a sparse graph because the total number of the links from dominatees to dominators is at most $5(n-k)$. Notice that we have at most $n-k$ dominatees, each of which is connected to at most 5 dominators. The node degree in CDS is bounded, however, the degree of some dominator node in CDS' may be arbitrarily large.

After we construct the backbone CDS and the induced graph CDS', if a node u wants to send a message to another node v, it follows the following procedure. If v is within the transmission range of u, node u directly sends message to v. Otherwise, node u asks its dominator to send this message to v (or one of its dominators) through the backbone. Then we show that CDS' (plus all implicit edges connecting dominatees that are no more than one unit apart) is a good spanner in terms of both hops and length. In the following proofs, we use $\Pi_{G_h}(s,t)$ and $\Pi_{G_l}(s,t)$ to denote the shortest hop path and the shortest length path in a graph G from node s to node t. Let $l(\Pi)$ and $h(\Pi)$ be the length and the number of hops of path Π respectively. The following proof was similar to that presented by Gao, *et al.* [14]. However, our proof shows that, given any two nodes s and t, there is a *unique* path such that its length is no more than a constant factor of $l(\Pi_{UDG_l}(s,t))$, and its hops is no more than a constant factor of $h(\Pi_{UDG_h}(s,t))$.

Lemma 5: The hops stretch factor of CDS' is bounded by a constant 3.

PROOF. Assume the shortest hop path from s to t in UDG is $\Pi_{UDG_h}(s,t) = v_1 v_2 ... v_k$, where $v_1 = s$ and $v_k = t$, as illustrated by Figure 4. We construct another path in CDS' from s to t and the number of hops of this path is at most $3k+2$.

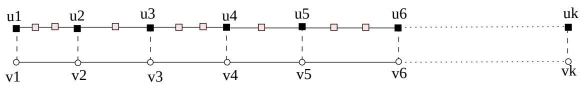


Fig. 4. CDS' is a hop-spanner.

For each node v_i in the path $\Pi_{UDG_h}(s,t)$, let u_i be its 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 UDG. Then from Algorithm 1, we know there must exist one or two connectors connecting u_i and u_{i+1}. Obviously, nodes u_1 and u_k are connected by a path $\Pi_{CDS'}(u_1,u_k)$ in CDS' using at most $3k$ hops. It implies that nodes s and t are connected by a path $\Pi_{CDS'}(s,t)$ (link su_1 followed by $\Pi_{CDS'}(u_1,u_k)$, followed by link $u_k t$) with at most $3k+2$ hops in CDS'. Thus, the hops stretch factor of CDS' is bounded by 3 (with an additional constant 2). ◻

Lemma 6: The length stretch factor of CDS' is bounded by a constant 6.

PROOF. Given any two nodes s and t such that $||st|| > 1$, we will show that the path $\Pi_{CDS'}(s,t)$ constructed in the proof of Lemma 5 has length at most 6 times the length of $\Pi_{UDG_l}(s,t)$.

First, for any path Π, $l(\Pi) \le h(\Pi)$, because the length of every link is no more than one unit. From Lemma 5, we also know that $h(\Pi_{CDS'}(s,t)) \le 3k+2$, where k is the minimum number of hops needed to connect s and t, i.e., $k = h(\Pi_{UDG_h}(s,t))$. Then

$$l(\Pi_{CDS'}(s,t)) \le h(\Pi_{CDS'}(s,t)) \le 3k+2.$$

Notice that, in the shortest path $\Pi_{UDG_l}(s,t) = w_1 w_2 ... w_m$, the sum of each two adjacent links $w_{i-1}w_i$ and $w_i w_{i+1}$ must be larger than one; otherwise we can use link $w_{i-1}w_{i+1}$ instead of $w_{i-1}w_i w_{i+1}$ to find a shorter path from the triangle inequality $||w_{i-1}w_{i+1}|| \le ||w_{i-1}w_i|| + ||w_i w_{i+1}||$. Therefore,

$$l(\Pi_{UDG_l}(s,t)) > \lfloor h(\Pi_{UDG_l}(s,t))/2 \rfloor .$$

Notice that $h(\Pi_{UDG_l}(s,t)) \ge h(\Pi_{UDG_h}(s,t)) = k$. So $k < 2l(\Pi_{UDG_l}(s,t)) + 2$. Then,

$$l(\Pi_{CDS'}(s,t)) \le 3k+2 \le 6l(\Pi_{UDG_l}(s,t)) + 6.$$

Consequently, the length stretch factor of CDS' is bounded by 6 (with an additional constant 6). Here, we are only interested in nodes s and t with $||st|| > 1$. ◻

Similarly, we can show that ICDS has a bounded node degree. As CDS' is a subgraph of ICDS', the hop and length stretch factors of ICDS' are also at most 3 and 6 respectively.

Several routing algorithms require the underlying topology be planar. However, the backbone CDS can be a non-planar graph. Notice in the formation algorithm of CDS, we do not use any geometry information. The resulting CDS maybe non-planar graph. Even using some geometry information, the CDS still is not guaranteed to be a planar graph. Here we give a counter example illustrated by Figure 5. The lengths of link u_1u_2, u_2u_3,

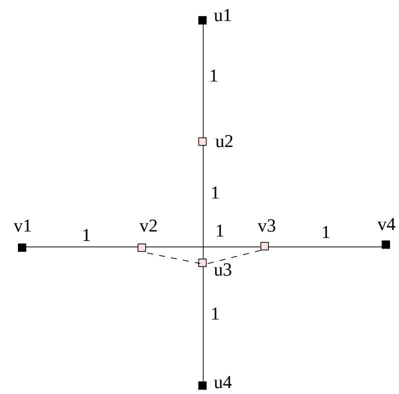


Fig. 5. CDS could be nonplanar.

u_3u_4, v_1v_2, v_2v_3, v_3v_4 are all one unit, while the lengths of link u_2v_2 and u_2v_3 are longer than one. For dominator nodes u_1 and u_4, there is only one 3-hop path $u_1u_2u_3u_4$. So the link u_2u_3 must be in CDS. For the same reason, v_2v_3 must be in CDS. Clearly, u_2u_3 intersects v_2v_3.

C. *Local Delaunay Triangulation on Induced Graph CDS'*

Several localized routing heuristics have been proposed recently for wireless ad hoc networks. Some routing algorithms such as GPSR [10], [2] require the underlying network topology to be planar. However, we know that CDS is not guaranteed to be a planar graph, so do its supergraphs CDS', ICDS and ICDS'. Thus, we cannot directly apply the geometry forwarding based routing algorithms on the backbone CDS or any of its supergraphs. When each node knows its geometry position, however, we can apply the *Localized Delaunay Triangulation* [30] on top of the ICDS graph to planarize the ICDS graph without losing the spanning properties. Herafter, we assume that each wireless node knows absolute or relative positions of itself and each of its neighbors. However, a broad variety of location dependent services will become feasible in the near future. Although the commercial Global Position System (GPS) has accuracy around ten meters, the modern systems have accuracy up to three meters [36]. Indoor location systems are based on the proximity of fixed objects with known coordinates (e.g. sensors), measuring angle of arrival and time delays of signals. Active Badge system, for example, has accuracy within 9 cm of their true position [36], with the work in progress to improve accuracy. If no indoor or outdoor location service is available, the distance between neighboring nodes can be estimated on the basis of incoming signal strengths or time delays.

C.1 Review of Local Delaunay Triangulation

For the completeness of the presentation, we review the algorithms proposed in [30] to construct the local Delaunay triangulation. For a set of nodes V, the algorithm first constructs a graph of $LDel^{(1)}(V)$ and then makes it planar efficiently using the second algorithm.

Algorithm 2: Localized Delaunay Triangulation

1. Each wireless node u broadcasts its location and listens to the messages from other nodes.
2. Assume that every wireless node u gathers the location information of $N_1(u)$. Node u computes the Delaunay triangulation $Del(N_1(u))$ of its 1-neighbors $N_1(u)$, including u itself.
3. Node u finds all Gabriel edges uv and marks them as *Gabriel edges*. Notice that here $\|uv\| \leq 1$, and the disk using uv as diameter is empty.
4. Node u finds all triangles $\triangle uvw$ from $Del(N_1(u))$ such that all three edges of $\triangle uvw$ have length at most one unit. If angle $\angle wuv \geq \frac{\pi}{3}$, node u broadcasts a message proposal(u, v, w) to form a 1-localized Delaunay triangle $\triangle uvw$ in $LDel^{(1)}(V)$ and listens to the messages from other nodes.
5. When node v receives a message proposal(u, v, w), v accepts the proposal of constructing $\triangle uvw$ if $\triangle uvw$ belongs to the Delaunay triangulation $Del(N_1(v))$ by broadcasting message accept(u, v, w); otherwise, it rejects the proposal by broadcasting message reject(u, v, w). Similarly does node w.
6. Node u accepts the triangle $\triangle uvw$ if both nodes v and w accept the message proposal(u, v, w). Similarly do node v and w.

It is easy to show that the total communication cost of the above algorithm is $O(n)$, where n is the number of total input nodes. The computation cost of each node is $O(d \log d)$ (from computing the Delaunay triangulation of $N_1(u)$). It was proved in [30] that the above algorithm does generate $LDel^{(1)}(V)$. It was also proved in [30] that if two 1-Delaunay triangles xyz and uvw intersect, then either the circumcircle of xyz contains one of the vertices in $\{u, v, w\}$ or the circumcircle of uvw contains one of the vertices in $\{x, y, z\}$. We then make this graph $LDel^{(1)}(V)$ planar as follows.

Algorithm 3: Planarize $LDel^{(1)}(V)$

1. Each wireless node u broadcasts the Gabriel edges incident on u and the triangles $\triangle uvw$ of $LDel^{(1)}(V)$ and listens to the messages from other nodes.
2. Assume node u gathered the *Gabriel edges* and 1-localized Delaunay triangles information of all nodes from $N_1(u)$. For two intersected triangles $\triangle uvw$ and $\triangle xyz$ known by u, node u removes the triangle $\triangle uvw$ if its circumcircle contains a node from $\{x, y, z\}$.
3. Each wireless node u broadcasts all remaining triangles incident on u and listens to the broadcasting by other nodes.
4. Node u keeps all triangles $\triangle uvw$ if both v and w have triangle $\triangle uvw$ remaining.

It was proved that the $LDel^{(1)}(V)$ has thickness 2, i.e., it can be decomposed to two planar graphs. Thus, it has at most $6n$ edges. Then it is easy to show that the total communication cost of planarizing $LDel^{(1)}(V)$ is $O(n)$.

C.2 Properties of $LDel(ICDS)$

Applying Algorithm 2 and Algorithm 3 on ICDS, we get a planar graph called $LDel(ICDS)$. Moreover, we will prove that ICDS has a bounded node degree and so does $LDel(ICDS)$. It was proved in [30] that $LDel(G)$ is a spanner if G is a unit disk graph. Notice that ICDS is a unit disk graph defined over all dominators and connectors. Consequently, $LDel(ICDS)$ is a spanner in terms of length. So here we first give a proof that $LDel(ICDS)$ has a bounded hops stretch factor.

Lemma 7: The hops stretch factor of $LDel(ICDS)$ is bounded by a constant.

PROOF. It was proved before that ICDS is a hop-spanner because ICDS contains CDS as a subgraph and CDS is a hop-spanner. Thus, we only have to show that for any link uv in ICDS, there is a path in $LDel(ICDS)$ connecting u and v using a constant number of hops.

It was proved in [30] that the length stretch factor of $LDel(G)$ is at most 2.5 for any unit disk graph G. Therefore, we know that there is a path in $LDel(ICDS)$ with length at most 2.5 to connect u and v. Then all nodes in that path are inside the disk centered at u with radius 2.5. There are two types of nodes inside this disk: dominators or connectors. Inside this disk, obviously there are at most a constant number $\ell_{2.5} < 36$ of dominators, which is from Lemma 2. We then show that there are at most a constant number of connectors inside the disk also.

For connectors, it either is connected to a dominator node inside the disk or is connected to a dominator node outside the disk, but the distance of that dominator node to u is at most 3.5. From Lemma 2, we know the number of dominators that can connect to a connector inside that disk is at most $\ell_{3.5}$. Notice that there are at most ℓ_3 connectors connected to a dominator node. Thus, there are at most $\ell_3 * \ell_{3.5}$ connectors inside the disk.

Then the total number of links in a path connecting u and v in graph $LDel(ICDS)$ is bounded by the number of dominators and connectors inside that disk, which is at most $\ell_3 * \ell_{3.5} + \ell_{2.5} < 49 * 64 + 36$. Then we know that $LDel(ICDS)$ is a hop-spanner. Notice that although $\ell_3 * \ell_{3.5} + \ell_{2.5}$ is very large here, the bound can be reduced by using more careful analysis. □

Notice that $LDel(ICDS)$ has thickness 2 implies that the average node degree is at most 12. Using the same technique, we can prove the following lemma.

Lemma 8: The node degree of ICDS is bounded by a constant $\ell_3 * \ell_3$.

PROOF. For any dominator node v, it can only connect to connectors, which are introduced by some dominator nodes within 3 hops of v. There are at most ℓ_3 such dominators, each of them can introduce at most ℓ_3 connectors.

For a connector node v, it can connect to both connectors and at most 5 dominators. The connectors are introduced by some dominator nodes within 2 hops of v. There are at most ℓ_2 such dominators, each of them can introduce at most ℓ_3 connectors.

Thus, the node degree in ICDS is bounded by $\ell_3 * \ell_3$ due to $\ell_3 * \ell_3 > \ell_2 * \ell_3 + 5$. □

It immediately implies that the graph $LDel(ICDS)$ has a bounded node degree $\ell_3 * \ell_3$. Notice that this implies that the number of messages sent by the dominator node or connector node is bounded by a constant also.

IV. SIMULATIONS

After building the planar backbone of the networks, we can run *Dominating-Set-Based Routing* [8] on it. When route a message on the planar backbone (such as $LDel(ICDS)$), we can use some other variant routing algorithms, such as Greedy Perimeter Stateless Routing (GPSR) [10], [2]. Because the routing on planar graphs was already studied, we will concentrate on studying the structural properties of the constructed planar backbone $LDel(ICDS)$. We want to study the stretch factors, the maximum and average node degree of the graph, and the communication cost to build these structures.

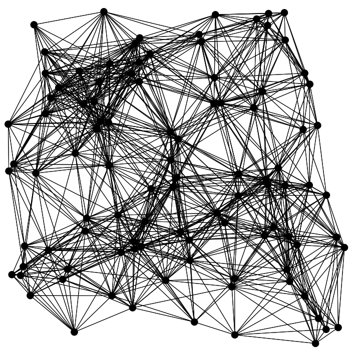


Fig. 6. A Unit Disk Graph.

In our experiments, we randomly generate a set V of n wireless nodes on a $200m$ by $200m$ square, i.e., randomly and uniformly choosing nodes' x-coordinate and y-coordinate values. The transmission radius of all wireless nodes is an experimental parameter. We then generate the $UDG(V)$, and test the connectivity of $UDG(V)$. If it is connected, we construct different topologies from V, such as CDS, CDS', ICDS, ICDS', LDel(ICDS), and so on. Then we measure the stretch factors, degree bound of these topologies, and the communication cost to construct them. Given n, we generate 10 vertex sets V of size n and then generate the graphs for each of these 10 vertex sets. The average and the maximum are computed over all these vertex sets. Figure 7 gives all different topologies defined in this paper for the unit disk graph illustrated by Figure 6. The transmission radius of each node here is set as $35m$.

In following Table I, l_a and l_m are the average and the maximum length stretch factor over all nodes and all graphs respectively; h_a and h_m are the average and the maximum hop stretch factor over all graphs respectively. Additionally, d_a and d_m denote the average and the maximum node degree, e is the average number of edges over all graphs. Here, the maximum node degrees of CDS', ICDS', and LDel(ICDS') are large because the backbone nodes have many links to the dominatee nodes when the graph is dense. As we expected, they are almost equal to the maximum node degree of the unit disk graph. The maximum node degree of the backbone graph such as CDS, ICDS, and LDel(ICDS) does not depend on the node density. Graph LDel(ICDS) has the lowest maximum degree because it removes the some crossing links in other graphs.

We further conduct some experiments to study the relations of the spanning ratios and the communication cost with the node density, the diameter of the original unit disk graph. The

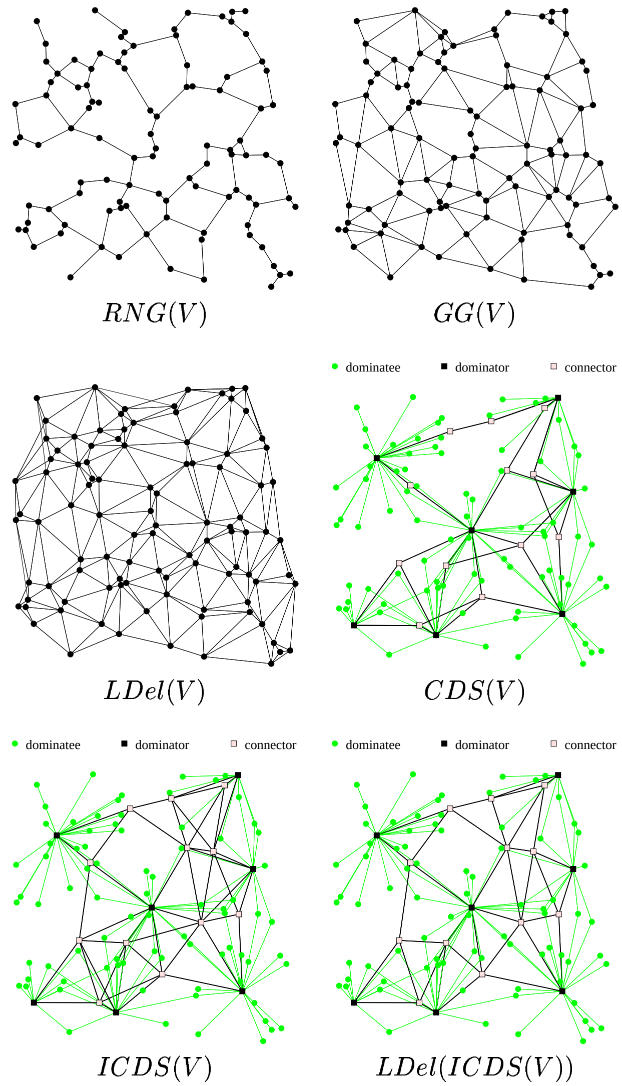


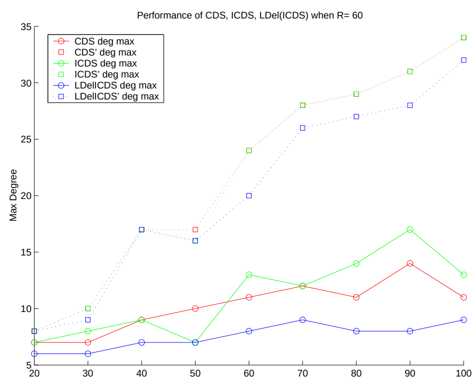
Fig. 7. Different Network topologies.

	d_a	d_m	l_a	l_m	h_a	h_m	e
UDG	21.4	42	-	-	-	-	1069
RNG	2.37	4	1.32	4.49	3.62	16	119
GG	3.56	9	1.12	2.08	2.58	8	178
LDel	5.56	12	1.05	1.44	1.95	5	276
CDS	1.09	16	-	-	-	-	54.4
CDS'	3.34	41	1.27	5.04	1.37	3.5	170
ICDS	1.72	16	-	-	-	-	85.8
ICDS'	4.03	41	1.23	4.17	1.32	3	201
LDel(ICDS)	1.20	9	-	-	-	-	60.0
LDel(ICDS')	3.51	38	1.23	4.20	1.40	4	176

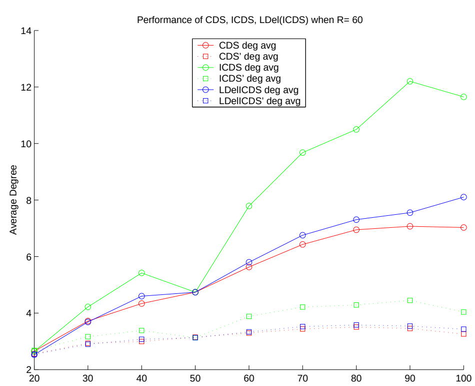
TABLE I

TOPOLOGY QUALITY MEASUREMENTS.

variation of the diameter of the graph is achieved by varying the transmission radius. Figures 8, 9, and 10 illustrate the relations of the node degree, the spanning ratio, and the communication cost with the node density. Here transmission range is always set as $60m$. Remember that we generate nodes in a $200m$ by $200m$ square region. The communication cost is computed based on the number of messages each node needs to send. Here the messages could be IamDominator, IamDominatee, 2HopsPath(u, w, v), 3HopsPath(u, w, x, v), proposal(u, v, w), accept(u, v, w), reject(u, v, w), and so on. We found that the maximum communication cost of each node (around 25 messages) to build CDS, or ICDS is considerablly smaller than our theoretical upper bound. We also found that the difference between the maximum communication cost of each node to build LDel(ICDS') and the communication cost to build CDS is almost fixed. Notice that the difference is actually the cost of building local Delaunay graph on top of the ICDS. This is due to the fact that the maximum degree of the ICDS graph is always bounded by a constant and the communications to build LDel by a node depends on its degree. Each node has to process the proposal messages sent by its neighbors, which implies that the number of accept and reject messages sent by a node are related to its degree.



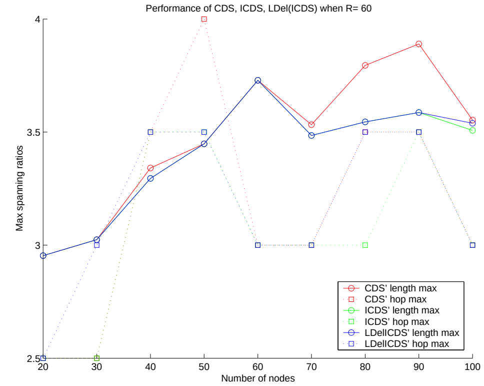
the maximum degree



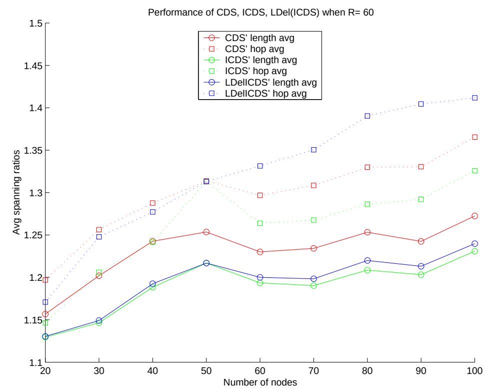
the average degree

Fig. 8. The relation of the graph degree with the node density.

Figures 11, and 12 illustrate the relations of the spanning ratios, and the communication costs with the transmission radius of the node. Here the number of the wireless nodes is fixed as 500.



the maximum spanning ratios



the average spanning ratios

Fig. 9. The relation of the spanning ratios with the node density.

V. Summary and Future Work

In this paper, we present a new algorithm to construct a sparse spanner for network backbone: the local Delaunay triangulation over the connected dominating set graph CDS. A communication efficient distributed algorithm was presented for the construction of a connected dominating set, whose size is guaranteed to be within a constant factor of the minimum. We show that CDS is efficient for both length and hops and has at most $O(n)$ edges while each node has a bounded degree. Then we apply the *localized Delaunay graph (LDel)* on the induced graph ICDS to generate a planar graph without sacrificing the constant hop and length stretch factor properties. We showed that the constructed topology $LDel(ICDS)$ has all the desirable features we listed in Section I. This topology can be constructed locally and is easy to maintain when the nodes move around. The computational complexity of each node is bounded by $O(d \log d)$, where d is the number of 1-hop neighbors. All our algorithms have the message complexity $O(n)$. Moreover, we showed that the number of messages sent by *each* node is bounded by a constant. We also conducted extensive simulations to study the spanning ratios of these structures and the communication cost to construct them when the nodes are randomly placed in a square region. Notice that, recently, Gao, *et al.* also proposed a similar method. However, their algorithms are not communication nor computation efficient. In addition, it is difficult to implement their clustering method.

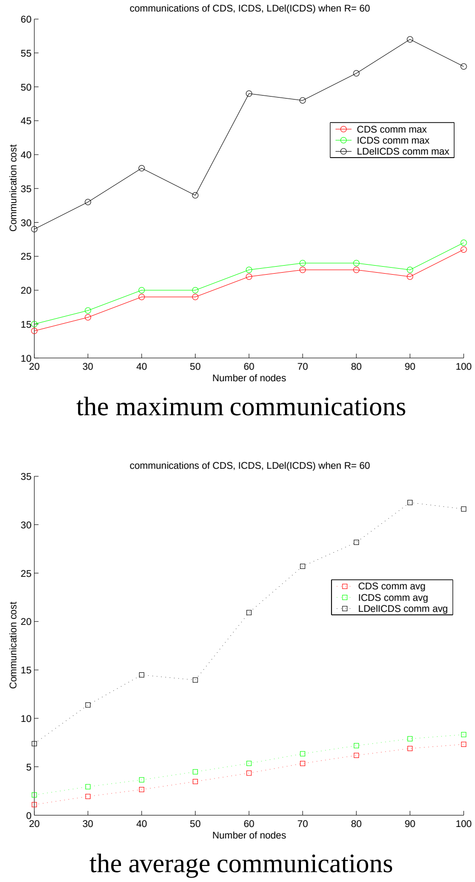


Fig. 10. The relation of the communication cost with the node density.

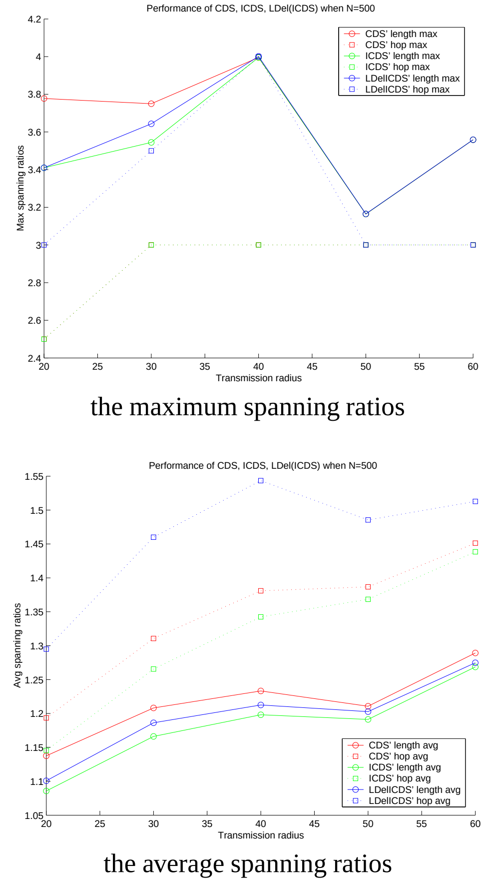


Fig. 11. The relation of the spanning ratios with the transmission radius.

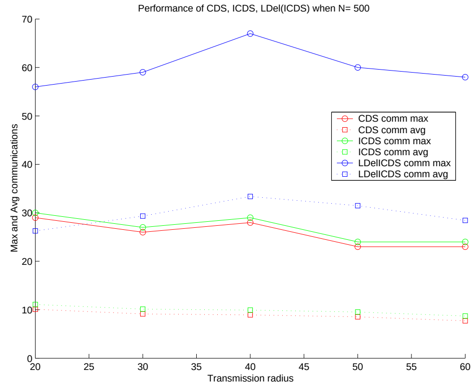
There are many interesting open problems left for further study. Remember that, we use the following assumptions on wireless network model: omni-directional antenna, single transmission received by all nodes within the vicinity of the transmitter, all nodes have the same transmission range, nodes being static for a reasonable period of time. The problem will become much more complicated if we relax some of these assumptions, although some preliminary follow-up works [37], [38], [39] were done recently. Another interesting open problem is to study the dynamic updating of the planar backbone efficiently when nodes are moving in a reasonable speed. Further future work is to lower the constant bounds given in this paper using a more tighter analysis.

VI. Acknowledgment

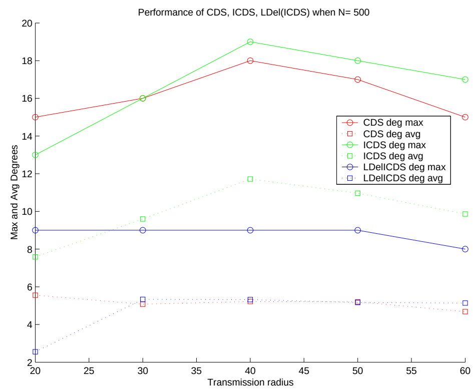
The authors would like to thank K. Alzoubi and P.-J. Wan of Department of Computer Science, Illinois Institute of Technology, for various helpful discussions of their method for forming CDS efficiently.

References

[1] "Ietf mobile ad hoc network working group," http://www.ietf.org/html.charters/manet-charter.html.
[2] 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.
[3] Gruia Călinescu, Ion Măndoiu, Peng-Jun Wan, and Alexander Zelikovsky, "Selecting forwarding neighbors in wireless ad hoc networks," in *ACM DialM*, 2001.
[4] B. Karp and H. T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *ACM/IEEE International Conference on Mobile Computing and Networking*, 2000.
[5] 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.
[6] 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.
[7] 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.
[8] 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.
[9] 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.
[10] 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.
[11] Mahtab Seddigh, Julio Solano Gonzalez, and Ivan Stojmenovic, "Rng and internal node based broadcasting algorithms in wireless one-to-one networks," *ACM Mobile Computing and Communications Review*, vol. 5, no. 2, 2001.
[12] Xiang-Yang Li, Peng-Jun Wan, Yu Wang, and Ophir Frieder, "Sparse power efficient topology for wireless networks," in *IEEE Hawaii Int. Conf. on System Sciences (HICSS)*, 2002.
[13] P. Bose, L. Devroye, W. Evans, and D. Kirkpatrick, "On the spanning ratio of gabriel graphs and beta-skeletons," in *Proceedings of the Latin American Theoretical Infocomatics (LATIN)*, 2002.
[14] J. Gao, L. J. Guibas, J. Hershburger, L. Zhang, and A. Zhu, "Geometric spanner for routing in mobile networks," in *Proceedings of the 2nd ACM*



the communications



the node degree

Fig. 12. The relation of the communication cost and the node degree with the transmission radius.

Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 01), 2001.

[15] 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.

[16] Alan D. Amis, Ravi Prakash, Dung Huynh, and Thai Vuong, "Max-min d-cluster formation in wireless ad hoc networks," in *Proc. of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM*, 2000, vol. 1, pp. 32–41.

[17] Roger Wattenhofer, Li Li, Paramvir Bahl, and Yi-Min Wang, "Distributed topology control for wireless multihop ad-hoc networks," in *IEEE INFOCOM'01*, 2001.

[18] Xiang-Yang Li, Peng-Jun Wan, and Yu Wang, "Power efficient and sparse spanner for wireless ad hoc networks," in *IEEE Int. Conf. on Computer Communications and Networks (ICCCN01)*, 2001, pp. 564–567.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

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

[28] 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.

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

[30] 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.

[31] J.M. Keil and C.A. Gutwin, "The delaunay triangulation closely approximates the complete euclidean graph," in *Proc. 1st Workshop Algorithms Data Structure (LNCS 382)*, 1989.

[32] J. M. Keil and C. A. Gutwin, "Classes of graphs which approximate the complete euclidean graph," *Discr. Comp. Geom.*, vol. 7, pp. 13–28, 1992.

[33] Ji-Cherng Lin, Shi-Nine Yang, and Maw-Sheng Chern, "An efficient distributed algorithm for minimal connected dominating set problem," in *Proc. of the Tenth Annual International Phoenix Conference on Computers and Communications 1991*, 1991, pp. 204–210.

[34] Alan D. Amis and Ravi Prakash, "Load-balancing clusters in wireless ad hoc networks," in *Proc. of the 3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology*, 2000.

[35] Peng-Jun Wan, Khaled M. Alzoubi, and Ophir Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," in *IEEE INFOCOM*, 2002.

[36] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *IEEE Computer*, pp. 57–66, August 2001.

[37] Xiang-Yang Li, "Various intersection graphs for wireless ad hoc networks: Simple heuristics and ptass," 2002, Submitted for publication.

[38] Xiang-Yang Li and Yu Wang, "Intersection graphs from wireless ad hoc networks," 2002, Submitted for publication.

[39] Xiang-Yang Li and Yu Wang, "Simple heuristics and ptass for intersection graphs in wireless ad hoc networks," 2002, Submitted for publication.