Geometric Spanners for Wireless Ad Hoc Networks

Yu Wang Xiang-Yang Li
Department of Computer Science, Illinois Institute of Technology
10 West 31st Street, Chicago, IL 60616, USA
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 combines the connected dominating set and the local Delaunay graph to form the backbone of wireless network. 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 both for hops and length; moreover, we show that, given any two nodes u and v, there is a path connecting them in the backbone such that its length is no more than 6 *times of the shortest path and the number of links is no more than* 3 *times of the shortest path; (4) it can be constructed locally and is easy to maintain when the nodes move around; (5) moreover, we show that 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.*

1 Introduction

We consider a wireless ad hoc network consisting of a set V of n wireless nodes distributed in a two-dimensional plane. Each node has some computation power and 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 edge between two nodes if and only if their Euclidean distance is at most one. The unit disk graph could have $O(n^2)$ edges. Hereafter, we always assume that $UDG(V)$ is a connected graph.

Due to the nodes' limited resource in wireless ad hoc networks, scalability is crucial for network operations. One effective approach is to maintain only $O(n)$ 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 backbone such that an efficient routing is possible.

We 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, or to use a hierarchical structure. When each wireless node knows its geometry position and can quickly retrieve the geometry information about the destination node of a routing request, several localized routing methods based on geometrical forwarding [4, 9] are also proposed to avoid the flooding. Recently, Karp and Kung [9] proposed a new protocol, *Greedy Perimeter Stateless Routing* (GPSR), which guarantees the delivery of the packet if there exists a path. Bose, *et al.* [4] also proposed a similar method. These methods maintain some planar subgraph such as the *relative neighborhood graph* (RNG) or the *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]. Recently, Gao, *et al.* [8] 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 [7] achieves a constant approximation in expectation, the approximation constant is too large for having any practical meaning.

Consequently, we will focus on the construction of a sparse network topology, i.e., a subgraph of $UDG(V)$, which has the following desirable features.

- **Sparseness**. The topology should be a sparse graph, i.e., with $O(n)$ links. This enables numerous algorithms running on this topology more efficient for both time and power consumption.

Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)
1063-6927/02 $17.00 © 2002 **IEEE**

- **Spanner**. We want the subgraph to be a spanner of $UDG(V)$ in terms of both length and hops. Here a subgraph G' is a spanner of a graph G for length if there is a positive real constant t such that for any two nodes, the length of the shortest path in G' is at most t times of the length of the shortest path in G. The constant t is called the *length stretch factor*. In the same way, if we consider the number of hops of paths instead of the length of them, then the subgraph G' is a spanner of a graph G for hops and the constant t is called the hops stretch factor.

- **Bounded degree**. Because every wireless node has limited power and computational resources, we hope the degree of each node in the constructed topology is 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 (no two edges cross each other in the graph). Some routing algorithms ask the topology be planar, such as right hand routing and GPSR [9].

- **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. More importantly, we expect that the time complexity of each node running the algorithm constructing 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.

In [4, 9], two planar subgraphs RNG and GG are used as underlying network topologies. However, Bose, *et al.* [3] proved that the length stretch factors of these two graphs are $O(n)$ and $O(\sqrt{n})$ respectively. Recently, some researchers [11, 17] 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 [11], all these three graphs can not guarantee a bounded node degree (for Yao graph, the node in-degree could be as large as $O(n)$). In [11, 12], 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. However, all these graphs [11, 12, 17] are spanners for length or power, but not for hops; and they are not guaranteed to be planar graphs.

Many researchers had proposed to use the *connected domination set* (CDS) as a virtual backbone (or spine) for routing in wireless as hoc networks [6, 18]. Efficient distributed algorithms for constructing CDSs in ad hoc wireless networks were well studied recently [1, 6, 15, 18]. [1] proposed a method for *minimum connected dominating set* (MCDS) whose time complexity is $O(n)$ and message complexity is $O(n \log n)$. In [7], they proposed a randomized algorithm for maintaining a dynamic dominating set with an $O(1)$ approximation to the optimal solution with very high probability. However, the approximation ratio is quite large.

We present a new algorithm to construct a sparse spanner for network topology. It first constructs a CDS of all wireless nodes using a clustering method similar to [5, 13]. Then a *localized Delaunay triangulations* (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 other 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. We also prove that the total communication cost of our algorithm is $O(n)$, which is within a constant factor of the optimum. Moreover, we show that 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.

The rest of the paper is organized as follows. In Section 2, we give preliminaries needed to describe our new algorithms, and briefly review the literature on related network topology design issues. Section 3 presents our new spanner formation algorithms based on CDS and LDel graphs. In addition, we prove some properties of the new spanner. Section 4 presents the experimental results. We conclude our paper in Section 5 by pointing out some possible future research directions.

2 Geometry Definitions and Notations

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. 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 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.

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 no four vertices of V are co-circular.

However, the main drawback of applying the Delaunay triangulation in the ad hoc wireless environment is that it can not be always constructed locally. In [10], Li, *et al.* defined a new geometry structure, called *k-localized Delaunay graph* (LDel^k), and they showed how to construct it efficiently. All nodes within a constant k hops of a node u in $UDG(V)$ are called the *k-local nodes* or *k-hops neighbors* of u, denoted by $N_k(u)$, which includes u itself. 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*. 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. Notice that, the definition of *k-localized Delaunay graph* (LDel) by Li [10] is different from the definition of RDG by Gao et al. [8]. Let $UDel(V) = Del(V) \cap UDG(V)$. Gao, *et al.* [8] 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.

3 New Spanner Formation Algorithms

3.1 Formation of Backbone

We begin this section by proposing our localized spanner formation algorithms, based on CDS and LDel. The algorithm building CDS has two phases: clustering and finding connectors. Obviously, any algorithm generating a maximum independent set is a clustering method. Various methods can then be used to connect the cluster-heads to form a connected graph. In the rest of section, we will interchange the terms cluster-head and dominator. The node that is not 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*.

3.1.1 Clustering

Many clustering algorithms have been proposed in the literature [1, 2, 5, 13, 16]. All algorithms assume that the nodes have distinctive identities (denoted by ID hereafter). The simplest method uses two messages similar to IamDominator and IamDominatee, and has 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. Here, we assume that each node knows the IDs of all its 1-hop neighbors, which can be achieved by asking each node to broadcast its ID to its 1-hop neighbors initially.

After clustering, one dominator node can be connected to many dominatees. However, it is well-known that any dominatee node can only be connected to at most *five* dominators in UDG model. Generally, we will show that for each node (dominator or dominatee), there are at most a constant number of dominators that are at most k units away.

Lemma 1 *For every node v, the number of dominators with k hops is bounded by a constant ℓ_k.*

PROOF. Because any two dominators are 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 with radius k. Then ℓ_k is bounded by how many disjoint half-unit disks we can put in the disk centered at v with radius $k+0.5$. We can get $\ell_k \leq \frac{\pi(k+0.5)^2}{\pi(0.5)^2}$ using an area argument. The bounds on ℓ_k can be improved by a tighter analysis. □

3.1.2 Finding Connectors

The second step is to find some *connectors* (also called *gateways*) among all the dominatees to connect the dominators. Then the connectors and the dominators form a CDS. Recently, Wan, *et al.* [16] proposed a communication efficient algorithm to find connectors based on Lemma 1 and the following observation.

Lemma 2 *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. $VirtG$ is connected.*

PROOF. We prove it by contradiction. Assume the constructed graph $VirtG$ is not connected. Let C_1, C_2, $\cdots$,C_k be the k connected components. Then any two components cannot have a common dominator as vertex. Let (u, v) be a pair of dominators such that the shortest-hops path $\Pi_{UDG}(u, v)$ connecting them in UDG has the smallest number of hops among all pairs of dominators from any two different connected components. Then path $\Pi_{UDG}(u, v)$ has at least 4 hops from definition. Let x be the middle node on $\Pi_{UDG}(u, v)$. Then node x must be a dominatee. If x is a dominator, then either x and u are from different components or x and v are from different components. Let's say nodes x and u. Then the subpath in $\Pi_{UDG}(u,v)$ connecting x and u has less number of hops than $\Pi_{UDG}(u,v)$. It is a contradiction to the selection of $\Pi_{UDG}(u,v)$.

Let node w be the dominator of x. Then w cannot be at the same component with u and v; otherwise, u and v are from the same component. If w has a different component with v, then $\Pi_{UDG}(x,v)$ concatenated with link wx has less hops than $\Pi_{UDG}(u,v)$, which is a contradiction to the selection of $\Pi_{UDG}(u,v)$. Thus, $VirtG$ is connected. ◻

It is natural to form a CDS by finding connectors to connect any pair of dominators u and v if they are connected in $VirtG$. This strategy is also adapted by Wan, *et al.* [16]. We first briefly review their basic idea of forming a CDS in a distributed manner. 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. If the path $\Pi_{UDG}(u,v)$ has two hops, then u finds the dominatee with the smallest ID to connect u and v. 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.

We then discuss in detail our approach to optimize the communication cost and the memory cost. For example, it is not obvious how node u can find such node w efficiently. In addition, 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 pick any node that comes first to the notice of the node that makes the selection of connectors.

Our method 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;
- 2HopsPath(u,w,v): node u tells its 1-hop neighbors that u has a 2-hops path uwv and w is the unique node selected by u among all intermediate nodes that can connect u and v.
- 3HopsPath(x,u,w,v): node x tells its 1-hop neighbors that x has a 3-hops path $xuwv$ and u and w are the uniquely selected nodes among all intermediate nodes. Node u is selected by node x and node w is selected by node u.

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, we know that 2HopsPath(u,w,v) and 3HopsPath(x,u,w,v) are also broadcasted at most a constant times by each node for all possible dominators v.

To save the memory cost of each wireless node, we design the following link lists for each node u:

- Dominators: it stores all dominators of u if there is any. If u itself is a dominator, no value is assigned for Dominators.
- Connector2HopsPath: for each dominator v that are 2-hops apart from u, u stores (w,v), where the intermediate node w is selected by u to connect u and v.
- Connector3HopsPath: for each dominator v that are 3-hops apart from u, node u stores (w,x,v) such that there is a path $uwxv$, and w is selected by u and x is the node selected by w to connect v.

Notice that for each node, there are at most five dominators. So the size of link list Dominators is at most five. Then from Lemma 1, for each node u, there are at most ℓ_k number of dominators v that are k-hops apart from u. Therefore, the sizes of link lists Connector2HopsPath, Connector3HopsPath are bounded by ℓ_2 and ℓ_3 respectively. Then we are in the position to give the distributed algorithm finding connectors. Assume that a maximal independent set is already constructed by a cluster algorithm.

Algorithm 1 *Finding Connectors*

- Every dominatee w broadcasts to its 1-hop neighbors IamDominatee(w,v) for each v stored at Dominators.
- Assume node u receives a message IamDominatee(w,v) for the first time. If $u \neq v$, v is not in Dominators, and there is no pair $(*,v)$ in Connector2HopsPath, then u adds (w,v) to Connector2HopsPath. Here $*$ denotes any node ID. If u is a dominatee, then it broadcasts a message 2HopsPath(u,w,v) to its 1-hop neighbors. Node u will discard any message IamDominatee($*,v$) afterward.
- When a node w (it must be a dominatee here) receives the message 2HopsPath(u,w,v), w marks itself as a *connector*, if u is a dominator.
- Assume a dominator x receives 2HopsPath(u,w,v), where $x \neq w$. If there is no triple $(*,*,v)$ in Connector3HopsPath, then x adds (u,w,v) to Connector3HopsPath and broadcasts the message 3HopsPath(x,u,w,v) to its 1-hop neighbors.
- When a node u (it must be a dominatee here) receives the message 3HopsPath(x,u,w,v), u marks itself as a connector. Node u sends a message to w asking w to be a connector.

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. This increases the robustness of the backbone. When only one connecting path between any pair of dominators is needed, we can add the following restrictions: a dominator node u stores a 2-hops or 3-hops path connecting it to another dominator node v if and only if node u has a smaller ID than v.

The graph constructed by the above algorithm **Finding Connectors** is called CDS graph (or *backbone* of the network). If we also add all edges that connect all dominatees to their dominators, the graph is called CDS'. The set of dominators and connectors forms a *connected dominating set*. They induce 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 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.

We first show that the number of connectors found is within a constant factor of the minimum.

Lemma 3 *The number of connectors found is at most ℓ_3 times of the minimum.*

PROOF. Let k be the number of dominators found by any algorithm using maximal independent set. Then the minimum number of connectors is $k-1$. From Lemma 1, we know for every node v, the number of dominators within 3 hops is bounded by a constant ℓ_3. Then the number of connectors connecting to v is also bounded by ℓ_3. So totally, the number of connectors is bounded by $\ell_3 k$. ◻

It is also known that the size of the connected dominating set found by the above algorithm is within a small constant factor of the minimum. Let *opt* be the size of the minimum connected dominating set. Then it was shown [14] that the size of the computed maximal independent set has size at most $4 * opt + 1$. We already showed that the size of the connected dominating set found by the above algorithm is at most $\ell_3 k + k$, where k is the size of the maximal independent set found by the clustering algorithm. It implies that the found connected dominating set has size at most $4(\ell_3 + 1) * opt + \ell_3 + 1$. Consequently, the computed connected dominating set is at most $4(\ell_3 + 1)$ factor of the optimum (with an additional constant $\ell_3 + 1$).

3.2 The Properties of Backbone

In this subsection we will study the properties of CDS, CDS', ICDS and ICDS'. We 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* $\max(\ell_3, 5 + \ell_2)$.

PROOF. There are two cases: a dominator node and a connector node. For a dominator w, it can only be connected to connectors v, which must have a dominator u that is 1-hop or 2-hops away from v. From Lemma 1, we know that the number of this kind dominators u is bounded by ℓ_3, so the degree of w is also bounded by ℓ_3. For a connector w, it can be connected to at most 5 dominator nodes and to some connectors. These connectors p should be directly connected to some dominators q, then we know the number of this kind dominators q is bounded by ℓ_2. So the degree of w is bounded by $5 + \ell_2$. Consequently, the node degree is bounded by $\max(\ell_3, 5 + \ell_2) \leq 49$. ◻

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 $5(n-k)$. Although the node degree in CDS is bounded, 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 hops 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 is similar to that presented by Gao, *et al.* [8]. However, we are able to show 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 *CDS' has hops stretch factor at most* 3.

PROOF. Assume the shortest hops 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 1. We construct another path in CDS' from s to t and the number of hops of this path is $O(k)$.

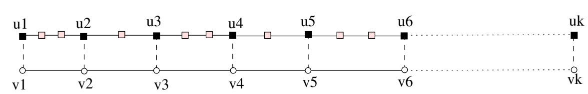

Figure 1. CDS' is a hop-spanner.

For each node v_i in the path $\Pi_{UDG_h}(s,t)$, let u_i be its dominator if v_i is not dominator, else let u_i be v_i itself. Notice that there is a 3-hops 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}. Then 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 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'. Therefore the hops stretch factor of CDS' is bounded by 3 (with an additional constant 2). ◻

Lemma 6 *CDS' has length stretch factor at most* 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, $l(\Pi) \leq h(\Pi)$, because the length of every link is no more than one unit. Lemma 5 implies that $h(\Pi_{CDS'}(s,t)) \leq 3k+2$, where $k = h(\Pi_{UDG_h}(s,t))$. Then $l(\Pi_{CDS'}(s,t)) \leq h(\Pi_{CDS'}(s,t)) \leq 3k+2$. Notice 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. Therefore, $l(\Pi_{UDG_l}(s,t)) > \lfloor h(\Pi_{UDG_l}(s,t))/2 \rfloor$. Notice that $h(\Pi_{UDG_l}(s,t)) \geq h(\Pi_{UDG_h}(s,t)) = k$. So $k \leq 2l(\Pi_{UDG_l}(s,t)) + 2$. Then $l(\Pi_{CDS'}(s,t)) \leq 3k+2 \leq 6l(\Pi_{UDG_l}(s,t)) + 6$. Consequently, CDS' has length stretch factor at most 6 (with an additional constant 6). ◻

As CDS' is a subgraph of ICDS', the hops and length stretch factors of ICDS' are also at most 3 and 6 respectively. The resulting CDS maybe non-planar graph even using some geometry information in our algorithm. A counter example is omitted here due to space limit.

3.3 Local Delaunay Triangulation on ICDS

Several localized routing heuristics were proposed recently for wireless ad hoc networks. Some routing algorithms such as GPSR [9] require the underlying topology to be planar. However, we know that the CDS graph is not guaranteed to be planar, so do its supergraphs CDS', ICDS and ICDS'. Therefore, we cannot directly apply the geometry forwarding based routing algorithms on them. When each node knows its geometry position, we apply the *Localized Delaunay Triangulation* [10] on top of ICDS to make it planar without losing the spanner properties.

3.3.1 Review of Local Delaunay Triangulation

For the completeness of the presentation, we review the algorithms proposed in [10] to construct the local Delaunay triangulation. The algorithm first constructs $LDel^{(1)}(V)$ and then makes it planar efficiently using Algorithm 3.

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 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 $\|uv\| \leq 1$.
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. If angle $\angle wuv \geq \frac{\pi}{3}$, 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 $Del(N_1(u))$ 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 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 $O(n)$. The computation cost of each node is $O(d \log d)$ (from computing the Delaunay triangulation of $N_1(u)$). It was proved in [10] that the above algorithm does generate $LDel^{(1)}(V)$.

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 $O(n)$.

3.3.2 Properties of $LDel(ICDS)$

Applying Algorithm 2 and Algorithm 3 on ICDS, we get a planar graph called $LDel(ICDS)$. Moreover, we can prove that ICDS has a bounded node degree and so does $LDel(ICDS)$. It is proved in [10] that $LDel(G)$ is a spanner if G is a UDG. Notice that ICDS is a UDG defined over all dominators and connectors. Thus, $LDel(ICDS)$ is a spanner in terms of length. We then prove that $LDel(ICDS)$ has a bounded hops stretch factor.

Lemma 7 *$LDel(ICDS)$ has bounded hops stretch factor.*

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 is proved in [10] that the length stretch factor of $LDel(G)$ is at most 2.5 for any UDG G. Therefore, 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. From Lemma 1, there are at most a constant number $\ell_{2.5}$ of dominators inside this disk. We then show that there are at most a constant number of connectors inside the disk also.

Inside the disk, a connector 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 1, 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 $\ell_3 * \ell_{3.5} + \ell_{2.5}$. Then $LDel(ICDS)$ 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 [10], which implies that the average node degree is at most 12. Using the same technique in Lemma 7, we can prove that the node degree of ICDS is bounded by a constant $\ell_3 * \ell_3$. It implies that the graph $LDel(ICDS)$ has a bounded node degree $\ell_3 * \ell_3$ and the number of messages sent by the dominator node or connector node is bounded by a constant also.

4 Simulations

After building the planar backbone of the networks, we can run *Dominating-Set-Based Routing* [18] on it. When route a message on the planar backbone (such as $LDel(ICDS)$), we can use some other variant routing algorithms, such as GPSR [4, 9]. Because the routing on planar graphs was already studied, we will concentrate on studying the structural properties of the constructed planar backbone $LDel(ICDS)$.

In our experiments, we choose a set V of $n = 100$ wireless nodes in a 10×10 square by randomly and uniformly choosing their x-coordinate and y-coordinate values. The transmission range of each node is set as $\sqrt{10}$. Then we generate $UDG(V)$, and test the connectivity of $UDG(V)$. If it is connected, we construct different topologies from V by various algorithms: some are already studied before, some are newly presented in the previous sections. Then we measure the sparseness, the stretch factors, and the degree bound of these topologies. In the experimental results presented here, the average and the maximum are computed over 100 vertex sets. Figure 3 gives all different topologies defined in this paper for a UDG illustrated by Figure 2.

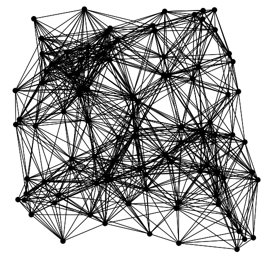

Figure 2. A Unit Disk Graph.

In Table 1, l_a and l_m are the average and the maximum length stretch factor respectively; h_a and h_m are the average and the maximum hops stretch factor respectively. We use d_a and d_m denote the average and the maximum node degree. And e is the average number of edges.

5 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 a connected dominating set graph CDS. We show that each node in CDS has a bounded degree while CDS' has bounded hops and length stretch factors and has at most $O(n)$ edges. Then we apply LDel on the induced graph ICDS to generate a planar graph without sacrificing the constant hops and length stretch factor properties. We showed that the constructed topology $LDel(ICDS)$ has all the desirable features we listed in Section 1. We also conducted experiments to show that this topology is efficient in practice. Notice that, recently, Gao, *et al.* also proposed a similar method. However, their algorithms are not communication and computation efficient. One interesting open problem is to study the dynamic updating of the planar backbone efficiently when nodes are moving.

6 Acknowledgment

The authors would like to thank K. Alzoubi and P.-J. Wan of Dept. of Computer Science, IIT, for various helpful discussions in forming clustering efficiently.

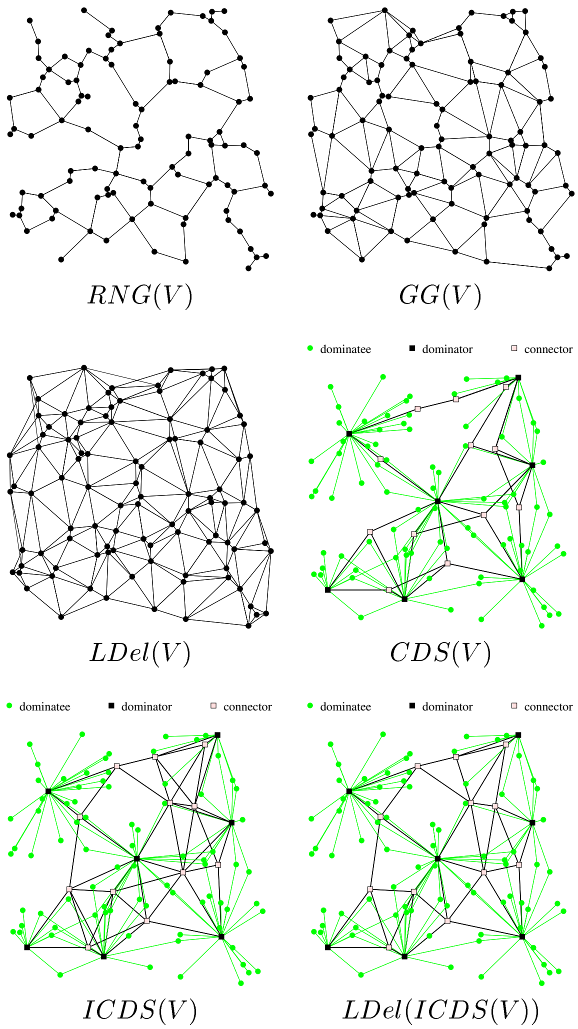

Figure 3. Different Network topologies.

	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 1. Topology quality measurements.

References

[1] K. M. Alzoubi, P.-J. Wan, and O. Frieder. New distributed algorithm for connected dominating set in wireless ad hoc networks. In *Proc. IEEE Hawaii Int. Conf. on System Sciences (HICSS)*, 2002.

[2] A. D. Amis, R. Prakash, D. Huynh, and T. Vuong. Max-min d-cluster formation in wireless ad hoc networks. In *Proc. of IEEE INFOCOM'00*, volume 1, pages 32–41, 2000.

[3] P. Bose, L. Devroye, W. Evans, and D. Kirkpatrick. On the spanning ratio of Gabriel graphs and beta-skeletons. Submitted to *SIAM Journal on Discrete Mathematics*, 2001.

[4] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *3rd Int. Workshop on Discrete Algorithms and methods for mobile computing and communications*, 1999.

[5] I. Chlamtac and A. Farago. A new approach to the design and analysis of peer-to-peer mobile networks, 1999.

[6] B. Das and V. Bharghavan. Routing in ad-hoc networks using minimum connected dominating sets. In *1997 IEEE Int. Conf. on Comm. (ICC'97)*, volume 1, pages 376–380, 1997.

[7] J. Gao, L. J. Guibas, J. Hershburger, L. Zhang, and A. Zhu. Discrete mobile centers. In *Proc. of the 17th ACM Symposium on Computational Geometry*, pages 188–196, 2001.

[8] J. Gao, L. J. Guibas, J. Hershburger, L. Zhang, and A. Zhu. Geometric spanner for routing in mobile networks. In *Proc. of the 2nd ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 01)*, 2001.

[9] B. Karp and H. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proc. of the ACM/IEEE Int. Conf. on Mobile Computing and Networking*, 2000.

[10] X.-Y. Li, G. Călinescu, and P.-J. Wan. Distributed construction of planar spanner and routing for ad hoc wireless networks. In *IEEE INFOCOM*, 2002. To appear.

[11] X.-Y. Li, P.-J. Wan, and Y. Wang. Power efficient and sparse spanner for wireless ad hoc networks. In *IEEE Int. Conf. on Computer Comm. and Networks*, pages 564-567, 2001.

[12] X.-Y. Li, P.-J. Wan, Y. Wang, and O. Frieder. Sparse power efficient topology for wireless networks. In *Proc. IEEE Hawaii Int. Conf. on System Sciences (HICSS)*, 2002.

[13] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal of Selected Areas in Communications*, 15(7):1265–1275, 1997.

[14] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25:59–68, 1995.

[15] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 13, No. 1, January 2002, 14-25.

[16] P.-J. Wan, K. M. Alzoubi, and O. Frieder. Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks. In *IEEE INFOCOM*, 2002. To appear.

[17] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang. Distributed topology control for wireless multihop ad-hoc networks. In *IEEE INFOCOM'01*, 2001.

[18] J. Wu and H. Li. A dominating-set-based routing scheme in ad hoc wireless networks. *The special issue on Wireless Networks in the Telecomm. Systems Journal*, 3:63–84, 2001.

Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)
1063-6927/02 $17.00 © 2002 **IEEE**