Localized Construction of Bounded Degree and Planar Spanner for Wireless Ad Hoc Networks

Yu Wang
Dept. of Computer Science
Illinois Institute of Technology
10 W. 31st Street, Chicago, IL 60616
wangyu1@iit.edu

Xiang-Yang Li
Dept. of Computer Science
Illinois Institute of Technology
10 W. 31st Street, Chicago, IL 60616
xli@cs.iit.edu

ABSTRACT

We propose a novel localized algorithm that constructs a bounded degree and planar spanner for wireless ad hoc networks modeled by unit disk graph (UDG). Every node only has to know its 2-hop neighbors to find the edges in this new structure. Our method applies the Yao structure on the local Delaunay graph [21] in an ordering that are computed locally. This new structure has the following attractive properties: (1) it is a planar graph; (2) its node degree is bounded from above by a positive constant $19 + \lceil \frac{2\pi}{\alpha} \rceil$; (3) it is a t-spanner (given any two nodes u and v, there is a path connecting them in the structure such that its length is no more than $t \leq \max\{\frac{\pi}{2}, \pi \sin\frac{\alpha}{2} + 1\} \cdot C_{del}$ times of the shortest path in UDG); (4) it can be constructed locally and is easy to maintain when the nodes move around; (5) moreover, we show that the total communication cost is $O(n)$, where n is the number of wireless nodes, and the computation cost of each node is at most $O(d \log d)$, where d is its 2-hop neighbors in the original unit disk graph. Here C_{del} is the spanning ratio of the Delaunay triangulation, which is at most $\frac{4\sqrt{3}}{9}\pi$. And the adjustable parameter α satisfies $0 < \alpha < \pi/3$. In addition, experiments are conducted to show this topology is efficient in practice, compared with other well-known topologies used in wireless ad hoc networks.

Previously, only centralized method [5] of constructing bounded degree planar spanner is known, with degree bound 27 and spanning ratio $t \simeq 10.02$. The distributed implementation of their centralized method takes $O(n^2)$ communications in the worst case. No localized methods were known previously for constructing bounded degree planar spanner.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication, Network topology; G.2.2 [**Graph Theory**]: Network problems, Graph algorithms

General Terms

Algorithms, Design, Theory

Keywords

Wireless ad hoc networks, topology control, bounded degree, planar, spanner, localized algorithm.

1. INTRODUCTION

We consider a wireless ad hoc network (or sensor 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 iff their Euclidean distance is at most one. Hereafter, $UDG(V)$ is always assumed to be connected. We also assume that all wireless nodes have distinctive identities and each wireless node knows its position information either through a low-power Global Position System (GPS) receiver or through some other ways. By one-hop broadcasting, each node u can gather the location information of all nodes within its transmission range. Notice, throughout this paper, a *broadcast* by a node u means u sends the message to all nodes within its transmission range. The main communication cost in wireless networks is to send out the signal while the receiving cost of a message is neglected here.

Topology control for wireless ad hoc networks has draw considerable attentions recently [11, 13, 20, 22, 23, 26, 27, 29, 30]. Topology control methods try to maintain a structure that can be used for efficient routing [7, 14, 15] or improve the overall networking performance [11, 20, 27]. Different structures with different properties have been proposed recently in the literature. In this paper, we will focus on the construction of a sparse network topology, i.e., a subgraph of $UDG(V)$, with the following desirable features.

Sparseness. The topology should be a sparse graph, i.e., with $O(n)$ links. This enables numerous algorithms, e.g., routing algorithm based on the shortest path, running on this topology more efficiently for both time and power consumption.

Spanner. We want the subgraph to be a spanner of $UDG(V)$. Here a subgraph G' is a spanner of a graph G 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*. A spanner is always power efficient for unicast routing.

Bounded degree. It is also desirable that the node degree in the constructed topology is small and bounded from above by a constant. A small node degree reduces the MAC-level contention and interference, also may help to mitigate the well known hidden and exposed terminal problems.

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, *Greedy Perimeter Stateless Routing* (GPSR) [15], *Greedy Face Routing* (GFG) [7], *Adaptive Face Routing*(AFR) [18]. and *Gready Other Adaptive Face Routing* (GOAFR) [19].

Efficient Localized Construction. Due to the limited resources and high mobility 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 is at most $O(d \log d)$, where d is the number of 1-hop or 2-hop neighbors.

In [7, 15], two planar subgraphs *relative neighborhood graph* (RNG) and *Gabriel graph* (GG) are used as underlying network topologies. However, Bose, *et al.* [3] proved that the length stretch factors of these two graphs are $\Theta(n)$ and $\Theta(\sqrt{n})$ respectively. They are precisely $n-1$ and $\sqrt{n-1}$ actually [28]. Recently, some researchers [22, 30] proposed to construct the wireless network topology based on Yao graph [31] (also called θ-graph [4]). 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 [22], all these three graphs can not guarantee a bounded node degree (for Yao graph, the node in-degree could be as large as $\Theta(n)$). In [22, 23], 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 [22, 23, 30] are not guaranteed to be planar. In [21] Li, *et al.* proposed a planar spanner *localized Delaunay triangulations* (LDel), and in [10] Gao *et al.* proposed a planar spanner *Restricted Delaunay Graph* for wireless ad hoc networks. However both of them could have unbounded node degree. The structure constructed by Hu [13] may not be a spanner (not to say that the algorithm itself has faults). Previously, no localized methods were known for constructing bounded degree and planar spanner.

Recently Bose *et al.* [5] proposed a centralized $O(n \log n)$-time algorithm that constructs a planar t-spanner for a given nodes set V, for $t = (1+\pi) \cdot C_{del} \simeq 10.02$, such that the node degree is bounded from above by 27. Hereafter, we use C_{del} to denote the spanning ratio of the Delaunay triangulation [9, 16, 17]. As we knew, this algorithm is the first method to compute a planar spanner of bounded degree. However the distributed implementation of this centralized method takes $O(n^2)$ communications in the worst case for a set V of n nodes. Recently, Li and Wang [24] improved this by giving a centralized method that constructs a planar structure with degree bounded by at most $19 + \lceil \frac{2\pi}{\alpha} \rceil$ and the spanning ratio at most $t \le \max\{\frac{\pi}{2}, \pi \sin\frac{\alpha}{2} + 1\} \cdot C_{del}$. Here α is an adjustable parameter satisfying $0 < \alpha < \pi/2$.

In this paper, we propose the first efficient localized algorithm to construct a bounded degree and planar spanner for wireless ad hoc networks. The contributions of this paper include: (i) the node degree of the new planar spanner is bounded by $19 + \lceil \frac{2\pi}{\alpha} \rceil$, (ii) its length stretch factor is $t \le \max\{\frac{\pi}{2}, \pi \sin\frac{\alpha}{2} + 1\} \cdot C_{del}$, where $0 < \alpha < \pi/3$, and (iii) it can be constructed locally using $O(n)$ messages and is easy to maintain when the nodes move around.

The rest of the paper is organized as follows. In Section 2, we propose our centralized method constructing bounded degree planar t-spanner for a unit disk graph. We then give the first localized method, in Section 3, to construct a bounded degree planar t-spanner for $UDG(V)$ with total communication cost $O(n)$ under the broadcasting communication model. In Section 4, experiments are conducted to show the new topology is efficient in practice, comparing to other well-known topologies used in wireless ad hoc networks. Finally, we briefly conclude our paper in Section 5.

2. CENTRALIZED CONSTRUCTION

Our algorithms borrow some ideas from the algorithm by Bose *et al.* [5] which constructs a bounded degree and planar spanner for a given points set V. They show that the length stretch factor of the final graph is $\frac{(\pi+1)2\pi}{(3\cos\pi/6)(1+\epsilon)}$ and node degree is at most 27. The running time of their algorithm is $O(n \log n)$. However, their method is impossible to have a localized even efficient distributed version, since they use BFS and several operations on polygons (such as degree-3 partitions). Notice that breadth-first-search may take $O(n^2)$ communications. In this section, we will give a new method for constructing a planar spanner with bounded node degree for $UDG(V)$, and show that it can be converted to a localized method in Section 3. Our method rigorously combines (localized) Delaunay triangulation and the ordered Yao structure [4, 31].

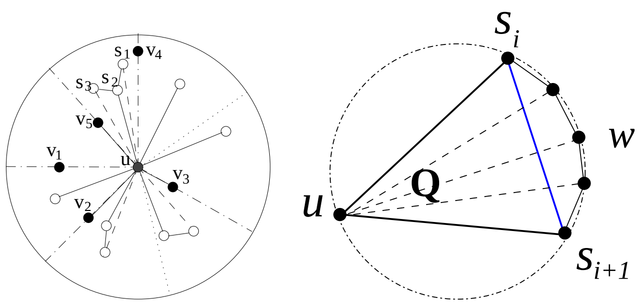
2.1 Centralized Algorithm for UDG

ALGORITHM 1. *Centralized Construction of Planar Spanner with Bounded Degree for* $UDG(V)$

1. First, compute the Delaunay triangulation $Del(V)$ of V.
2. Remove the edges longer than 1 in $Del(V)$. Call the remaining graph unit Delaunay triangulation $UDel(V)$. For every node u, we know its unit Delaunay neighbors $N_{UDel}(u)$ and its node degree $d(u)$ in $UDel(V)$.
3. Find an order π of V as follows: Let $G_1 = UDel(V)$ and $d_G(u)$ be the node degree of u in graph G. Remove the node u with the smallest degree $d_{G_i}(u)$ (smaller ID breaks tie) from graph G_i, and call the remaining graph G_{i+1}. Set $\pi_u = n - i + 1$. Repeat this procedure for $1 \le i \le n$. Let P_v denote the predecessors of v in π, i.e., $P_v = \{u \in V : \pi_u < \pi_v\}$. Since G_i is always a planar graph, the smallest value of $d_{G_i}(u)$ is at most 5. Then, in ordering π, node u at most have 5 edges to its predecessors P_u in $UDel(V)$.
4. Let E be the edge set of $UDel(V)$, E' be the edge set of the desired spanner. Initialize E' to an empty set and mark all nodes in V *unprocessed*. Following the increasing order π, run the following steps to add some edges from E to E' (only consider the unit Delaunay neighbors $N_{UDel}(u)$ of u):
   (a) For the unprocessed node u with the smallest order π_u, let $v_1, v_2, \cdots, v_k$ be the processed neighbors of u in $UDel(V)$ (see Figure 1(a)). Here $k \le 5$. Then k *open* sectors at node u are defined by rays emanated from u to the processed nodes v_i in $UDel(V)$. For each sector centered at u, we divide it into a minimum number of *open* cones of degree at most α, where $\alpha \le \pi/3$ is a parameter.
   (b) For each cone, let $s_1, s_2, \cdots, s_m$ be the geometrically ordered neighbors of u in $N_{UDel}(u)$ in this cone. Notice $s_1, s_2, \cdots, s_m$ are all unprocessed nodes. For each cone, first add the shortest edge us_i in E to E', then add to E' all the edges $s_j s_{j+1}$, $1 \le j < m$. Here such edges $s_j s_{j+1}$ are not necessarily in $UDel(V)$.
   (c) Mark node u processed.

5. Repeat this procedure in the increasing order of π, until all nodes are processed. Let $BPS_1(UDG(V))$ or $BPS_1(V)$ denote the final graph formed by edge set E'.

Notice that in the algorithm we use *open* sectors, which means that in the algorithm we do not consider adding the edges on the boundaries (any edge involved previously processed neighbors). For example, in Figure 1(a), the cones do not include any edges uv_i. This guarantees the algorithm does not add any edges to node v_i after v_i has been processed. This approach, as we will show it later, bounds the node degree.



(a) (b)

Figure 1: (a) Constructing planar spanner with bounded degree for $UDG(V)$: process node u. (b) No new edges can be added by other nodes to intersect $s_i s_{i+1}$, where $s_i s_{i+1}$ is added by u and not in $UDel(V)$.

2.2 Analysis of Algorithm for UDG

In this subsection, we show some nice properties of the generated graph $BPS_1(UDG(V))$ by proving the following three theorems.

THEOREM 1. *The maximum node degree of $BPS_1(UDG(V))$ is at most $19 + \lceil \frac{2\pi}{\alpha} \rceil$.*

PROOF. There are two cases when an edge uv can be added to u in $BPS_1(UDG(V))$. Let us discuss them one by one.

Case 1: When we process node u, edge uv has already been added by some processed node w before. Two subcases here:

Subcase 1.1: The edge uv has been added by a processed node v ($w = v$). For example, in Figure 1(a), node u has edges from v_2, v_3 and v_5 before it is processed. Each predecessor v only adds one such edge to node u.

Subcase 1.2: The edge uv has been added by processed node w (w is not v). Node v is also an unprocessed node when processing w. For example, in Figure 1(a), node s_2 has edges from s_1 and s_3 added by processing node u before node s_2 is processed. Notice that both v and u are neighbors of this processed node w. For each predecessor w, it adds at most two such edges to node u.

Because for each u, it has at most 5 predecessor neighbors (processed neighbors), and each predecessor can add at most 3 edges to it (one edge from Subcase 1.1, or two edges from Subcase 1.2, or both). Thus, the number of edges added by its predecessors before u is processed is bounded by 15.

Case 2: When node u is being processed, we can add one edge uv for each cone. Since we have at most 5 sectors emanated from u and each cone must have angle at most α, it is easy to show that we can have at most $4 + \lceil \frac{2\pi}{\alpha} \rceil$ cones at u. So the number of this kind of edges is also bounded by $4 + \lceil \frac{2\pi}{\alpha} \rceil$.

Notice that after node u is processed, no edges will be added to it. Consequently, the degree of each node u is bounded by $19 + \lceil \frac{2\pi}{\alpha} \rceil$, when the structure is generated by above algorithm. □

For example, when $\alpha = \pi/3$, the maximum node degree is at most 25. Method presented in [5] does not work for $UDG(V)$.

THEOREM 2. *Graph $BPS_1(UDG(V))$ is a planar graph.*

PROOF. When each node u is being processed, we add two kinds of edges: (1) edge us_i, where s_i is the nearest unprocessed node in some cone divided by u; (2) some edges $s_i s_{i+1}$, where s_i and s_{i+1} are consecutive unprocessed neighbors of u in graph $UDel(V)$. Such edge $s_i s_{i+1}$ will be called as *diagonal* hereafter since it must be a diagonal of some polygonal face in $UDel(V)$. See Figure 1(a) for illustration.

Observe that $UDel(V)$ is a planar graph and edges us_i belong to $UDel(V)$. Obviously, the possible intersection in the final structure is caused by at least one edge that does not belong to $UDel(V)$. Then this edge must be a diagonal edge, say $s_i s_{i+1}$. Thus, there are some edges (such as uw in Figure 1(b)) in $Del(V)$ between us_i and us_{i+1} with length longer than 1 (otherwise, $s_i s_{i+1} \in UDel(V)$). Then all such endpoints w of these long edges and s_i, s_j, u will form a polygon, denoted by Q, in $UDel(V)$. All the diagonals of polygon Q intersecting $s_i s_{i+1}$ are longer than 1, as uw is. Thus, they will never be added by our algorithm. This finishes our proof. □

THEOREM 3. *Graph $BPS_1(UDG(V))$ is a t-spanner, where $t = \max\{\frac{\pi}{2}, \pi \sin \frac{\alpha}{2} + 1\} \cdot C_{del}$.*

PROOF. Keil and Gutwin [17] showed that the Delaunay triangulation has spanning ratio at most $C_{del} = \frac{4\sqrt{3}}{9}\pi$ using induction on the increasing order of the lengths of all pair of nodes. We can show that the path connecting nodes u and v constructed in [17] also satisfies that all edges of that path is shorter than $\|uv\|$. Consequently, for any edge $uv \in UDG(V)$ we can find a path in $UDel(V)$ with length at most $C_{del}\|uv\|$, and all edges of the path is shorter than $\|uv\|$. So we only need to show that for any edge $uv \in UDel(V)$, there exists a path in $BPS_1(UDG(V))$ between u and v with length at most $\ell\|uv\|$. Then $BPS_1(UDG(V))$ is a $\ell \cdot C_{del}$-spanner. Then we prove the above claim. Consider an edge uv in $UDel(V)$. If $uv \in BPS_1(UDG(V))$, the claim holds. So assume that $uv \notin BPS_1(UDG(V))$.

Assume w.l.o.g. that $\pi_u < \pi_v$. It follows from the algorithm that, when we process node u, there must exist a node v' in the same cone with v such that $\|uv\| > \|uv'\|$, $uv' \in BPS_1(UDG(V))$, and $\angle v'uv < \alpha \leq \pi/3$. Let $v' = s_1, s_2, \cdots, s_l = v$ be this sequence of nodes in the ordered unprocessed neighborhood of u in $UDel(V)$ from v' to v. Let $v' = w_1, w_2, \cdots, w_k = v$ be the sequence of neighbors of u in $Del(V)$ from v' to v. Obviously, the set $\{s_1, s_2, \cdots, s_l\}$ is a subset of $\{w_1, w_2, \cdots, w_k\}$.

Similar to [5], consider the polygon P, formed by edge uw_1, uw_k and path $w_1 w_2 \cdots w_k$. We will show that the path $w_1 w_2 \cdots w_k$ has length that is at most a small constant factor of the length $\|uv\|$. Let us consider the shortest path from w_1 to w_k that is *totally inside* the polygon P. Let $S(w_1, w_k)$ denote such path. This path consists of diagonals of P and is contained inside $\triangle uw_1w_k$. For example, in Figure 2, $S(w_1, w_k) = w_1 w_7 w_9$.

Assume that $\|uv'\| = x$. Let w be the point on segment uv such that $\|uw\| = \|uv'\|$. Assume that $\|uv\| = y$, then $\|wv\| = y - x$. Notice that node v' is the closest Delaunay neighbor in such cone. Obviously, all Delaunay neighbors w_i in this cone are outside of the sector defined by segments uw and uv'. We will show that such path $S(w_1, w_k)$ is contained inside the triangle $\triangle ww_1w_k$. First, if no Delaunay neighbor is inside $\triangle ww_1w_k$, then $S(w_1, w_k) = w_1 w_k$. Thus, the claim trivially holds. If there are some Delaunay neighbors inside $\triangle ww_1w_k$, then w_1 will connect to the w_i forming the smallest angle $\angle uw_1w_i$. Similarly, node w_k will connect to the

w_j forming the smallest angle $\angle uw_kw_j$. Obviously w_i and w_j are inside $\triangle ww_1w_k$, thus, the shortest path connecting them is also inside $\triangle ww_1w_k$. Since path $S(w_1, w_k)$ is the shortest path inside the polygon P to connect w_1 and w_k, by convexity, the length of $S(w_1, w_k)$ is at most $\|v'w\| + \|wv\| = 2x \sin\frac{\theta}{2} + y - x$. Here $\theta = \angle v'uv < \alpha$.

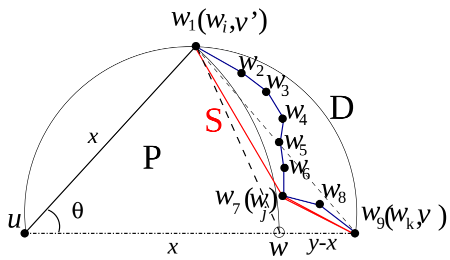


Figure 2: The shortest path in polygon P.

An edge w_iw_j of $S(w_1, w_k)$ has endpoints w_i and w_j in the neighborhood of u. Let $D(w_i, w_j)$ be the sequence of edges between w_i and w_j in the ordered neighborhood of u, which are added by processing u. For example, in Figure 2, $D(w_1, w_7) = w_1w_2w_3w_4w_5w_6w_7$. We can bound the length of $D(w_i, w_j)$ by $\pi/2\|w_iw_j\|$ by the argument in [5, 6]. In [6], it is shown that the length of $D(w_i, w_j)$ is at most $\pi/2$ times $\|w_iw_j\|$, provided that (1) the straight-line segment between w_i and w_j lies outside the Voronoi region induced by u, and (2) that the path lies on one side of the line through w_i and w_j. In other words, we need $D(w_i, w_j)$ to be *one-sided Direct Delaunay path* [1] [9]. In [5], they showed that both these two conditions hold when $\angle w_iuw_j < \pi/2$. This is trivially satisfied since $\angle w_iuw_j < \alpha \leq \pi/2$.

Thus, we have a path $uw_1w_2 \cdots w_k$ to connect u and v with length at most $x + (2x \sin\frac{\theta}{2} + y - x) \cdot \pi/2$, which is at most $y \cdot \max\{\frac{\pi}{2}, \pi \sin\frac{\alpha}{2} + 1\}$ from $x \leq y$.

Since any such node w_i is not inside the polygon Q (defined in the Figure 1(b) of proof for Theorem 2), the path $us_1s_2 \cdots s_k$ (in $BPS_1(UDG(V))$) is not longer than the length of path $uw_1 \cdots w_k$.

Consequently, $BPS_1(UDG(V))$ is a spanner with length stretch factor at most $\max\{\frac{\pi}{2}, \pi \sin\frac{\alpha}{2} + 1\} \cdot C_{del}$. □

For example, when $\alpha = \pi/3$, the spanning ratio is at most $(\frac{\pi}{2} + 1) \cdot C_{del}$; when $\alpha = 2 \arcsin(\frac{1}{2} - \frac{1}{\pi}) \simeq 20.9^o$, the spanning ratio is at most $\frac{\pi}{2} \cdot C_{del}$. We expect to further improve the bound on the spanning ratio by using the following property: all such Delaunay neighbors s_i is inside the circumcircle of the triangle uvv'; see Figure 2.

Notice that we can build Delaunay triangulation in $O(n \log n)$, and do ordering in time $O(n \log n)$ (using heap for the ordering based on degrees), and Yao structure in $O(n)$ (each edge is processed at most a constant times and there are $O(n)$ edges to be processed). Consequently, the time complexity of our algorithm is $O(n \log n)$, same with the method by Bose *et al.* [5]. However, our algorithm has smaller bounded node degree, and (more importantly) our algorithm has potential to become a localized version for wireless ad hoc networks application as we will describe next.

3. LOCALIZED CONSTRUCTION

In [29], Wang *et al.* showed that an algorithm presented in [2] does construct a bounded degree spanner for UDG with $O(n)$ messages (with unit $\log n$ bits) under the broadcast communication model. Li *et al.* [21] presented the first algorithm that constructs a planar spanner using only $O(n)$ messages under the broadcast communication model. No localized method is known before for constructing a planar spanner with bounded node degree.

In this section, we then show how to extend the algorithms presented in previous section to generate bounded degree planar spanner for UDG in a localized manner. Our algorithm is based on the efficient localized construction of a planar spanner $LDel^{(2)}(V)$ for UDG presented by Li *et al.* [21]. For completeness of the presentation, we first review the definitions and give an efficient localized construction of $LDel^{(2)}(V)$ in $O(n)$ total communications.

3.1 Construct $LDel^{(2)}(V)$ Locally

We first introduce some geometric structures and notations to be used in this section. Let $N_k(u)$ be the set of nodes of V that are within k hops distance of u in the unit-disk graph $UDG(V)$. An edge uv is called constrained *Gabriel edge* if $\|uv\| \leq 1$ and the open disk using uv as diameter does not contain any node from V. It is well known [25] that the constrained Gabriel graph is a subgraph of the Delaunay triangulation, more precisely, $GG(V) \subseteq UDel(V)$. Recall that a triangle $\triangle uvw$ belongs to the Delaunay triangulation $Del(V)$ if its circumcircle $disk(u, v, w)$ does not contain any other node of V in its interior. Here we often assume that there are no four nodes of V co-circumcircle. The following definition is one of the key ingredients of the localized algorithm constructing $LDel^{(2)}$.

DEFINITION 1. *A triangle $\triangle uvw$ satisfies k-*localized Delaunay property *if the interior of the circumcircle $disk(u, v, w)$ does not contain any node of V that is a k-hops-neighbor of u, v, or w; 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.

DEFINITION 2. *The k-*localized Delaunay graph *over a node set V, denoted by $LDel^{(k)}(V)$, has exactly all Gabriel edges and edges of all k-localized Delaunay triangles.*

Given a set of points V, the *unit Delaunay triangulation*, denoted by $UDel(V)$, is the graph obtained by removing all edges of the Delaunay triangulation $Del(V)$ that are longer than one unit. It was proved in [10, 21] that $UDel(V)$ is a t-spanner of $UDG(V)$. They [21] proved that graph $UDel(V)$ is a subgraph of the k-localized Delaunay graph $LDel^{(k)}(V)$. Graph $LDel^{(1)}$ is not a planar graph, and $LDel^{(k)}$ is planar for $k > 1$. In [21], Li *et al.* proposed a communication efficient method to construct $LDel^{(1)}$ and then make it planar in total $O(n)$ messages. Here each message has $O(\log n)$ bits. They [21] cannot construct $LDel^{(2)}$ in $O(n)$ messages due to the difficulty of collecting the 2-hop neighbors for every node in $O(n)$ messages. In this paper, we gave the *first* method to construct $LDel^{(2)}$ using $O(n)$ messages.

ALGORITHM 2. *Construct $LDel^{(2)}$ Locally*

1. Every node u collects the location information of $N_2(u)$ based on an efficient method [8] described later. It computes the Delaunay triangulation $Del(N_2(u))$ of its 2-neighbors $N_2(u)$, including u itself.

2. For each edge uv of $Del(N_2(u))$, let $\triangle uvw$ and $\triangle uvz$ be two triangles incident on uv. Edge uv is a Gabriel edge if both angles $\angle uwv$ and $\angle uzv$ are less than $\pi/2$ and $\|uv\| \leq$

[1] For any pair of nodes u and v, let $u = w_1, w_2, \cdots, w_k = v$ be the sequence of nodes whose Voronoi region intersects segment uv and the Voronoi regions at w_i and w_j share a common boundary segment. The the Direct Delaunay path $DT(u, v)$ is $w_1w_2 \cdots w_k$.

2. Node u marks all *Gabriel edges* uv, which will never be deleted.

3. Each node u finds all triangles $\triangle uvw$ from $Del(N_2(u))$ 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 $N_1(u)$ to form a localized Delaunay triangle $\triangle uvw$ in $LDel^{(2)}(V)$, and listens to the messages from its neighboring nodes.

4. When a node u receives a message proposal(u, v, w), u accepts the proposal of constructing $\triangle uvw$ if $\triangle uvw$ belongs to $Del(N_2(u))$ by broadcasting accept(u, v, w) to $N_1(u)$; otherwise, it rejects the proposal by broadcasting reject(u, v, w) to $N_1(u)$.

5. A node u adds the edges uv and uw to its set of incident edges if the triangle $\triangle uvw$ is in $Del(N_2(u))$ and both v and w have sent either accept(u, v, w) or proposal(u, v, w).

We now briefly review the communication efficient method [8] to collect $N_2(u)$ for every node u. Computing the set of 1-hop neighbors with $O(n)$ messages is trivial: every node broadcasts a message announcing its ID. Computing the 2-hop neighborhood is not trivial, as the UDG can be dense. The broadcast nature of the communication in ad hoc wireless networks is however very useful when computing local information. The approach by Calinescu [8] is based on the specific connected dominating set (virtual backbone) introduced by Alzoubi, Wan, and Frieder [1]. This connected dominating set is based on a maximal independent set (MIS). In Calinescu's algorithm, each node uses its adjacent node(s) in the MIS to broadcast its relevant information (its ID and position) over a larger area (constant hops away from the adjacent MIS nodes) on the virtual backbone. Listening to the information about other nodes broadcast by the MIS nodes enables a node to compute its 2-hop neighborhood. The algorithm uses heavily the nodes in the connected dominating set, an example in [8] shows that overloading certain nodes might be unavoidable. The number of messages taken by this method is $O(n)$, which is proved in [8] by using the properties of the specific connected dominating set in [1]. Using the area argument, we can show that the constant in $O(n)$ is at most $C_1 = 3 \times (2 \times 7 + 1)^2 = 675$, since in this method the message from node u can only be re-broadcast by the MIS nodes which are in 7-hops of u and their connectors. The constant can be improved by a tighter analysis.

Finally, we prove the following lemma which will be used in analysis of our new algorithm.

LEMMA 4. *An edge uv is in $LDel^{(2)}(V)$ iff $\|uv\| \leq 1$ and there is a disk passing through u, and v, which does not contain node from $N_2(u) \cup N_2(v)$ inside.*

PROOF. It is trivial that if an edge uv is in $LDel^{(2)}(V)$ then that kind disk exists, since either uv is a Gabriel edge or uv is an edge from a 2-localized Delaunay triangle. Then we prove the other direction.

Assume that there is a disk D_1 passing through u, and v, and there is no node from $N_2(u) \cup N_2(v)$ inside this circle D_1. If uv is a diameter of circle D_1, then it is a Gabriel edge which must be in $LDel^{(2)}(V)$. Otherwise, let D_3 be the disk whose diameter is uv (with center c_3). Disk D_3 must contain some node, say w, inside as shown in Figure 3. Disk D_1 cannot contain w inside. Assume D_1 has center c_1. Let D be a disk centered at some point c on the segment c_1c_3 and passing through u and v. Then we can move the center c of disk D along c_1c_3 from c_1 to c_3 and set the radius of D be $\|cu\|$, until the disk touches the *first* node w from $N_2(u) \cup N_2(v)$. Call resulting disk D_2, which is shown in Figure 3. Since D_2 does not contain any node from $N_2(u) \cup N_2(v)$ inside, we only need show it is empty from $N_2(w)$ to prove that $\triangle uvw$ is a 2-localized Delaunay triangle and thus uv is in $LDel^{(2)}(V)$. We prove this by contradiction.

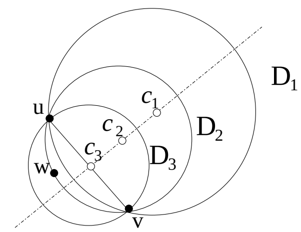
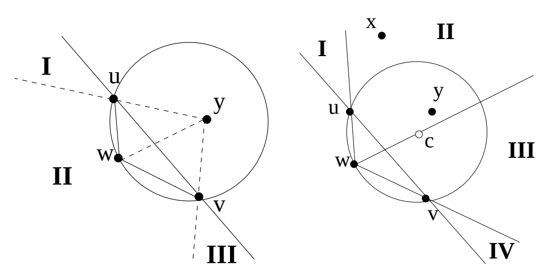


Figure 3: Disk D_2 touches a node w from $N_2(u) \cup N_2(v)$.

Assume that there is a node y from $N_2(w)$ inside $disk(u, v, w)$. Clearly, node y cannot be from $N_2(u) \cup N_2(v)$. Node y must be two hops away from w, otherwise $y \in N_2(u)$. In addition, node y cannot be inside the cap defined by arc uwv since $\|uw\| \leq 1$ and $\|wv\| \leq 1$. Assume that a node x is one hop neighbor of both y and w. Notice that x cannot be one hop neighbor of u or v, otherwise, y will become the two-hop neighbor of u or v, which is a contradiction to the property of disk D_2. Then we know that edges uw, uv, vw, xy and xw are shorter than one unit, while edges uy, vy, wy, xu and xv are longer than one unit. There are two cases about the location of node x: on the different side of uv as y and on the same side of uv as y, as shown in Figure 4. Clearly, node x is outside of the disk D_2, otherwise, D_2 will contain a 2-hop neighbor x of u inside (through path uwx).



(a) Different side (b) Same side

Figure 4: Two cases in the proof: x is on the same side or different side of uv as y.

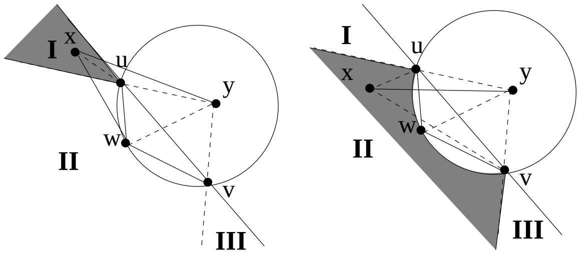
For the first case, we divide the half-space bounded by line uv, which contains w and excludes the cap uwv, into three regions as shown in Figure 4 (a).

If x is inside the region I, see Figure 5 (a) for an illustration. Since $\|xw\| \leq 1$, $\|uw\| \leq 1$, and $\|xu\| > 1$, we have $\angle xwu > \pi/3$. Thus, $\angle xuw < 2\pi/3$. Since $\|xy\| \leq 1$, $\|xu\| > 1$, and $\|uy\| > 1$, we have $\angle yux < \pi/3$. Thus, $\angle wuy = 2\pi - \angle xuw - \angle yux > \pi$, which is impossible.

If x is inside the region II, see Figure 5 (b) for an illustration. Since $\|xu\| > 1$, $\|yu\| > 1$, and $\|xy\| \leq 1$, we have $\angle xuy < \pi/3$. Similarly, we have $\angle uxv < \pi/3$, $\angle xvy < \pi/3$, and $\angle uyv < \pi/3$. Thus, $2\pi = \angle xuy + \angle uxv + \angle xvy + \angle uyv < 4\pi/3$, which is a contradiction.

When x is in region III, the proof is the same as it is in region I.

For the second case, we further divide it into four subcases when



(a) Subcase 1 (b) Subcase 2

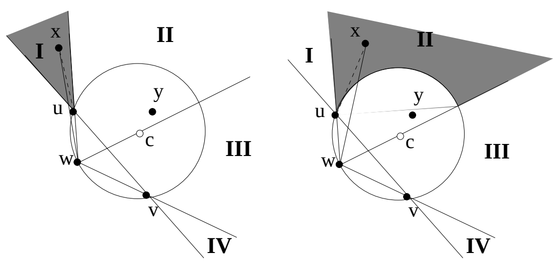
Figure 5: Node x is inside region I or region II.

node x is inside region I, II, III, or IV. Obviously, $\angle uyv + \angle uwv > \pi$ and $\angle uyv < \pi/3$. Thus, $\angle uwv > 2\pi/3$, which implies $\angle uvw < \pi/3$.

If node x is inside the region I, see Figure 6 (a) for an illustration. Since $\angle uwv > 2\pi/3$, we have $\angle wuv < \pi - \angle uwv < \pi/3$. Notice that $\angle wux + \angle wuv > \pi$, so $\angle wux > 2\pi/3$. This implies that $1 \geq \|wx\| > \|ux\| > 1$. It is a contradiction.

If node x is inside the region II, see Figure 6 (b) for an illustration. Here c is the circumcenter of the disk D. Thus, $\angle wux > \pi/2$. This implies that $1 \geq \|wx\| > \|ux\| > 1$. It is a contradiction.

When node x is inside the region III, or IV, the proofs are similar to the cases II, or I respectively.



(a) Subcase 1 (b) Subcase 2

Figure 6: Node x is inside region I or region II.

Then we know the circumcircle $disk(u, v, w)$ of the triangle $\triangle uvw$ does not contain any node from $N_2(u) \cup N_2(v) \cup N_2(w)$ inside. Thus uv is in $LDel^{(2)}(V)$. This finishes the proof. □

3.2 Bound the Degree Locally

In the previous section, we have described a localized algorithm that can construct a planar spanner using $O(n)$ messages. However, some node in structure $LDel^{(2)}(V)$ could have degree as large as $O(n)$. We then give an efficient method to bound the node degree.

ALGORITHM 3. *Localized Construction of Planar Spanner with Bounded Degree for $UDG(V)$*

1. First, compute the planar localized Delaunay triangulation $LDel^{(2)}(V)$, so that every node u knows all its neighbors $N_{LDel^{(2)}}(u)$ and its node degree $d(u)$ in $LDel^{(2)}(V)$. Assume a synchronized method is used to collect $N_{LDel^{(2)}}(u)$ for every node u.

2. Build a local order π of V as follows: (Every node u initializes $\pi_u = 0$, i.e., unordered.)

   (a) If node u has $\pi_u = 0$ and $d(u) \leq 5$, then u queries [2] each node v, from its unordered neighbors, the current degree $d(v)$. If node u has the smallest ID among all unordered neighbors v with $d(v) \leq 5$, node u sets
   $$\pi_u = \max\{\pi_v \mid v \in N_{LDel^{(2)}}(u)\} + 1,$$
   and broadcasts π_u to its neighbors $N_{LDel^{(2)}}(u)$.

   (b) If node u receives a message from its neighbor v saying that $\pi_v = k$, it updates its $d(u) = d(u) - 1$ and also updates the order π_v stored locally. So $d(u)$ represents how many neighbors are not ordered so far.

   If node u finds that $d(u) \leq 5$ and $\pi_u = 0$, it goes to Step 2 (a).

   When node u finds that $d(u) = 0$ and $\pi_u > 0$, it can go to step 3.

3. Build structures based on local order π as follows: (Initialize all nodes unprocessed)

   (a) If a unprocessed node u has the highest local order in its unprocessed neighbors N_u in $LDel^{(2)}(V)$, let k be the number of processed neighbors [3] of u in $LDel^{(2)}(V)$. Node u divides its transmission range into k *open* sectors cut by the rays from u to these processed neighbors. Then divide each sector into a minimum number of *open* cones of degree at most α with $\alpha \leq \pi/3$. For each cone, let $s_1, s_2, \cdots, s_m$ be the ordered unprocessed neighbors of u in $N_{LDel^{(2)}}(u)$. For this cone, node u first adds an edge us_i, where s_i is the nearest neighbor among $s_1, s_2, \cdots, s_m$. Node u then tells $s_1, s_2, \cdots, s_m$ to add all the edges $s_j s_{j+1}$, $1 \leq j < m$. Node u marks itself processed, and tells all nodes in $N_{LDel^{(2)}}(u)$ that it is processed.

   (b) If a unprocessed node v receives a message for adding edge vv' from its neighbor u, it adds edge vv'.

4. When all nodes are processed, the final network topology is denoted by $BPS_2(UDG(V))$ or $BPS_2(V)$.

3.3 Analysis of Localized Algorithm

We first show that the algorithm does process all nodes. First of all, the algorithm cannot stop at stage of ordering nodes locally. This can be shown by contradiction. Assume that some nodes are unordered. The graph formed by these unordered are planar, and thus it contains some nodes with at most 5 unordered neighbors. Among these nodes, the node with the smallest ID will perform step 2 (a), thus reducing the number of unordered nodes consequently.

Notice that the ordering computed by our method is not a total ordering. Some nodes may have the same order. However, no two neighboring nodes in $LDel^{(2)}(V)$ receive the same order. Thus, after all nodes are ordered, the algorithm will process all nodes. Observe that the algorithm does not process two neighboring nodes at the same time. Assume that there are two nodes, say u and v, are processed at the same time. Remember that we process a node only if it has the highest ordering among its unprocessed neighbors.

[2] If some unordered neighbor with $d(v) \leq 5$ has smaller ID, we call such query round a *failed round*. Node u performs a new round of queries only if it finds that the number of its unordered neighbors has been reduced (d(u) has reduced in step 2 (b)). So there are at most 5 rounds of queries.

[3] There are at most 5 processed neighbors since graph $LDel^{(2)}(V)$ is planar.

Thus, nodes u and v must receive the same order, i.e., $\pi_u = \pi_v$, which is impossible in our ordering method.

Additionally, remember that our algorithm checks if $d_u \leq 5$ for computing an ordering locally. Here number 5 can be replaced by any integer larger than 5. Using larger integer may make the algorithm run faster, but on the other hand, it worsens the theoretical bound on the node degree.

It is not difficult to show that the constructed topology is still connected and has bounded node degree. Proofs are similar with $BPS_1(UDG(V))$, which are omitted here due to space limit.

Notice that, the algorithms [5, 24] always add the edges in the Delaunay triangulation to construct a bounded degree planar spanner for a set of points. Thus, the planarity of the final structure is straightforward. The algorithm we proposed in Section 2 may add some edges (such as edges $s_i s_{i+1}$ added in step 4(b) of Algorithm 1) that do not belong to the $UDel(V)$. To prove the planarity of the structure $BPS_1(UDG(V))$, we show that no two added diagonal edges intersect. The property that edges, which possibly intersect $s_i s_{i+1}$ in the centralized algorithm, are all Delaunay edges is crucial in the proof of Theorem 2. This property does not hold anymore in the localized algorithm. We will show that $BPS_2(UDG(V))$ is a planar graph using a different approach.

THEOREM 5. *Graph $BPS_2(UDG(V))$ is a planar graph.*

PROOF. Notice that Algorithm 3 only adds the edges in $LDel^{(2)}(V)$ or edge $s_i s_{i+1}$ such that us_i and us_{i+1} are edges of $LDel^{(2)}(V)$ and s_i, s_{i+1} are consecutive neighbors of u in $LDel^{(2)}(V)$ and $\angle s_i u s_{i+1} < \pi/3$. We call such edge $s_i s_{i+1}$ the *diagonal* edge of the graph $LDel^{(2)}(V)$. Clearly, these diagonal edges cannot intersect any edge from $LDel^{(2)}(V)$. Thus, the only possible intersections in $BPS_2(UDG(V))$ are caused by some diagonal edges. See Figure 7 (a) for an illustration of such two intersected diagonal edges uy and vx. Assume that $\angle uyv < \angle uxv$. Then y is outside of the circumcircle $disk(u, v, x)$ of the triangle $\triangle uvx$.

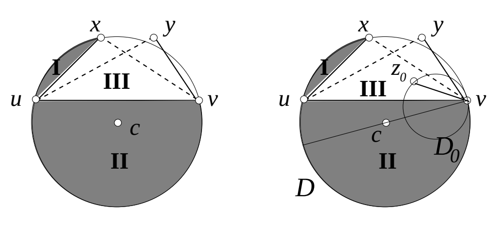


Figure 7: (a) Two diagonal edges uy and vx intersect. (b) z_0 belongs to the sector $\measuredangle uvy$.

If the disk $disk(u, v, x)$ does not contain a node from $N_2(x) \cup N_2(v)$ inside, then edge xv belongs to the graph $LDel^{(2)}(V)$. This is a contradiction to the fact that edges vu and vy are neighboring edges in graph $LDel^{(2)}(V)$. Thus, there must have some node, say z, from $N_2(x) \cup N_2(v)$ inside the disk $disk(u, v, x)$.

If the node z is inside the region II, then z cannot be from $N_2(v)$. Otherwise, we cannot find an empty circle passing through u and v that is free of nodes of $N_2(u) \cup N_2(v)$ inside. This contradicts to the fact edge uv belongs to the graph $LDel^{(2)}(V)$. Thus, node z must be from $N_2(x)$, but not from $N_1(x)$ (otherwise $z \in N_2(v)$ again). Assume that there is a 2-hop path xwz connecting x and z. We then show that $w \notin disk(u, v, x)$. If node w is inside the region I or III, then $\|uw\| \leq 1$. Thus, any circle passing through u and v will contain w or z inside. Since $w \in N_1(u)$ and $z \in N_2(u)$, edge uv cannot belong to graph $LDel^{(2)}(V)$. It is a contradiction. Similarly, if node w is inside the region II, nodes x and w will cause a contradiction to the fact $uv \in LDel^{(2)}(V)$.

Thus node $w \notin disk(u, v, x)$. Then similar to the proof of Lemma 4, we can show that it is impossible to have node $z \in N_2(x)$ in region II. Similarly, region I cannot contain any node from $N_2(u) \cup N_2(x)$. Thus, only region III can possibly contain some node z inside. Then $\|vz\| \leq 1$. This is proved as follows: if z is inside the triangle $\triangle vux$, it is obvious since the three sides of this triangle have length at most 1; if z is inside the cap defined by arc xv, $\|vz\| \leq \|vx\|$ since $\angle vux < \pi/3$.

Let c be the circumcenter of disk $disk(u, v, x)$. Let D be a disk passing through v with center on the segment vc. Clearly, D is inside the disk $disk(u, v, x)$. Among all such disks, we find the largest disk D_0 that is empty of node inside, i.e., the disk that passing through some node z_0, and node v. Then edge vz_0 belongs to graph $LDel^{(2)}(V)$. We then show that z_0 must belong to the sector $\measuredangle uvy$. If z_0 is inside the cap cut by segment vy, then any disk passing through v and y will contain u or z_0 inside since $\angle yuv + \angle yz_0v > \pi$. It contradicts to the existence of edge vy in graph $LDel^{(2)}(V)$. As shown in Figure 7 (b), if z_0 belongs to the sector $\measuredangle uvy$, and $vz_0 \in LDel^{(2)}(V)$, then y and u cannot be consecutive neighbors of v in $LDel^{(2)}(V)$. It is a contradiction. □

THEOREM 6. *Graph $BPS_2(UDG(V))$ is a t-spanner, where $t = \max\{\frac{\pi}{2}, \pi \sin\frac{\alpha}{2} + 1\} \cdot C_{del}$.*

PROOF. We only need to show that for any edge $uv \in UDel(V)$, there exists a path in $BPS_2(UDG(V))$ between u and v with length at most $\ell\|uv\|$. Then $BPS_2(UDG(V))$ is a $\ell \cdot C_{del}$-spanner. We prove the above claim. Consider an edge uv in $UDel(V)$. If $uv \in BPS_2(UDG(V))$, the claim holds. So assume that $uv \notin BPS_2(UDG(V))$.

Assume w.l.o.g. that $\pi_u > \pi_v$. It follows from the algorithm that, when we process node u, there must exist a node x in the same cone with v such that $\|uv\| > \|ux\|$, $ux \in BPS_2(UDG(V))$, and $\angle xuv < \alpha \leq \pi/3$. There is two cases: ux is in $UDel(V)$ or not.

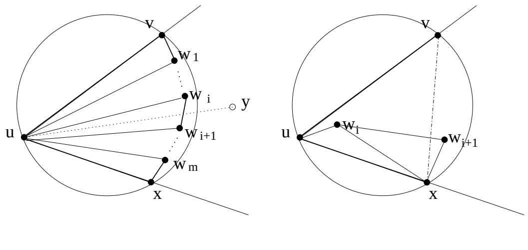
Case 1: $ux \in UDel(V)$. We will show that no edges other than Delaunay edges are added to u between ux and uv. Then we can use the same proof in Theorem 3 to prove that there is a path in $BPS_2(UDG(V))$ to connect u and v with length at most $\max\{\frac{\pi}{2}, \pi \sin\frac{\alpha}{2} + 1\} \cdot \|uv\|$.

Let $w_1, w_2, \cdots, w_m$ be the sequence of Delaunay neighbors of u in $Del(V)$ from v to x. See Figure 8 (a) as illustrations. First all the neighbors w_i should be inside the circumcircle $disk(u, v, x)$ of the triangle $\triangle uvx$, since otherwise any circle passing through u and w_i will contain either x or v inside which is a contradiction with the fact uw_i is Delaunay triangle. Then we prove all the edges $w_i w_{i+1}$ are shorter than one unit.

Remember that $\|uv\| \leq 1$, $\|ux\| \leq 1$ and $\angle xuv \leq \pi/3$, then we have $\|xv\| \leq 1$. If w_i and w_{i+1} are both inside the triangle $\triangle vux$ or the cap cut by segment vx, $\|w_i w_{i+1}\| < 1$. Therefore, the only case that edge $w_i w_{i+1}$ is longer than one unit is shown in Figure 8 (b). Assume that $\|w_i w_{i+1}\| > 1$. Since $\|xw_{i+1}\| < 1$ and $\|xw_i\| < 1$, we have $\angle w_i w_{i+1} x < \pi/2$. Thus, $\angle xuv + \angle w_i w_{i+1} x < \pi/3 + \pi/2 < \pi$. It implies node x is inside the circumcircle $disk(u, w_i, w_{i+1})$. This is a contradiction and finishes the proof of no long edges among all the edges $w_i w_{i+1}$.

Thus, we know all edges $w_i w_{i+1} \in UDel(V)$, in addition, they are also in $LDel^{(2)}(V)$. Therefore we can not have an additional edge uy added to $LDel^{(2)}(V)$ in sector $\measuredangle vux$, since such edge breaks the planar property of $LDel^{(2)}(V)$. See Figure 8 (a) as illustrations.

Case 2: $ux \notin UDel(V)$. Assume ux is added to $LDel^{(2)}(V)$

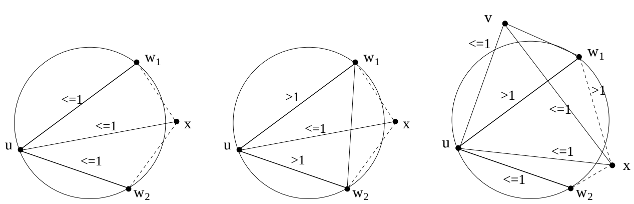


(a) (b)

Figure 8: (a) All the neighbors w_i should be in the circumcircle $disk(u, v, x)$, and no edges other than Delaunay edges are added to u between ux and uv; (b) No edge $w_i w_{i+1}$ can have length longer than one.

in the sector $\measuredangle w_1 u w_2$, where w_1 and w_2 are consecutive Delaunay neighbors of node u. There are three cases for Delaunay edges $w_1 u$ and $w_2 u$. We prove that all of them do not exist by contradiction.

Subcase 2.1: both edges $w_1 u$ and $w_2 u$ are no more than one unit, shown in Figure 9 (a). From the property of Delaunay, x must be outside of the circumcircle $disk(u, w_1, w_2)$ of the triangle $\triangle u w_1 w_2$. Thus, $\angle u w_1 x + \angle u w_2 x > \pi$. Any circle passing though u and x will contain either w_1 or w_2 inside. Notice that $w_1, w_2 \in N_1(u)$. It contradicts to the existence of edge ux in $LDel^{(2)}(V)$.



(a) Subcase 2.1 (b) Subcase 2.2 (c) Subcase 2.3

Figure 9: All subcases in Case 2 do not exist.

Subcase 2.2: both edges $w_1 u$ and $w_2 u$ are longer than one unit, shown in Figure 9 (b). Since $\|uw_1\| > 1 \geq \|ux\|$, $\angle u w_1 x < \pi/2$. Similarly, $\angle u w_2 x < \pi/2$. Then we have $\angle u w_1 x + \angle u w_2 x < \pi$, which is a contradiction with x is outside of the circumcircle $disk(u, w_1, w_2)$.

Subcase 2.3: ux is added to $LDel^{(2)}(V)$ when one of $w_1 u$ and $w_2 u$ is shorter than one unit and the other is longer than one unit. Assume that $\|w_1 u\| > 1$. See Figure 9 (c) as illustrations.

Since edge $ux \in LDel^{(2)}(V)$, we know $\|x w_1\| > 1$. Otherwise w_1 and w_2 are in $N_2(u)$, then any circle passing though u and x will contain either w_1 or w_2 inside. Since $\|uw_1\| > 1$ and $\|ux\| \leq 1$, we have $\angle u w_1 x < \pi/3$. From x is outside the circumcircle $disk(u, w_1, w_2)$, we have $\angle u w_1 x + \angle u w_2 x > \pi$. Thus, $\angle u w_2 x > 2\pi/3$, which implies $\|ux\| > \|uw_2\|$. Therefore, there is no edge from $UDel(V)$ in downside of ux, which selects ux as the shortest neighbor.

Then assume an edge $uv \in UDel(V)$ in upper-side is in the same cone as ux and is longer than ux. Since $\|uv\| \leq 1$, $\|ux\| \leq 1$ and $\angle vux < \pi/3$, we have $\|vx\| \leq 1$. Notice that $w_1 \notin \triangle uvx$ because of $\|uw_1\| > 1$. Again from the property of Delaunay, v and x must be outside of the circumcircle $disk(u, w_1, w_2)$. It implies that $\angle v w_1 x + \angle vux > \pi$. Thus, $\angle v w_1 x > \pi - \angle vux > 2\pi/3$. Then $1 \geq \|vx\| > \|x w_1\| > 1$ causes a contradiction. Therefore Subcase 2.3 shown in Figure 9 (c) does not exist also.

Consequently, it is impossible that any node u will add an edge $ux \notin UDel$ as the shortest link to $BPS_2(UDG(V))$ in a cone that has some edges uv from $UDel$. Together with proof of Case 1, it finishes our proof of spanner property of $BPS_2(UDG(V))$. □

THEOREM 7. *Algorithm 3 uses at most $O(n)$ messages, where each message has $O(\log n)$ bits.*

PROOF. Notice that it was shown in [8] that we can collect the 2-hop neighbor information for all nodes using total $C_1 \cdot n$ messages. Constant C_1 here is at most 675. This constant can be improved by a tighter analysis.

The communication cost of building $LDel^{(2)}$ is $C_2 \cdot n$ since every node only has to propose at most 6 triangles and each propose is replied by two nodes. Constant C_2 here is at most 18.

The second step (local ordering) takes $C_3 \cdot n$ messages, since processing every node u only causes following broadcasts: (1) node u queries at most 5 times, when its $d(u)$ is decreased and $1 \leq d(u) \leq 5$; (2) some nodes v reply u's queries, the total number of this kind of replies is at most $\sum i = 15$ times, where $1 \leq i \leq 5$ and (3) node u claims its new order after it was ordered. Notice that, since node u queries at most $i \leq 5$ unordered nodes in its ith query, only these i nodes reply it in that query round. Constant C_3 here is at most $5 + 15 + 1 = 21$.

The third step (bounded degree) also takes $C_4 \cdot n$ messages, because every node only broadcasts two kind of messages: (1) tells its neighbors to add some edges, and (2) claims that it is processed. The total messages of telling neighbors to add some edges is $12n$ since the total added edges is at most $3n$ from the planar property of the final topology. Notice that each edge uv in the final topology can be added due to at most 4 messages of adding edges (2 from the endpoints u and v, 2 from the two nodes beside the edge uv). Plus the second kind of messages (once per node), the constant C_4 here is at most $12 + 1 = 13$.

Thus, the total communication cost is bounded by $O(n)$ where the constant can be at most $C_1 + C_2 + C_3 + C_4 = C_1 + 18 + 21 + 13 = C_1 + 52 = 675 + 52 = 727$. Here most comes from the slack analysis of collecting $N_2(u)$. □

In addition, it is easy to show that the computation cost of each node is at most $O(d_2 \log d_2)$, where d_2 is the number of its 2-hop neighbors in the original unit disk graph. This can be improved to $O(d_1 \log d_1 + d_2)$, where d_1 is the number of its 1-hop neighbors in the original unit disk graph. The improvement is based on the fact that we only need the triangles $\triangle wuv$ in $LDel^{(2)}(V)$ that has angle $\angle wuv \geq \pi/3$. All such triangles are definitely in $LDel^{(1)}(V)$. Thus, we can construct the Delaunay triangulation $Del(N_1(u))$ of $N_1(u)$ in the first step of Algorithm 2. Then check the candidate triangles to see if they contain any node from $N_2(u)$ inside its circumcircle. If it does not, then it belongs to $Del(N_2(u))$ also.

Observe that, after each node u collects the 2-hop neighbors $N_2(u)$ (Step 1 of Algorithm 2), our algorithms can be performed asynchronously. However, collecting $N_2(u)$ need synchronized communication since otherwise, a node cannot determine if it indeed already collected $N_2(u)$.

4. EXPERIMENTS

In this section we measure the performance of the new bounded degree and planar spanner by conducting some experiments. In our experiments, we randomly generate a set V of n wireless nodes and its $UDG(V)$, and test the connectivity of $UDG(V)$. If it is connected, we construct different localized topologies from V, including our new topologies ($BPS_1(V)$ and $BPS_2(V)$), some well-known planar topologies (*Gabriel graph $GG(V)$*, *relative neighborhood graph $RNG(V)$* and *localized Delaunay triangulations*

$LDel(V)$), and some bounded degree spanners (*Yao graph* $YG(V)$ and *Yao and Sink* $YG^*(V)$). Then we measure the sparseness, the power efficiency and the communication cost of these topologies. In the experimental results presented here, we generate 50 random wireless nodes in a 10×10 square; the number of cones is set to 8 when we construct $YG(V)$ and $YG^*(V)$; the angle parameter $\alpha = \pi/3$ when we construct $BPS_1(V)$ and $BPS_2(V)$; the transmission range is set as 3. We generate 100 vertex sets V (each with 50 vertices) and then generate the graphs for each of these 100 vertex sets. The average and the maximum are computed over all these 100 vertex sets. Figure 10 gives all seven different topologies for the unit disk graph illustrated by the first figure of Figure 10. It shows that all these topologies except $YG(V)$ and $YG^*(V)$ are planar.

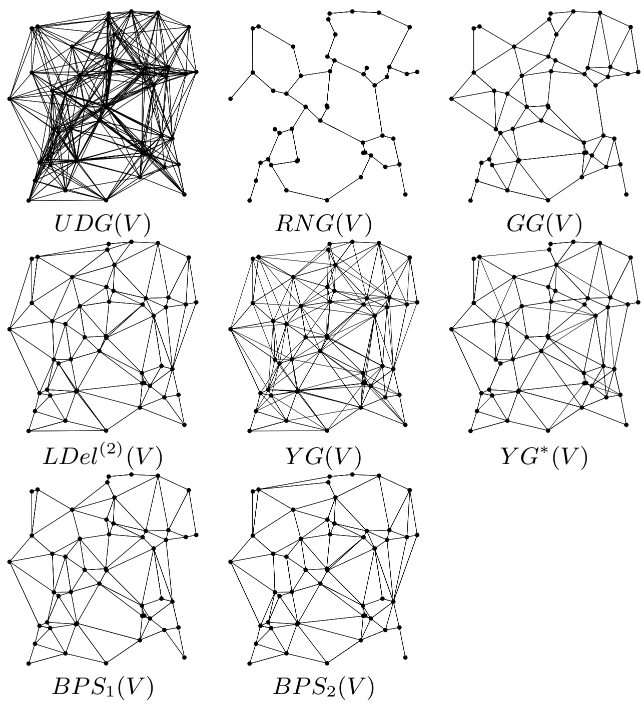


Figure 10: Different topologies from the same $UDG(V)$.

4.1 Node Degree

The node degree of the wireless networks should not be too large. Otherwise a node with a large degree has to communicate with many nodes directly. This increases the interference and the overhead at this node. The node degree should neither be too small: a small node degree usually implies that the network has a lower fault tolerance and it also tends to increase the overall network power consumption as longer paths may have to be taken. Thus, the node degree is an important performance metric for the wireless network topology. The node degrees of each topology are shown in Table 1. Here d_{avg}/d_{max} is the average/maximum node degree. It shows that $BPS_1(V)$ and $BPS_2(V)$ have much less number of edges (average node degrees) than $LDel(V)$, $YG(V)$ and $YG^*(V)$. In other words, these graphs are sparser, which is also verified by Figure 10. Recall that theoretically, only $YG^*(V)$, $BPS_1(V)$ and $BPS_2(V)$ have bounded node degree (both for in-degree and out-degree). In [22, 23], Li *et al.* gave an example to show that $RNG(V)$, $GG(V)$, $YG(V)$ and $LDel(V)$ could have large node degree (in-degree for $YG(V)$). Notice that in our experiments since the wireless nodes are randomly distributed in 2-d space, the maximum node degree of these graphs are not as big as the example. It is proved that node degree of $YG^*(V)$ is bounded from above by $(k+1)^2 - 1$ (in-degree is at most $k(k+1)$, out-degree is at most k), where $k = 8$ is the number of cones. In this paper, we prove that $BPS_1(V)$ and $BPS_2(V)$ have bounded node which is at most $19 + \lceil \frac{2\pi}{\alpha} \rceil = 25$ when $\alpha = \pi/3$. All these theoretical bounds of node degree can be shown from the maximum node degrees in Table 1.

Table 1: Node degrees & stretch factors of different topologies.

	d_{avg}	d_{max}	t_{avg}	t_{max}	ρ_{avg}	ρ_{max}
UDG	16.84	35	1.000	1.000	1.000	1.000
RNG	2.28	5	1.316	4.141	1.057	2.932
GG	3.36	8	1.119	2.024	1.000	1.000
$LDel$	5.25	11	1.048	1.406	1.000	1.000
YG	8.10	19	1.041	1.721	1.002	1.445
YG^*	4.81	11	1.070	1.990	1.003	1.459
BPS_1	4.43	9	1.074	1.841	1.004	1.678
BPS_2	4.46	10	1.072	1.841	1.003	1.663

4.2 Spanner Properties

Besides bounded node degree, the most important design metric of wireless networks is perhaps the power efficiency, as it directly affects both the node and the network lifetime. So while our new topologies increase the sparseness, how does it affect the power efficiency of the constructed network? We then define *power stretch factor* for measuring the power efficiency. A subgraph G' is a power spanner of a Graph G if there is a positive real constant ρ such that for any two nodes u and v, the minimum power consumed by all paths between u and v in G' is at most ρ times of the minimum power consumed by all paths between them in G. The constant ρ is called the power stretch factor. Here we assume the total transmission power consumed by path $v_0, v_1, ..., v_k$ is $\sum_{i=1}^{k} ||v_{i-1}v_i||^{\beta}$, where the power attenuation constant β is a real constant depended on the wireless environment. In our simulations $\beta = 2$. Table 1 also summarizes our experimental results of the length and power stretch factors of all these topologies. Here, t_{avg}/t_{max} is the average/maximum length stretch factor; ρ_{avg}/ρ_{max} is the average/maximum power stretch factor. It is not surprise that the average/maximum power stretch factors of $BPS_1(V)$ and $BPS_2(V)$ are at the same level of those of the $YG(V)$ and $YG^*(V)$ while they are planar and much sparser.

4.3 Communication Cost

In Section 3 we proved that the localized algorithm constructing $BSP_2(V)$ uses at most $O(n)$ messages. We found that when the number of wireless nodes increases the average messages used by each node for constructing $BPS_2(V)$ is still in the same level. In this experiment, we generate from 50 to 300 random wireless nodes in a 10×10 square and run our localized algorithm to build $BSP_2(V)$. The average and the maximum are computed over 20 vertex sets. All other parameters and settings are same with previous experiments. Table 2 summarizes our experimental results of the node degree, length and power stretch factors, and communication costs of $BPS_2(V)$. Here, $d_{avg}(UDG)/d_{max}(UDG)$ is the average/maximum node degree for the original unit disk graph; $tot\_msg_{avg}/tot\_msg_{max}$ is the average/maximum total messages cost for constructing $BPS_2(V)$; $nod\_msg_{avg}/nod\_msg_{max}$ is the average/maximum messages cost in each node during the construction. Notice that here we do not count the messages used in building $LDel^{(2)}(V)$, since in [21] it was proved that the communication cost of building $LDel^{(2)}(V)$ is $O(n)$. In other words, we only consider the messages used in the second and third steps of Algorithm 3. The first two rows of Table 2 show the network becomes more and more dense while the number of wireless nodes increases. Experimental results of communication costs on each node show that the localized method does not cost more messages on each node even the graph becomes more dense. Simulations in Table 2 also show that the performances of our new topology $BPS_2(V)$ are stable when number of nodes changes.

Table 2: Performances and communication costs of $BPS_2(V)$.

num_of_nodes	50	100	150	200	250	300
$d_{avg}(UDG)$	16.52	34.99	51.81	68.15	85.87	103.85
$d_{max}(UDG)$	29	62	94	114	140	175
d_{avg}	4.19	4.39	4.54	4.60	4.58	4.63
d_{max}	8	9	11	11	9	9
t_{avg}	1.094	1.101	1.100	1.098	1.099	1.096
t_{max}	1.958	1.968	1.949	1.978	1.995	1.977
ρ_{avg}	1.017	1.012	1.012	1.009	1.009	1.010
ρ_{max}	1.918	1.937	1.900	1.932	1.916	1.937
tot_msg_{avg}	393	812	1229	1655	2076	2498
tot_msg_{max}	398	821	1244	1670	2090	2512
nod_msg_{avg}	7.86	8.12	8.19	8.27	8.30	8.32
nod_msg_{max}	12	13	15	14	16	14

5. CONCLUSION

In this paper, we proposed both centralized and localized algorithms to construct planar spanners with bounded node degree for wireless ad hoc networks. The centralized algorithm can be implemented in time $O(n \log n)$. The localized algorithm can be implemented using $O(n)$ messages under the broadcast communication model for wireless networks. The basic idea of this new method is to use (localized) Delaunay triangulation to make planar spanner graph, then apply some ordered Yao graph to bound the node degree. It is carefully designed to not lose all good properties when combining them. As we know, this is the first localized algorithm to construct bounded degree and planar spanner. We also conducted experiments to show this topology is efficient in practice compared with other well-known topologies for wireless ad hoc networks.

Centralized algorithm can also be extended to bound the total edge length to be within a constant factor of Euclidean minimum spanning tree, see [24]. It is open how to bound the total edge length of $BPS_2(UDG(V))$ in a localized manner.

6. REFERENCES

[1] K. Alzoubi, P.-J. Wan, and O. Frieder. Message-optimal connected-dominating-set construction for routing in mobile ad hoc networks. In *ACM MobiHoc'02*, 2002.

[2] S. Arya, G. Das, D. Mount, J. Salowe, and M. Smid. Euclidean spanners: short, thin, and lanky. In *Proc. 27th ACM STOC*, pages 489–498, 1995.

[3] 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.

[4] P. Bose, J. Gudmundsson, and P. Morin. Ordered theta graphs. In *Proc. of Canadian Conf. on Computational Geometry (CCCG)*, 2002.

[5] P. Bose, J. Gudmundsson, and M. Smid. Constructing plane spanners of bounded degree and low weight. In *Proceedings of the European Symposium on Algorithms (ESA)*, 2002.

[6] P. Bose and P. Morin. Online routing in triangulations. In *Proc. of the 10th Int. Symp. on Algorithms and Computation ISAAC*, 1999.

[7] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *ACM/Kluwer Wireless Networks*, 7(6), pages 609–616, 2001. In *ACM DIALM 99*, pages 48–55, 1999.

[8] G. Călinescu. Computing 2-hop neighborhoods in ad hoc wireless networks. In *AdHoc-Now'03*, 2003.

[9] D. Dobkin, S. Friedman, and K. Supowit. Delaunay graphs are almost as good as complete graphs. *Discrete Computational Geometry*, 1990.

[10] 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 MobiHoc*, 2001.

[11] M. Grünewald, T. Lukovszki, C. Schindelhauer, and K. Volbert. Distributed maintenance of resource efficient wireless network topologies, 2002. Proc. of the 8th European Conference on Parallel Computing (Euro-Par'02).

[12] J. Gudmundsson, C. Levcopoulos, and G. Narasimhan. Improved greedy algorithms for constructing sparse geometric spanners. In *Scandinavian Workshop on Algorithm Theory*, pages 314–327, 2000.

[13] L. Hu. Topology control for multihop packet radio networks. *IEEE Transactions on Communications*, 41, 1993.

[14] L. Jia, R. Rajaraman, and C. Scheideler. On local algorithms for topology control and routing in ad hoc networks, 2002. Submitted for publication.

[15] B. Karp and H. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proc. of the ACM MobiCom*, 2000.

[16] J. Keil and C. Gutwin. The delaunay triangulation closely approximates the complete Euclidean graph. In *Proc. 1st Workshop Algorithms Data Structure (LNCS 382)*, 1989.

[17] J. Keil and C. Gutwin. Classes of graphs which approximate the complete euclidean graph. *Discrete Computational Geometry*, 7, 1992.

[18] F. Kuhn, R. Wattenhofer, and A. Zollinger. Asymptotically Optimal Geometric Mobile Ad-Hoc Routing. In *ACM DialM*, 2002.

[19] F. Kuhn, R. Wattenhofer, and A. Zollinger. Worst-case optimal and average-case efficient geometric ad-hoc routing. In *ACM MobiHoc*, 2003.

[20] L. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer. Analysis of a cone-based distributed topology control algorithms for wireless multi-hop networks. In *ACM PODC*, 2001.

[21] X.-Y. Li, G. Calinescu, P.-J. Wan, and Y. Wang. Localized delaunay triangulation with application in ad hoc wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, to appear, 2003. Short version was in INFOCOM 2002.

[22] X.-Y. Li, P.-J. Wan, and Y. Wang. Power efficient and sparse spanner for wireless ad hoc networks. In *IEEE ICCCN*, pages 564–567, 2001.

[23] X.-Y. Li, P.-J. Wan, Y. Wang, and O. Frieder. Sparse power efficient topology for wireless networks. In *IEEE 35th HICSS*, 2002.

[24] X.-Y. Li and Y. Wang. Efficient construction of low weight bounded degree planar spanner. In *9th COCOON*, 2003.

[25] F. P. Preparata and M. I. Shamos. *Computational Geometry: an Introduction*. Springer-Verlag, 1985.

[26] R. Rajaraman. Topology control and routing in ad hoc networks: A survey. *In SIGACT News*, 33:60–73, 2002.

[27] R. Ramanathan and R. Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In *IEEE INFOCOM*, 2000.

[28] W. Wang, X.-Y. Li, K. Moaveninejad, Y. Wang, and W.-Z. Song. The Spanning Ratios of $beta$-Skeleton. *Canadian Conference on Computational Geometry (CCCG)*, 2003.

[29] Y. Wang, X.-Y. Li, and O. Frieder. Distributed spanner with bounded degree for wireless networks. *International Journal of Foundations of Computer Science*, 14:183–200, 2003.

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

[31] A. C.-C. Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM J. Computing*, 11:721–736, 1982.