Mobile Networks and Applications 11, 161–175, 2006

DOI: 10.1007/s11036-006-4469-5

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

YU WANG
Department of Computer Science, University of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, NC 28223, USA

XIANG-YANG LI
Department of Computer Science, Illinois Institute of Technology, 10 West 31st Street, Chicago, IL 60616, USA

Published online: 31 March 2006

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 [1] 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 the unit disk graph); (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 $O(n \log n)$ bits, 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 $0 < \alpha \leq \pi/3$.

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 because 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. 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 a localization service. By one-hop broadcasting, each node u can gather the location information of all nodes within the transmission range of u. Notice, throughout this paper, a *broadcast* by a node u means u sends the message to all nodes within its transmission range. Remember that, in wireless ad hoc networks, the radio signal sent out by a node u can be received by all nodes within the transmission range of u.

Unlike wired networks, in wireless ad hoc networks, each node can move and thus change the topology of the network. In this case, we need to adjust the transmission power to keep some properties of the network topology such as connectivity or power efficiency. The lifetime of a wireless network, which depends on battery power, is usually restricted because of limited capacity and resources on each node. Thus a main goal of topology control is to increase the longevity of such networks which can be obtained by designing power efficient algorithms [3–8].

One effective approach [4–6,8–14] is to maintain only a linear number of links using a *localized* construction method. In other words, we construct a sparse distributed structure as network topology for the wireless network. However, this sparseness of the constructed network topology should not compromise too much on the power consumptions of communications. So we hope that in the sparse topology every shortest route in the constructed network topology is efficient. Here a route is *efficient* if its length is no more than a constant factor of the least length needed to connect the source and the destination. A trade-off can be made between the sparseness of the topology and the efficiency. Obviously, not all sparse subgraphs are good candidates for the underlying network topologies.

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

- ***Connectivity***. Connectivity is the most basic feature of the network topology. It guarantees that there exists at least one path from one node to any other nodes. Notice that here we require that the subgraph of $UDG(V)$ is connected if $UDG(V)$ is connected.

- ***Sparseness***. The topology should be a sparse graph, i.e., with $O(n)$ links. This makes numerous algorithms, e.g., routing algorithm based on the shortest path, running on this topology more efficient 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 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, and also may help to mitigate the well known hidden and exposed terminal problems. In addition, a structure with small degree will improve the overall throughput [15].
- ***Planar***. The topology is a planar graph (no two edges crossing each other in the graph). Some routing algorithms require the topology to be planar, such as right hand routing, *Greedy Perimeter Stateless Routing* (GPSR) [16], *Greedy Face Routing* (GFG) [17], *Adaptive Face Routing*(AFR) [18]. and *Gready Other Adaptive Face Routing* (GOAFR) [19].
- ***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 or 2-hop neighbors.

In [16,17], two planar subgraphs *relative neighborhood graph* (RNG) and *Gabriel graph* (GG) are used as underlying network topologies. However, Bose et al. [20] 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 [31]. Recently, some researchers [8,12] proposed to construct the wireless network topology based on the Yao graph [28] (also called θ-graph [35]). 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 [12], 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 [12,13], 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 [8,12,13] are not guaranteed to be planar. Li et al. [1] proposed a planar spanner *localized Delaunay triangulations* (LDel), and Gao et al. [21] proposed a planar spanner *Restricted Delaunay Graph* for wireless ad hoc networks. However both of them can have unbounded node degree. The planar structure constructed by Hu [22] may not be a spanner. Previously, no localized methods were known for constructing a bounded degree and planar spanner.

Recently Bose et al. [2] proposed a centralized $O(n \log n)$-time algorithm that constructs a planar t-spanner for a given node 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 [23–25]. As far as we know, their algorithm is the first method to compute a planar spanner of bounded degree. However the distributed implementation of their centralized method takes $O(n^2)$ communications in the worst case for a set V of n nodes. Recently, Li and Wang [26] 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 a spanning ratio of at most $t \leq \max\{\frac{\pi}{2}, \pi \sin\frac{\alpha}{2} + 1\} \cdot C_{del}$. Here α is an adjustable parameter satisfying $0 < \alpha \leq \pi/2$.

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 $t \leq \max\{\frac{\pi}{2}, \pi \sin\frac{\alpha}{2} + 1\} \cdot C_{del}$, where $0 < \alpha \leq \pi/3$, and (iii) it can be constructed locally using $O(n)$ messages (each message with $O(\log n)$ bits) and is easy to maintain when the nodes move around.

The rest of the paper is organized as follows. In Section 2, we review the 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. Prior Art: Centralized Construction for UDG

Our localized algorithm is developed based on the centralized algorithm developed in [26], which constructs a planar spanner with bounded node degree for $UDG(V)$. The basic idea of the centralized method is to combine Delaunay triangulation and the ordered Yao structure [28]. Our localized method is significantly different from this centralized method: our method uses a novel combination of the Yao structure and the local Delaunay graph. For completeness of presentation, we review the centralized method (shown in Algorithm 1) here. We assume that every node u has a unique ID, denoted by $ID(u)$.

Algorithm 1: Centralized construction of planar spanner with bounded degree

(1) First, compute the Delaunay triangulation $Del(V)$ of the set V of n wireless nodes.
(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 $i = 1$, $G_1 = UDel(V)$ and $d_G(u)$ be the node degree of u in graph G. Remove the node u with the smallest degree $d_{Gi}(u)$ (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 $1 \leq i \leq n$. Let P_v denote the predecessors of v in π, i.e., $P_v = \{u \in V\colon \pi_u < \pi_v\}$. Since G_i is always a planar graph, the smallest value of $d_{Gi}(u)$ is at most 5. Then, in order π, node u has at most 5 edges to its predecessors P_u.
(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 $v_1, v_2, \ldots, v_k$ be the processed neighbors of u in $UDel(V)$ (see Figure 1). Here $k \leq 5$. Then k *open* sectors centered 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 $\alpha \leq \pi/3$ is a parameter.
   (b) For each cone, let $s_1, s_2, \ldots, s_m$ be the geometrically ordered neighbors of u in $N_{UDel}(u)$ in this cone. Notice $s_1, s_2, \ldots, s_m$ 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 $s_j s_{j+1}$, $1 \leq j < m$. Notice that here such edges $s_j s_{j+1}$ are not necessarily in $UDel(V)$. One such example is that node u has a Delaunay neighbor x such that ux intersects edge $s_i s_{i+1}$ and $|ux| > 1$. In this case, edge $s_i s_{i+1}$ is not Delaunay edge, but s_i and s_{i+1} are consecutive neighbors of u in UDel since ux is removed.
   (c) Mark node u processed.

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

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

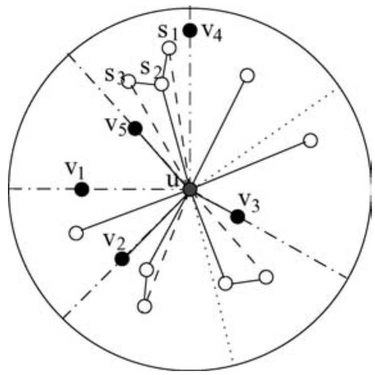


Figure 1. Constructing Planar Spanner with Bounded Degree for $UDG(V)$: Process node u. Here nodes v_i represents these nodes have already been processed by our method.

Our localized algorithm borrows some idea from our centralized method, and the proof of the correctness and the property of the structure constructed locally also uses some statements proved for centralized method. The following results were proven in [26].

Theorem 1. *Graph $BPS_1(V)$ is a planar graph. The maximum node degree of the graph $BPS_1(V)$ is at most $19 + \lceil \frac{2\pi}{\alpha} \rceil$. The spanning ratio of $BPS_1(V)$ is at most $t = \max\{\frac{\pi}{2}, \pi \sin\frac{\alpha}{2} + 1\} \cdot C_{\mathrm{del}}$. Here $0 < \alpha \leq \pi/3$.*

The proof of the spanner property is attached in the appendix (Section 7) since we will use it in the proof of our localized method. When $\alpha = \pi/3$, then the maximum node degree is at most 25. It improves the previous bound 27 on the maximum node degree by Bose et al. [2]. When $\alpha = \pi/3$, the spanning ratio is at most $(\frac{\pi}{2} + 1) \cdot C_{del}$; when $\alpha = 2\arcsin(\frac{1}{2} - \frac{1}{\pi}) \simeq 20.9^\circ$, then the spanning ratio is at most $\frac{\pi}{2} \cdot C_{\mathrm{del}}$.

Notice that the time complexity of the centralized algorithm is $O(n \log n)$, the same as with the method by Bose et al. [2]. However, this centralized algorithm has a smaller bounded node degree, and (more importantly) this algorithm has the potential to be turned into a localized algorithm as we will describe in this paper.

3. Localized Construction

In [14], Wang et al. showed that an algorithm presented in [30] does construct a bounded degree spanner for UDG with $O(n)$ messages (with unit $\log n$ bits) under the broadcast communication model, i.e., a signal sent by a node u can be received by all nodes within its transmission range. Li et al. [1] 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 show how to extend the centralized algorithm [26] reviewed in the previous section to generate a bounded degree and planar spanner for UDG in a localized manner. Remember that 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. Our algorithm is based on the efficient localized construction of a planar spanner $LDel^{(2)}(V)$ for UDG defined by Li et al. [1]. For completeness of the presentation, we first review the definitions and the 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 $UDG(V)$. All angles are measured in radians and take values in the range $[0, \pi]$. For any three points p_1, p_2, and p_3, the angle between the two rays $p_1 p_2$ and $p_1 p_3$ is denoted by $\angle p_3 p_1 p_2$ or $\angle p_2 p_1 p_3$. The *closed* infinite area inside the angle $\angle p_3 p_1 p_2$, also referred to as a sector, is denoted by $\sphericalangle p_3 p_1 p_2$. The triangle determined by p_1, p_2, and p_3 is denoted by $\triangle p_1 p_2 p_3$.

An edge uv is called constrained *Gabriel edge* (or simply Gabriel edge here) if $||uv|| \le 1$ and the open disk using uv as diameter does not contain any node from V. It is well known [32] 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 circum-disk, denoted as $disk(u,v,w)$, 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 $LDel^{(2)}(V)$.

Definition 1. A triangle $\triangle uvw$ satisfies the *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-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 [21,36] that $UDel(V)$ is a t-spanner of $UDG(V)$. Li et al. [1] proved that graph $UDel(V)$ is a subgraph of the k-localized Delaunay graph $LDel^{(k)}(V)$. Graph $LDel^{(1)}(V)$ is not a planar graph, and $LDel^{(k)}(V)$ is planar for $k > 1$. In [1], Li et al. proposed a communication efficient method to construct $LDel^{(1)}(V)$ and then make it planar in total $O(n)$ messages. Here each message has $O(\log n)$ bits.

In this paper, by plugging in the work from [33], we give the *first* method to construct $LDel^{(2)}(V)$ using $O(n)$ messages.

Algorithm 2: ***Localized construction of planar spanner $LDel^{(2)}(V)$***

(1) Every node u collects the location information of $N_2(u)$ based on an efficient method [33] (reviewed 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 $||uv|| \le 1$. 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 $\angle wuv \ge \frac{\pi}{3}$, 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).

First, we prove the following lemma which will be used in the analysis of our new algorithm. The proof of the lemma is included in the Appendix (Section 7).

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

We then review the communication efficient method proposed by Calinescu [33] to collect $N_2(u)$ for every node u when the geometry information is known. 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 [33] is based on the specific connected dominating set introduced by Alzoubi, Wan, and

Frieder [34]. This connected dominating set is based on a maximal independent set (MIS). In the algorithm, each node uses its adjacent node(s) in the MIS to broadcast over a larger area relevant information. 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 [33] shows that overloading certain nodes might be unavoidable.

We start from the moment the virtual backbone is already constructed, and every node knows the ID and the position of its neighbors. The idea of the algorithm is for every node to efficiently announce its ID and position to a subset of nodes which includes its 2-hop neighbors. The responsibility for announcing the ID and position of a node v is taken by the MIS nodes adjacent to v. Each such MIS node assembles a packet containing: <ID; *position*; *counter*>, with the ID and position of v, and a counter variable being set to 2. The MIS node then broadcasts the packet.

A connector node is used to establish a link in between several pairs of virtually-adjacent MIS nodes, and will not retransmit packets which do not travel in between these pairs of MIS nodes. The connector node will rebroadcast packets with nonzero counter originated by one of the nodes in a pair of virtually-adjacent MIS nodes, thus making sure the packet advances towards the other MIS node in the pair. Recall that the path in between a pair of virtually-adjacent MIS nodes has one or two connector nodes.

When receiving a packet of type < ID; *position*; *counter* >, a MIS node checks whether this is the first message with this ID, and if yes decreases the counter variable and rebroadcasts the packet. A node listens to the packets broadcasted by all the adjacent MIS nodes (here it is convenient to assume a MIS is adjacent to itself), and, using its internal list of 1-hop neighbors, checks if the node announced in the packet is a 2-hop neighbor or not — thus constructing the list of 2-hop neighbors.

The number of messages taken by this method is $O(n)$, which is proved in [33] by using the properties of the specific connected dominating set in [34]. Using the area argument, we can show that the constant in $O(n)$ is at most $3 \times (2 \times 7 + 1)^2 = 675$, 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.

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 for wireless ad hoc networks when every node has the same maximum transmission range. 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, as shown in Algorithm 3.

Algorithm 3: ***Localized construction of planar spanner with bounded degree***

(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 $d(u) \le 5$, then u queries[1] 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 $d(v) \le 5$, 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 $\pi_v = k$ for the first time, 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 $d(u) \le 5$ 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 an 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[2] 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 $\alpha \le \pi/3$. For each cone, let $s_1, s_2, \ldots, s_m$ 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 $s_1, s_2, \ldots, s_m$. Node u then tells $s_1, s_2, \ldots, s_m$ to add all the edges $s_j\, s_{j+1}$, $1 \le j < m$. Node u marks itself processed, and tells all nodes in $N_{LDel^{(2)}}(u)$ that it is processed.

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

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

[1] If some unordered neighbor v with $d(v) \le 5$ 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.

[2] There are at most 5 processed neighbors of u in $LDel^{(2)}(V)$ when u is being processed, because of the way the ordering is constructed and the fact that the graph $LDel^{(2)}(V)$ is planar.

3.3. Analysis of localized algorithm

We first show that the algorithm does process all nodes. First of all, the algorithm cannot stop at the stage of ordering nodes locally. This can be shown by contradiction. Assume that there are some nodes that are unordered. The graph formed by these unordered nodes is 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), and reduce the number of unordered nodes consequently.

Notice that the ordering computed by our method is not a global 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. 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 that is not less than 5. Using a larger integer may make the algorithm run faster, but on the other hand, it worsens the theoretical bound on the node degree.

We first show that the localized algorithm is communication efficient.

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

Proof: Notice that it was shown in [33] that we can collect the 2-hop neighbor information for all nodes using a total of $O(n)$ messages. The communication cost of building $LDel^{(2)}(V)$ is $O(n)$ since every node only has to propose at most 6 triangles and each proposal is replied to by two nodes.

The second step (local ordering) takes $O(n)$ messages, since every node only queries at most 5 rounds, and at the ith round of query the node sends at most $6-i$ query messages. For each query, only the queried node replies. After it was ordered, it broadcasts once to inform its neighbors.

The third step (bounded degree) also takes $O(n)$ messages, because every node only broadcasts twice: (i) to tell its neighbors to add some edges, and (ii) to claim that it is processed. The total number of messages of telling neighbors to add some edges is $O(n)$ since the total number of added edges is $O(n)$ from the planar property of the final topology. So the total communication cost is bounded by $O(n)$.

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 $LDel^{(1)}(V)$ from the definition of local Delaunay. 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 $Del(N_2(u))$ too.

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 $N_2(u)$ needs synchronized communication since otherwise, a node cannot determine if it has indeed collected $N_2(u)$.

BOUNDED DEGREE, PLANARITY AND SPANNING RATIO: Next, we show that the constructed final topology is still a planar spanner and has bounded node degree.

Theorem 4. *The maximum node degree of the graph $BPS_2(V)$ is at most $19 + \lceil \frac{2\pi}{\alpha} \rceil$.*

Proof: Notice that for a node u there are 2 cases that an edge uv is added to the $BPS_2(V)$. Let us discuss them one by one.

Case 1: When we process node u, some edges uv have already been added by some processed nodes w before. There are two subcases for this case.

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

Subcase 1.2: The edge uv has been added by a processed node w ($w \neq v$). Node v is an unprocessed node when processing w. For example, in figure 1, 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 at most adds two edges to node u.

Notice that each u can have at most 5 predecessor neighbors (i.e., processed neighbors), and each of the predecessors can add at most 3 edges to u (either Subcase 1.1 or Subcase 1.2, or both). Thus, the number of this kind of edges (edges added by its predecessors before u is processed) is bounded by 10+5 = 15.

Case 2: When node u is processed, we can add one edge uv for each cone. Since we have at most 5 sectors emanating from u and each cone must have an angle of 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 $4 + \lceil \frac{2\pi}{\alpha} \rceil$.

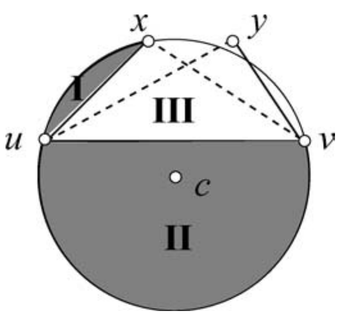


Figure 2. Two diagonal edges uy and vx intersect. The circum-disk *disk* (u, v, x) of the triangle $\triangle uvx$ is decomposed of three regions I, II, and III.

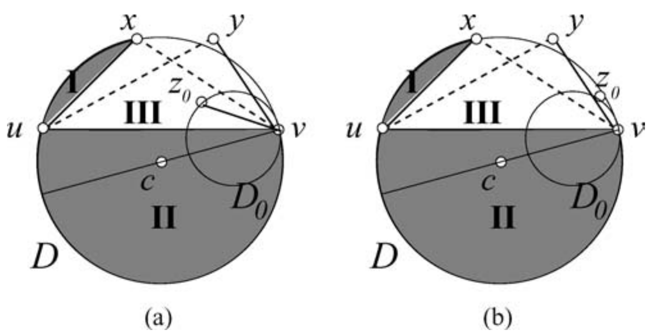


Figure 3. (a) z_0 is inside the cap cut by segment vy; (b) z_0 belongs to the sector $\sphericalangle\, uvy$.

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.

Notice that the algorithms in [2] and [26] 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 discussed 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 $BPS_1(V)$, in [26] we showed 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 for the centralized algorithm. However, this property does not hold anymore in the localized algorithm. We will show that $BPS_2(V)$ is a planar graph using a different approach.

Theorem 5. *$BPS_2(V)$ is a planar graph.*

Proof: Notice that Algorithm 3 only adds some 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 $\angle s_i\ us_{i+1} < \pi/3$. We call such an edge $s_i\ s_{i+1}$ the *diagonal* edge of the graph $LDel^{(2)}(V)$. Notice that[3] these diagonal edges cannot intersect with any edge from $LDel^{(2)}(V)$. Thus, the only possible intersections, if there is any, in $BPS_2(V)$ are caused by two diagonal edges. Without loss of generality, we assume that two diagonal edges uy and vx intersect with each other. Since uy is a diagonal edge, u and y are consecutive neighbors of some node, say p, in $LDel^{(2)}(V)$. From our previous discussion, the only possible intersection to the diagonal edge uy must be some diagonal edge incident at node p. Thus, p is either x or v here. See Figure 2 for an illustration of such two intersected diagonal edges uy and vx. Here we assume that p is v. In other words, edges vu and vy are consecutive neighboring edges in graph $LDel^{(2)}(V)$. Assume that $\angle uyv < \angle uxv$. Notice that $\angle uyv = \angle uxv$ will not happen by assuming that the nodes are in general position, i.e., no four vertices are co-circular. Then y is outside of the circumcircle $disk(u,v,x)$ of the triangle $\triangle uvx$.

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 consecutive neighboring edges in graph $LDel^{(2)}(V)$. Thus, there must be some node, say z, from $N_2(x) \cup N_2(v)$ inside the disk $disk(u,v,x)$. We then discuss the possible locations of z case by case.

If there is a node z that 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 the fact that 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 $\|uw\| \le 1$. 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 2, we can show that to have a node $z \in N_2(x)$ in region II is impossible. Similarly, region I cannot contain any node from $N_2(u) \cup N_2(x)$. Therefore, only region III can possibly contain some node z inside. Then $\|vz\| \le 1$. 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, $\|vz\| \le \|vx\|$ 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 $disk(u,v,x)$, since D is $disk(u,v,x)$ when c is the center of D. Among all such disks, we find the largest disk D_0 that does not have any nodes inside, i.e., the disk that passes 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 $\sphericalangle\, uvy$. 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 $\angle yuv + \angle yz_0\ v > \pi$. See Figure 3(a) for

[3] This is due to the following reason. The graph $LDel^{(2)}(V)$ is a planar graph. For each diagonal edge $s_i\ s_{i+1}$, nodes s_i and s_{i+1} are consecutive neighbors of a node u. This means that s_i, s_{i+1} and u belong to the same polygon face of $LDel^{(2)}(V)$. Thus, $s_i\ s_{i+1}$ cannot intersect any edge from $LDel^{(2)}(V)$.

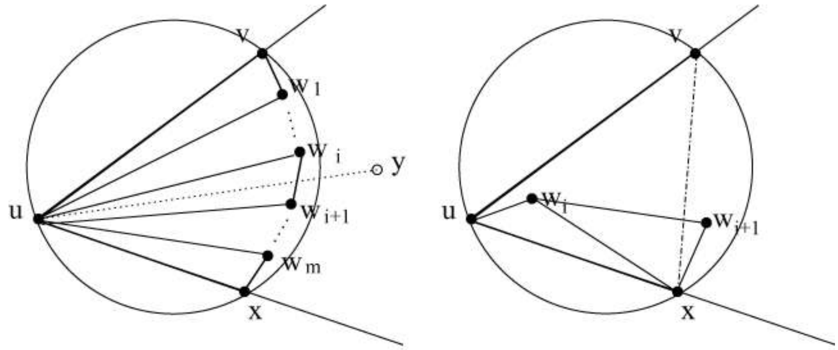


Figure 4. (a) All the neighbors w_i should be in the circumcircle $disk(u, v, x)$, and no edges other than Delaunay edges are added to u between ux and uv; (b) No edge $w_i\ w_{i+1}$ can have length longer than one.

illustrations. It contradicts to the existence of edge vy in graph $LDel^{(2)}(V)$.

As shown in Figure 3(b), if z_0 belongs to the sector $\sphericalangle\, uvy$, and $vz_0 \in LDel^{(2)}(V)$, then nodes y and u cannot be consecutive neighbors of v in $LDel^{(2)}(V)$. It is a contradiction.

Then we prove that the graph $BPS_2(V)$ has a bounded spanning ratio.

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

Proof: To prove the spanning property, we only need to study the bound on the spanning ratio for each individual edge instead of the bound on the spanning ratio for each shortest path. This can be simply proved. A similar proof is given in [12] as the proof of Lemma 1. Notice that[4] for any edge $uv \in UDG(V)$ we can find a path in $UDel(V)$ with length at most $C_{del}\|uv\|$, where $C_{del} = \frac{4\sqrt{3}}{9}\pi$, and every edge 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 $BPS_2(V)$ between u and v whose length is at most a constant ℓ times $\|uv\|$. Then $BPS_2(V)$ is a $\ell \cdot C_{del}$-spanner.

Now we prove the above claim. Consider an edge uv in $UDel(V)$. If $uv \in BPS_2(V)$, the claim holds. So assume that $uv \notin BPS_2(V)$.

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 $\|uv\| > \|ux\|$, $ux \in BPS_2(V)$, and $\angle xuv < \alpha \leq \pi/3$. There are 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 as in Theorem 7 (in the Appendix) to prove that there is a path in $BPS_2(V)$ connecting u and v with length at most $\max\{\frac{\pi}{2},\ \pi \sin\frac{\alpha}{2} + 1\} \cdot \|uv\|$.

Let $w_1, w_2, \ldots s, w_m$ be the sequence of Delaunay neighbors of u in $Del(V)$ from v to x. See Figure 4(a) for 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 that uw_i is Delaunay triangle. Then we prove that all the edges $w_i\ w_{i+1}$ are shorter than one unit.

Remember that if $\|uv\| \leq 1$, $\|ux\| \leq 1$ 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 4(b). Assume that ≤ngth$w_i\ w_{i+1} >$ 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 $w_i\ w_{i+1} \in UDel(V)$, and in addition, they are also in $LDel^{(2)}(V)$ (since $UDel(V) \subseteq LDel^{(2)}(V)$). Therefore we can not have an additional edge uy added to $LDel^{(2)}(V)$ in sector $\sphericalangle\, vux$, since such an edge breaks the planar property of $LDel^{(2)}(V)$. See Figure 4(a) for illustrations.

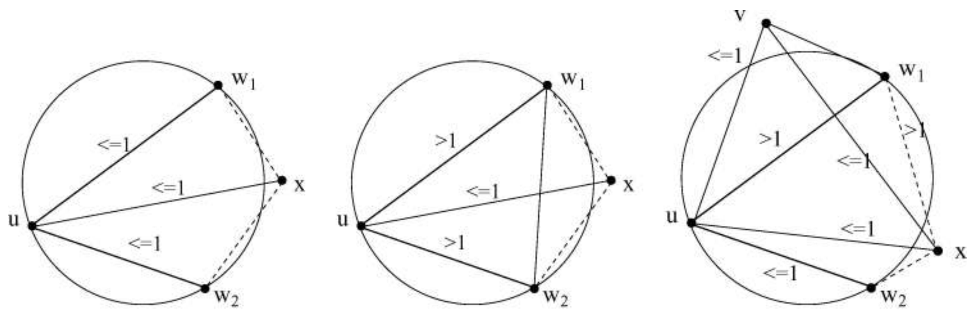
Case 2: $ux \notin UDel(V)$. Assume ux is added to $LDel^{(2)}(V)$ in the sector $\sphericalangle\, w_1\ uw_2$, where w_1 and w_2 are consecutive Delaunay neighbors of node u. There are three cases for Delaunay edges $w_1 u$ and $w_2 u$. We prove that all of them do not exist by contradiction.

Subcase 2.1: both edges $w_1\ u$ and $w_2\ u$ are no more than one unit, shown in Figure 5(a). From the property of Delaunay, x must be outside of the circumcircle $disk(u,w_1,w_2)$ of the triangle $\triangle\ uw_1\ w_2$. Thus,$\angle uw_1\ x + \angle uw_2\ x > \pi$. Any circle passing through u and x will contain either w_1 or w_2 inside. Notice that $w_1, w_2 \in N_1(u)$. It contradicts the existence of edge ux in $LDel^{(2)}(V)$.

Subcase 2.2: both edges $w_1\ u$ and $w_2\ u$ are longer than one unit, shown in Figure 5(b). Since $\|uw_1\| > 1 \geq \|ux\|$, $\angle uw_1\ x < \pi/2$. Similarly, $\angle uw_2\ x < \pi/2$. Then we have $\angle uw_1\ x + \angle uw_2\ x < \pi$, which contradicts the assumption that 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

[4]Please refer to the proofs of Lemma 4 and Theorem 5 in [36]. They proved that $UDel(V)$ is a t-spanner of $UDG(V)$.



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

Figure 5. All subcases in Case 2 do not exist.

than one unit. Assume that $\|w_1\, u\| > 1$. See Figure 5(c) as illustrations.

Since edge $ux \in LDel^{(2)}(V)$, we know $\|xw_1\| > 1$. Otherwise, if 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. Plus $\|uw_1\| > 1$ and $\|ux\| \leq 1$, we have $\angle uw_1\, x < \pi/3$. From x is outside the circumcircle $disk(u,\, w_1,\, w_2)$, $\angle uw_1\, x + \angle uw_2\, x > \pi$. Thus,$\angle uw_2\, x > 2\pi/3$, which implies $\|ux\| > \|uw_2\|$. Therefore, in Algorithm 3, no edge uv from $UDel(V)$ which is below edge ux will select ux as the shortest neighbor in the same cone, because it will select uw_2.

Consider that an edge $uv \in UDel(V)$, which is above edge ux, selects ux as the shortest neighbor. 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 $\angle vw_1\, x + \angle vux > \pi$. Thus, $\angle vw_1\, x > \pi - \angle vux > 2\pi/3$. Then $1 \geq \|vx\| > \|xw_1\| > 1$ causes a contradiction. Therefore Subcase 2.3 shown in figure 5(c) does not exist too.

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

3.4. Dynamic update

After the construction of the topology, dynamic maintenance is also an important issue, since an ad hoc network could be highly dynamic. Three major events may cause the topology obsoleted: due to

(1) node moving,
(2) node joining or leaving, and
(3) node failure.

Therefore, a dynamic update method for our proposed topology is needed. Usually, there are two kinds of update methods: on-demand update or periodical update. Most of the existing topology control algorithms are invoked periodically, while some algorithms perform the updating only when it is required (i.e., on-demand). Our algorithm can adapt and combine both of these two update methods. If no major topology changes (for example, some small node movements do not affect the topology), no update will be performed until some pre-set timer expires. In other words, we perform our algorithm periodically with a pre-set time. The time could be set quite long depending on the types of the applications. But for some major topology change (such as a node's death or a tremendous movement of nodes), an on-demand update will be performed. Notice that since our algorithm is a localized algorithm, the update process can be performed only in a local area (within 2-hop neighborhood) where the topology change occurs. For example, When a node u moves around, if a triangle $\triangle\, xyz$ in the local Delaunay disappears or a new triangle $\triangle\, xyz$ appears in the new local Delaunay, then u is a (2-hop) neighbor of either x or y or z (if $LDel^2$ is used). In other words, the movement of a node u only affects its local neighborhood of the local Delaunay triangulation, thus also the structure defined in this paper.

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 proposed 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 8. We generate 100 vertex sets V (each with 50 vertices) and then generate the

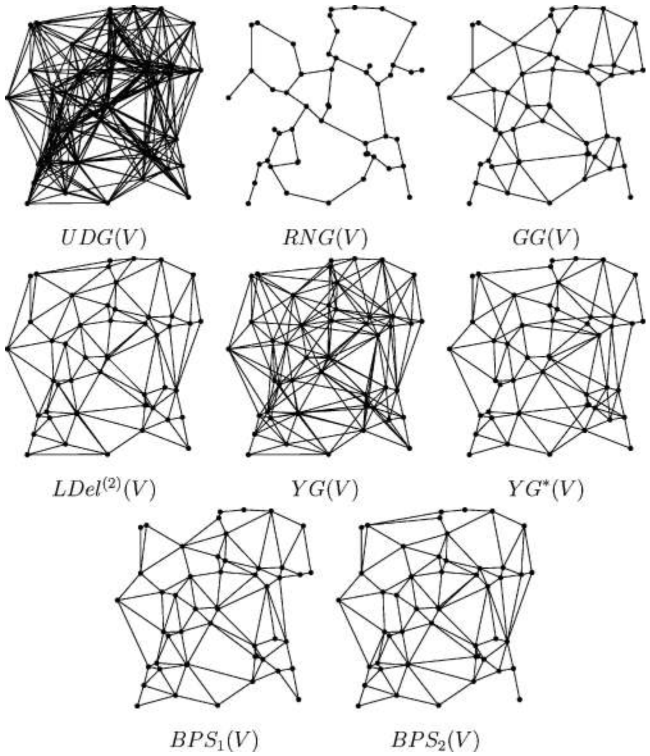


Figure 6. Different topologies from the same *UDG*(*V*).

Table 1
Node degrees and stretch factors of different topologies.

	d_{avg}	d_{max}	t_{avg}	t_{max}	ρ_{avg}	ρ_{max}
UDG	16.83	35	1.000	1.000	1.000	1.000
RNG	2.27	5	1.320	5.049	1.059	2.942
GG	3.36	8	1.120	2.131	1.000	1.000
LDel	5.25	11	1.048	1.405	1.000	1.000
YG	8.11	19	1.040	1.681	1.002	1.459
YG^*	4.81	11	1.070	1.990	1.003	1.459
BPS_1	4.44	9	1.075	1.965	1.004	1.755
BPS_2	4.45	9	1.074	1.965	1.004	1.823

graphs for each of these 100 vertex sets. The average and the maximum are computed over all of these 100 vertex sets. Figure 6 gives all seven different topologies for the unit disk graph illustrated by the first figure of Figure 6. It shows that all of these topologies except $YG(V)$ and $YG^*(V)$ are planar.

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 $d_{\text{avg}}/d_{\text{max}}$ is the average/maximum node degree. It shows that $BPS_1(V)$ and $BPS_2(V)$ have a 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 6. 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 [12,13], Li et al. gave an example to show that $RNG(V)$, $GG(V)$, $YG(V)$ and $LDel(V)$ can 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 is not as big as the example. It is proved that the node degree of $YG^*(V)$ is bounded from above by $(k+1)^2 - 1$ (the in-degree is at most $k(k+1)$, the 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 a bounded node degree of at most $19 + \lceil \frac{2\pi}{\alpha} \rceil = 25$ when $\alpha = \pi/3$. All of these theoretical bounds on the node degree can be verified by the maximum node degrees in Table 1. Both $BPS_1(V)$ and $BPS_2(V)$ have smaller maximum node degrees than $YG(V)$.

4.2. Spanner Properties

Besides the 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 the *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 that the total transmission power consumed by path $v_0, v_1, \ldots, v_k$ 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 of these topologies. Here, $t_{\text{avg}}/t_{\text{max}}$ is the average/maximum length stretch factor; $\rho_{\text{avg}}/\rho_{\text{max}}$ is the average/maximum power stretch factor. It is not surprising that the average/maximum power stretch factors of $BPS_1(V)$ and $BPS_2(V)$ are small and at the same level of those of the $YG(V)$ and $YG^*(V)$ while they are planar and much sparser. Notice that Yao graph does perform a little bit better in our simulations in term of spanner properties, but it is not a planar structure and also *cannot* bound the nodal degree.

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

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

num_of_nodes	50	100	150	200	250	300
$d_{avg}(UDG)$	16.81	34.98	51.79	68.25	85.89	103.87
$d_{max}(UDG)$	35	63	93	114	141	177
d_{avg}	4.43	4.49	4.53	4.61	4.58	4.63
d_{max}	9	9	11	11	10	9
t_{avg}	1.079	1.091	1.090	1.092	1.093	1.089
t_{max}	1.958	1.964	1.949	1.965	1.968	1.963
ρ_{avg}	1.005	1.007	1.006	1.005	1.005	1.006
ρ_{max}	1.865	1.891	1.850	1.872	1.861	1.873
tot_msg_{avg}	443	912	1379	1855	2340	2798
tot_msg_{max}	448	921	1394	1870	2326	2812
nod_msg_{avg}	8.86	9.13	9.19	9.27	9.30	9.32
nod_msg_{max}	13	14	16	15	17	15

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 50 vertex sets. All other parameters and settings are the same as those in 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 $LDel^{(2)}(V)$. In other words, we only consider the messages used in the second and third steps of Algorithm 3. Remember that by plugging in the work from [33], we can construct $LDel^{(2)}(V)$ using $O(n)$ messages. However, the hidden constant is pretty large. Therefore, in this experiment, we use a naive method (in which each node broadcasts its one-hop neighbor information to its all neighbors) to collect 2-hop neighbor information and directly build $LDel^{(2)}(V)$ based on the information. 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 the number of nodes changes.

5. Conclusion

In this paper, we proposed a localized algorithm to construct planar spanners with bounded node degree for wireless ad hoc networks based on a centralized method we developed. 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 a localized Delaunay graph to construct a planar spanner graph, and then to apply some ordered Yao graph to bound the node degree. It is carefully designed not to lose all the good properties when combining them. To the best of our knowledge, this is the first localized algorithm for constructing a bounded degree and planar spanner. We also conducted experiments to show that this topology is efficient in practice compared with other well-known topologies for wireless ad hoc networks. It is still an open problem of how to bound the total edge length of our localized structure $BPS_2(V)$.

6. Acknowledgment

The authors greatly appreciate the anonymous reviewers and the editors for their constructive comments for improving the paper.

7. Appendix

7.1. Proof of Spanner Property for Centralized Method

Here we review the proof of spanner property for the centralized method, since the proof of localized method uses some techniques presented here.

Theorem 7. *Graph $BPS_1(V)$ is a t-spanner, where* $t \le \max\{\frac{\pi}{2}, \pi \sin\frac{\alpha}{2} + 1\} \cdot C_{\text{del}}$.

Proof: For completeness, we review the proof here. Keil and Gutwin [25] showed that the Delaunay triangulation is a t-spanner for a constant $C_{del} = \frac{4\sqrt{3}}{9}\pi$ using induction on the increasing order of the lengths of all pairs of nodes. We can show that the path connecting nodes u and v constructed in [25] also satisfies that the length of each edge of that path is at most $\|uv\|$. Consequently, for any edge $uv \in UDG(V)$ we can find a path in $UDel(V)$ with length at most a $t = \frac{4\sqrt{3}}{9}\pi$ times $\|uv\|$, 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 $BPS_1(V)$ between u and v whose length is at most a constant ℓ times $\|uv\|$. Then $BPS_1(V)$ is a $\ell \cdot C_{del}$-spanner.

Now we prove the claim above. Consider an edge uv in $UDel(V)$. If $uv \in BPS_1(V)$, the claim holds. So assume that $uv \notin BPS_1(V)$.

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 $\|uv\| > \|uv'\|$, $uv' \in BPS_1(V)$, and $\angle v'uv < \alpha \le \pi/3$. Let $v' = s_1, s_2, \ldots, s_l = v$ be this sequence of nodes in the ordered unprocessed neighborhood of u in $UDel(V)$ from v' to v. Let $v' = w_1, w_2, \ldots, w_k = v$ be the sequence of neighbors of u in $Del(V)$ from v' to v. Obviously, the set $\{s_1, s_2, \ldots, s_l\}$ is a subset of $\{w_1, w_2, \ldots, w_k\}$.

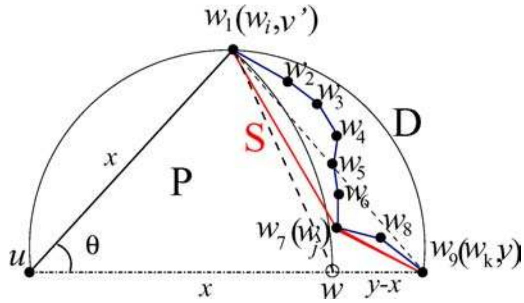


Figure 7. The shortest path in polygon P.

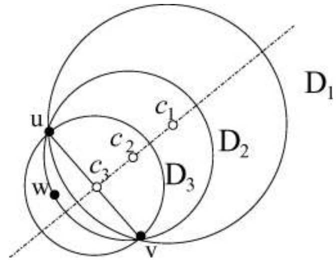


Figure 8. Disk D_2 touches a node w from $N_2(u) \cup N_2(v)$.

Similar to [2], consider the polygon P, formed by edge uw_1, uw_k and path $w_1\ w_2 \dots w_k$. We will show that the path $w_1\ w_2 \dots w_k$ 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 $S(w_1, w_\mathrm{k})$ denote such a path. This path consists of diagonals of P and is contained inside $\triangle uw_1\ w_k$. For example, in Figure 7, $S(w_1,w_k) = w_1\ w_7\ w_9$.

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 $S(w_1, w_\mathrm{k})$ is contained inside the triangle $\triangle ww_1\ w_\mathrm{k}$. First, if no Delaunay neighbors are inside $\triangle ww_1\ w_\mathrm{k}$, then $S\ (w_1,w_\mathrm{k}) = w_1\ w_\mathrm{k}$. Thus, the claim trivially holds. If there are some Delaunay neighbors inside $\triangle ww_1\ w_\mathrm{k}$, then w_1 will connect to the one w_i forming the smallest angle $\angle uw_1\ w_i$. Similarly, node w_k will connect to the one w_j forming the smallest angle $\angle uw_\mathrm{k}\ w_\mathrm{j}$. Obviously w_i and w_j are inside $\triangle ww_1\ w_\mathrm{k}$, thus, the shortest path connecting them is also inside $\triangle ww_1\ w_\mathrm{k}$. Since path $S(w_1,w_\mathrm{k})$ is the shortest path inside the polygon P to connect w_1 and w_k, by convexity, the length of $S(w_1,w_\mathrm{k})$ is at most $\|v'\ w\| + \|wv\| = 2x \sin\frac{\theta}{2} + y - x$. Here $\theta = \angle v'\ uv < \alpha$.

An edge $w_i\ w_\mathrm{j}$ of $S(w_1,w_\mathrm{k})$ has endpoints w_i and w_j in the neighborhood of u. Let $D(w_i,w_\mathrm{j})$ 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 7, $D(w_1, w_7) = w_1\ w_2\ w_3\ w_4\ w_5\ w_6\ w_7$. We can bound the length of $D(w_i, w_j)$ by $\pi/2\ \|w_i\ w_\mathrm{j}\|$ by the argument in [2,29]. In [29], it is shown that the length of $D(w_i,w_\mathrm{j})$ is at most $\pi/2$ times $\|w_i\ w_\mathrm{j}\|$, 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 $D(w_i,w_\mathrm{j})$ to be *one-sided Direct Delaunay path*[5] [23]. In [2], they showed[6] that both of these two conditions hold when $\angle w_i\ uw_j < \pi/2$. This is trivially satisfied since $\angle w_i\ uw_\mathrm{j} < \alpha \le \pi/2$.

Thus, we have a path $uw_1, w_2, \dots, w_\mathrm{k}$ to connect u and v with length at most

$$
\begin{aligned}
& x + \left(2x \sin\frac{\theta}{2} + y - x\right) \cdot \pi/2 \\
& \le y \cdot \left(\frac{\pi}{2} + \frac{x}{y} \cdot \left(\pi \sin\frac{\alpha}{2} - \frac{\pi}{2} + 1\right)\right) \\
& \le y \cdot \max\left\{\frac{\pi}{2}, \pi \sin\frac{\alpha}{2} + 1\right\}
\end{aligned}
$$

Since any such node w_i is not inside the polygon Q formed by the Unit Delaunay neighbors of u (see [26] for more detail), the path $us_1, s_2, \dots, s_1$ (which is in $BPS_1(V)$) is not longer than the length of path $uw_1w_2\cdots w_\mathrm{k}$.

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

7.2. *Proof of Lemma 2*

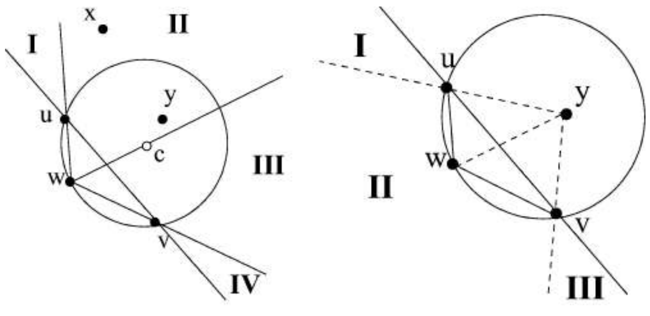
Lemma 2. An edge uv is in $LDel^{(2)}(V)$ iff $\|uv\| \le 1$ and there is a disk passing through u and v which does not contain a node from $N_2(u) \cup N_2(v)$ inside.

Proof: It is trivial that if an edge uv is in $LDel^{(2)}(V)$ then that kind of 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 the 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 8. Disk D_1 cannot

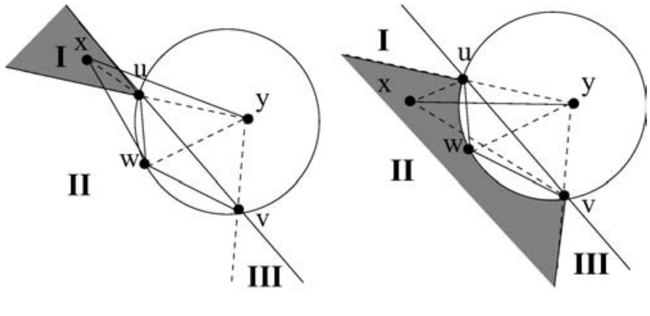
[5]For any pair of nodes u and v, let $u = w_1, w_2, \cdots, w_\mathrm{k} = v$ be the sequence of nodes whose Voronoi region intersect segment uv and the Voronoi regions at w_i and w_j share a common boundary segment. The the Direct Delaunay path $DT(u,v)$ is $w_1w_2\cdots w_\mathrm{k}$.

[6]Firstly, the Voronoi region centered at u will not intersect the segment w_iw_j. This can be proved by showing that $\|up\| > \max\{\|w_i\ p\|, \|w_j\ p\|\}$ for any point p on segment w_iw_j, which is due to $\angle uw_ip + \angle uw_j\ p > \angle w_i\ up + \angle w_jup = \angle w_i\ uw_j$. Notice that $\angle w_i\ uw_\mathrm{j} < \alpha \le \pi/2$. Secondly, the path $D(w_i,w_\mathrm{j})$ is on one-side of $w_i\ w_\mathrm{j}$ because it is part of the shortest path connecting w_1 and w_k. Thirdly, the path $D(w_i,w_j)$ is Direct Delaunay path $DT(w_i,w_\mathrm{j})$. This can be proved by showing that $Vor(w_q)$ intersects the segment $w_i\ w_j$ for any $i \le q \le j$. This is obvious since the circumcenter (belonging to $Vor(w_\mathrm{q})$) of any triangle $uw_{q-1}\ w_q$ is on the same side of $w_i\ w_j$ as u.



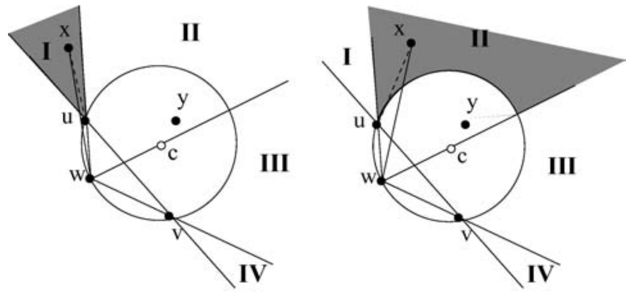
(a) Different side (b) Same side

Figure 9. Two cases in the proof: x is on the same side or different side of uv as y.



(a) Subcase 1 (b) Subcase 2

Figure 10. Node x is inside region I or region II.



(a) Subcase 1 (b) Subcase 2

Figure 11. Node x is inside region I or region II.

contain w inside. Assume D_1 has center c_1. Let D be a disk centered at some point c on the segment $c_1\ c_3$ and passing through u and v. Then we can move the center c of disk D along $c_1\ c_3$ from c_1 to c_3 and set the radius of D be $\|cu\|$, until the disk touches the *first* node w from $N_2(u) \cup N_2(v)$ or becomes D_3.

If the disk becomes D_3, then uv is a Gabriel edge and in $LDel^{(2)}(V)$. Otherwise, the disk D touches some node w, which is shown in Figure 8 as disk D_2. Then D becomes the circumcircle $disk(u,v,w)$ of u, v and w. 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.

Assume that there is a node y from $N_2(w)$ inside $disk(u,v,w)$. Clearly, node y cannot be from $N_2(u) \cup N_2(v)$, since D_2 does not contain any node from $N_2(u) \cup N_2(v)$ inside. 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 $\|uv\| \leq 1$. Assume that a node x is one hop neighbor of both y and w. Notice that x cannot be a 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. 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 9. Clearly, node x is outside of the disk D, otherwise, D will contain a 2-hop neighbor x of u inside (through path uwx).

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 9(a).

If x is inside the region I, see Figure 10(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 10(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 $\angle xvy < \pi/3$. Thus, $2\pi = \angle xuy + \angle uxv + \angle xvy + \angle xvy < 4\pi/3$, which is a contradiction.

When node x is inside 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 node x is inside region I, II, III, or V. 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 11(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 11(b) for an illustration. Here c is the circumcenter of the disk D. Notice that when node x is on the diagonal wc and just outside the circle, $\angle wux$ has the minimum value slightly larger than $\pi/2$. 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 V, the proofs are similar to the cases II, or I respectively.

Then we know that 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.

References

[1] 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, 14 (10), (2003) 1035–1047. Short version was in INFOCOM 2002.

[2] 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).

[3] M. Bahramgiri, M.T. Hajiaghayi, and V.S. Mirrokni, Fault-tolerant and 3-dimensional distributed topology control algorithms in wireless multi-hop networks, in: IEEE International Conference on Computer Communications and Networks (ICCCN) (2002).

[4] L. Hu, Topology control for multihop packet radio networks, IEEE Trans. Communications, 41 (10), (1993).

[5] 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 Symposium on Principle of Distributed Computing (PODC) (2001).

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

[7] R. Ramanathan and R. Rosales-Hain, Topology control of multihop wireless networks using transmit power adjustment, in: *IEEE INFOCOM* (2000).

[8] R. Wattenhofer, L. Li, P. Bahl and Y.-M. Wang, Distributed topology control for wireless multihop ad-hoc networks, in: IEEE INFOCOM'01 (2001).

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

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

[11] L. Jia, R. Rajaraman and C. Scheideler, On local algorithms for topology control and routing in ad hoc networks, in: *Proceedings of the Fifteenth Annual ACM Symposium on Parallel Algorithms and Architectures*, 2003.

[12] X.-Y. Li, P.-J. Wan and Y. Wang, Power efficient and sparse spanner for wireless ad hoc networks, in: *IEEE International Conference on Computer Communications and Networks (ICCCN01)*, (2001), pp. 564–567.

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

[14] 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.

[15] L. Kleinrock and J. Silvester, Optimum transmission radii for packet radio networks or why six is a magic number, in: *Proc. of the IEEE National Telecommunications Conference*, 1978, pp. 431–435.

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

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

[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] 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).

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

[22] Limin Hu, Topology control for multihop packet radio networks, *IEEE Transactions on Communications*, vol. 41, 1993.

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

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

[25] J. M. Keil and C. A. Gutwin, Classes of graphs which approximate the complete euclidean graph, Discrete Computational Geometry, 7 (1992).

[26] X.-Y. Li and Y. Wang, Efficient construction of low weight bounded degree planar spanner, International Journal of Computational Geometry and Applications, 14, (1/2), (2004) 69–84. Short version was in COCOON 2003.

[27] J. Gudmundsson, C. Levcopoulos, and G. Narasimhan, "Improved greedy algorithms for constructing sparse geometric spanners," in: *Scandinavian Workshop on Algorithm Theory*, (2000), pp. 314–327.

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

[29] P. Bose and P. Morin, Online routing in triangulations, in: *Proc. of the 10th Annual Int. Symp. on Algorithms and Computation ISAAC* (1999).

[30] S. Arya, G. Das, D. Mount, J. Salowe and M. Smid, Euclidean spanners: short, thin, and lanky, in: *Proc. 27th ACM STOC*, (1995) pp. 489–498.

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

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

[33] G. Călinescu, Computing 2-hop neighborhoods in ad hoc wireless networks, 2002, in *Proc. AdHoc-Now* (2003).

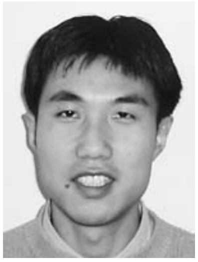
[34] K. Alzoubi, P.-J. Wan and O. Frieder, Message-optimal connected-dominating-set construction for routing in mobile ad hoc networks, in *3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)* (2002).

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

[36] X.-Y. Li, G. Calinescu and P.-J. Wan, Distributed construction of planar spanner and routing for ad hoc wireless networks, in *IEEE INFOCOM* (2002).

Yu Wang is an assistant professor in the Department of Computer Science, University of North Carolina at Charlotte. He received his Ph.D. degree in computer science from Illinois Institute of Technology in 2004, his B.S. degree and M.S. degree in computer science from Tsinghua University, China, in 1998 and 2000. His current research interests include computer networks, wireless networks, mobile computing, algorithm design, and artificial intelligence. His recent work focuses on designing power efficient algorithms for wireless ad hoc networks and sensor networks. He published more than 40 papers in peer-reviewed journals and conferences. He served as program committee member for sevaral conferences (such as IEEE INFOCOM, IEEE MASS, IEEE ICCCN, etc.). He also served as reviewers for a number of international journals and conferences. His paper titled "Sparse Power Efficient Topology for Wireless

Networks" won a Best Paper Award from the 35th IEEE Hawaii International Conference on System Sciences in 2002. He is a member of the ACM, IEEE, and IEEE Communication Society. For more information, please see http://www.cs.uncc.edu/∼ywang32.
E-mail: ywang32@uncc.edu



Xiang-Yang Li has been an Assistant Professor of Computer Science at the Illinois Institute of Technology since 2000. He received M.S. (2000) and Ph.D. (2001) degree at Department of Computer Science from University of Illinois at Urbana-Champaign. He received his Bachelor degree at Department of Computer Science and Bachelor degree at Department of Business Management from Tsinghua University, P.R. China in 1995. He is a member of the Chinese national team prepared for the International Mathematics Olympics (IMO) from 1988 to 1990. His research interests span the wireless ad hoc networks, game theory, computational geometry, and cryptography and network security. Recently, he focuses on performing research on the cooperation, energy efficiency, and distributed algorithms for wireless ad hoc and sensor networks. He has published about 60 conference papers in top-quality conferences such as ACM MobiCom, ACM MobiHoc, ACM SODA, ACM STOC, IEEE INFOCOM, etc. He has more than 30 journal papers published or accepted for publish. He is a Member of the ACM, IEEE, and IEEE Communication Society.
Xiang Yang Li served various positions (such as conference chair, local arrangement chair, financial chair, session chair, TPC member) at a number of international conferences such as IEEE INFOCOM, ACM MobiHoc, ACM STOC. Li recently also co-organized a special issue of ACM MONET on non-cooperative computing in wireless networks. For more information, please see http://www.cs.iit.edu/∼xli.
E-mail: xli@cs.iit.edu