Mobile Networks and Applications 10, 911–923, 2005
© 2005 Springer Science + Business Media, Inc. Manufactured in The Netherlands.
DOI: 10.1007/s11036-005-4447-3

Localized Algorithms for Energy Efficient Topology in Wireless Ad Hoc Networks

WEN-ZHAN SONG
School of Engineering and Computer Science, Washington State University, Vancouver, WA 98686

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

XIANG-YANG LI* and OPHIR FRIEDER
School of Engineering and Computer Science, Washington State University, Vancouver, WA 98686

Published online: 24 October 2005

Abstract. Topology control in wireless ad hoc networks is to select a subgraph of the communication graph (when all nodes use their maximum transmission range) with some properties for energy conservation. In this paper, we propose two novel localized topology control methods for homogeneous wireless ad hoc networks.

Our first method constructs a structure with the following attractive properties: power efficient, bounded node degree, and planar. Its power stretch factor is at most $\rho = \frac{1}{1-(2\sin\frac{\pi}{k})^\beta}$, and each node only has to maintain at most $k + 5$ neighbors where the integer $k > 6$ is an adjustable parameter, and β is a real constant between 2 and 5 depending on the wireless transmission environment. It can be constructed and maintained locally and dynamically. Moreover, by assuming that the node ID and its position can be represented in $O(\log n)$ bits each for a wireless network of n nodes, we show that the structure can be constructed using at most $24n$ messages, where each message is $O(\log n)$ bits.

Our second method improves the degree bound to k, relaxes the theoretical power spanning ratio to $\rho = \frac{\sqrt{2}^\beta}{1-(2\sqrt{2}\sin\frac{\pi}{k})^\beta}$, where $k > 8$ is an adjustable parameter, and keeps all other properties. We show that the second structure can be constructed using at most $3n$ messages, where each message has size of $O(\log n)$ bits.

We also experimentally evaluate the performance of these new energy efficient network topologies. The theoretical results are corroborated by the simulations: these structures are more efficient in practice, compared with other known structures used in wireless ad hoc networks and are easier to construct. In addition, the power assignment based on our new structures shows low energy cost and small interference at each wireless node.

Keywords: wireless ad hoc networks, topology control, bounded degree, planar, spanner, efficient localized algorithm, power assignment

1. Introduction

Wireless *ad hoc* networks have been undergoing a revolution that promises to* have a significant impact throughout society. Unlike traditional fixed infrastructure networks, there is no centralized control over ad hoc wireless networks, which consist of an arbitrary distribution of radios in certain geographical area. In ad hoc networks, mobile devices can communicate via multi-hop wireless channels; a node can reach all nodes in its transmission region, while two far-away nodes communicate through the message relaying by intermediate nodes. Wireless ad hoc networks trigger many challenging research problems, as it intrinsically has many special characteristics and some unavoidable limitations, compared with other wired or wireless network. An important requirement of these networks is that they should be self-organizing, i.e., transmission ranges and data paths are dynamically restructured with changing topology. Energy conservation and network performance are probably the most critical issues in ad hoc wireless networks, because wireless devices are usually powered by batteries only and have limited computing capability and memory.

The *topology control* technique is to let each wireless device adjust its transmission range and select certain neighbors for communication, while maintaining a structure that can support energy efficient routing and improve the overall network performance. By enabling each wireless node shrinking its transmission power (which is usually much smaller than its maximal transmission power) to sufficiently cover the farthest selected neighbor, topology control can not only save energy and prolong network life, but also can improve network throughput through mitigating the MAC-level medium contention. Unlike traditional wired networks and cellular wireless networks, the wireless devices are often moving during the communication, which could change the network topology in some extent. Hence it is more challenging to design a topology control algorithm for ad hoc wireless networks: the topology should be locally and self-adaptively

*The work of Xiang-Yang Li is partially supported by NSFCCR-0311174.

maintained without affecting the whole network, and the communication cost during maintaining should not be too high.

Topology control has drawn significant research interest [9,16,18,19,21,22,26,27] in last few years. Different topologies have different properties, however, none of them can achieve all three preferred properties for unicast applications on wireless ad hoc networks: power spanner, planar, degree-bounded. Until recently, Wang and Li [25] proposed a localized algorithm to build a degree-bounded planar spanner both in centralized and distributed way, which is based on the combination of *localized Delaunay triangulations* (LDel) [17] and *Yao* structure [28]. It is the first localized algorithm that can achieve all the three desirable features. However, the node degree of their structure can reach 25 in the worst case; and the communication cost of their method can be large, although it is shown that the total number of messages is $O(n)$, the hidden constant could be as high as several hundreds since the method needs to collect the 2-hop information for every node.

In this paper, we propose two novel methods to build a power efficient planar structures with much less communication costs and lower node degree bounds. Our first structure has the following attractive properties:

1. It is power efficient: given any two nodes u and v, there is a path connecting them in the structure with a total power cost no more than $\rho = \frac{1}{1-(2\sin\frac{\pi}{k})^{\beta}}$ times of the power cost of any path connecting them in the original homogeneous network;
2. Its node degree is bounded from above by a positive constant $k+5$ where integer $k > 6$ is an adjustable parameter;
3. It is a planar structure, which enables several localized routing algorithms;
4. It can be constructed and maintained in a localized and dynamic way.

Moreover, by assuming that the node ID and its position can be represented in $O(\log n)$ bits each for a wireless network of n nodes, we show that the structure can be constructed using at most $24n$ messages, where each message is $O(\log n)$ bits. Our second method reduces the degree bound to k, and keeps all other properties, except that the theoretical power spanning ratio is relaxed to $\rho = \frac{\sqrt{2}^{\beta}}{1-(2\sqrt{2}\sin\frac{\pi}{k})^{\beta}}$, where $k > 8$ is an adjustable parameter. We show that the second structure can be constructed using at most $3n$ messages, where each message has size of $O(\log n)$ bits.

We also experimentally evaluate the performance of these new energy efficient network topologies. The theoretical results are corroborated in the simulations: our new structures are more efficient in practice and easier to construct, compared to other known structures used in wireless ad hoc networks. By shrinking the transmission range of each node to reach the farthest neighbors in our new structures, the experiment shows that each node indeed costs low energy and has a small number of *physical neighbors*. The *physical neighbors* are those nodes within its transmission region, and smaller number of *physical neighbors* means less interference.

The rest of the paper is organized as follows. In Section 2, we describe some most preferred properties of topology control protocol in wireless ad hoc networks and review the related works in this area. We then present our two localized methods, in Section 3, to construct degree-bounded planar power spanners for $UDG(V)$ with total communication cost $O(n)$ under the broadcasting communication model. In Section 2, we conduct extensive simulations to validate our theoretical results. Finally, we conclude the paper in Section 5.

2. Preliminaries

2.1. Network model

A wireless ad hoc network (or sensor network) consists of a set V of n wireless nodes distributed in a two-dimensional plane. Each node has the same *maximum* transmission range R.[1] 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. In other words, we assume that two nodes can always receive the signal from each other directly if the Euclidean distance between them is no more than one unit. 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 some other ways. More specifically, in our protocol, it would be enough if each node knows the relative position of its one-hop neighbors. The relative position of neighbors can be estimated by the *direction of signal arrival* and the *strength of signal*. By one-hop broadcasting, each node u can gather the location information of all nodes within its transmission region.

In the most common power-attenuation model, the power to support a link uv is assumed to be $||uv||^{\beta}$, where $||uv||$ is the Euclidean distance between u and v, β is a real constant between 2 and 5 depending on the wireless transmission environment.

2.2. Preferred properties

Wireless ad hoc network topology control schemes are to maintain a structure that can be used for efficient routing [3,10] or improve the overall networking performance [9,16,22] by selecting a subset of links or nodes used for communication. In the literature, the following desirable features are well-regarded and preferred in wireless ad hoc networks:

Power spanner: In ad hoc wireless networks, two far-apart nodes can communicate with each other through the relay of intermediate nodes; hence, each node only need set small transmission ranges. This has two advantages: (1) reducing

[1] In practice, R can be defined as the minimum of all the maximum node transmission ranges.

the signal interference, (2) saving transmission power. To guarantee the advantage, a good network topology should be energy efficient, that is to say, the total power consumption of the shortest path (most power efficient path) between any two nodes in the final topology should not exceed a constant factor of the power consumption of the shortest path in original network. Given a path $v_1 v_2 \ldots v_h$ connecting two nodes v_1 and v_h, the energy cost of this path is $\sum_{j=1}^{h-1} \|v_j v_{j+1}\|^\beta$. The path with the least energy cost is called the *shortest path* in a graph. Formally speaking, a subgraph H is called a *power spanner* of a graph G if there is a positive real constant ρ such that for any two nodes, the power consumption of the shortest path in H is at most ρ times of the power consumption of the shortest path in G. The constant ρ is called the *power stretch factor*. A *power spanner* is usually energy efficient for routing.

Obviously, for any weighted graph G and a subgraph $H \subseteq G$, we have the following lemma.

Lemma 1 [18]. Subgraph H of a graph G has stretch factor ρ if and only if for any link $uv \in G$, $d_H(u, v) \leq \rho \cdot d_G(u, v)$, where $d_G(u, v)$ is the total power consumption of the shortest path between u and v in G.

Lemma 1 implies that, to generate a power efficient structure, we only need to guarantee that any two adjacent nodes u and v in G are connected by a path in H with energy cost no more than a constant factor of the cost of link uv.

Degree bounded. 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. A common believe in the literature is that small node degree will imply small interference. Although this is recently disproved in [4], we found that in practice our structures with small node degree indeed have small interferences (it is because that our structures often select short links). Structures with a small node degree also have applications in Bluetooth wireless networks. In Bluetooth based wireless ad hoc networks, the *master* node degree is preferred to be less than 7, according to Bluetooth specifications, to maximize the efficiency. In addition, a structure with small degree will improve the overall network throughout [13].

Planar. Many routing algorithms require the planar topology to guarantee the message delivery, such as right hand routing, *Greedy Perimeter Stateless Routing* (GPSR) [10], *Greedy Face Routing* (GFR) [3], *Adaptive Face Routing* (AFR) [14], and *Greedy Other Adaptive Face Routing* (GOAFR) [15].

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 it based only on the information of all nodes within a constant hops of u. More importantly, we expect that the total communication cost of the algorithm is $O(n)$ messages, where each message is $O(\log n)$ bits; 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.

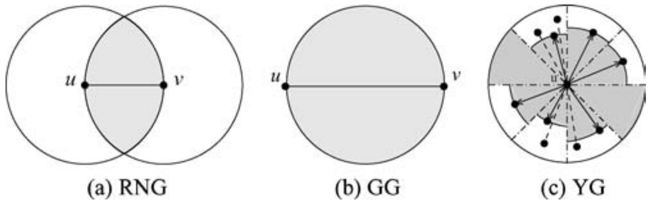


Figure 1. The definitions of *RNG*, *GG*, and *YG*. The shaded area is empty of nodes inside.

2.3. Related works

Several structures (such as relative neighborhood graph RNG, Gabriel graph GG, Yao structure, etc) have been proposed for topology control in wireless ad hoc networks. The *relative neighborhood graph*, denoted by *RNG*(*V*) [23], consists of all edges uv such that the intersection of two circles centered at u and v and with radius $\|uv\|$ do not contain any node w from the set V (see figure 1(a)). The *Gabriel graph* [7] *GG*(*V*) contains edge uv if and only if *disk*(u, v) contains no other points of V, where *disk*(u, v) is the disk with edge uv as a diameter. (See figure 1(b)). Denote *GG*(*UDG*) and *RNG*(*UDG*) as the intersection of *UDG*(*V*) with *GG*(*V*) and *RNG*(*V*) respectively. Both *GG*(*UDG*) and *RNG*(*UDG*) are connected, planar, and contain the Euclidean *minimum spanning tree MST* of *V* if *UDG* is connected. Delaunay triangulation, denoted by *Del*, is also used as an underlying structure by several routing protocols. Here a triangle $\triangle uvw$ belongs to Delaunay triangulation *Del* if its circumcircle does not contain any node inside. Let *Del*(*UDG*) be the set of edges in Delaunay, which are also in UDG. It is well known that $RNG(UDG) \subseteq GG(UDG) \subseteq Del(UDG)$. The structure *Del*(*UDG*) has a bounded length spanning ratio [17]; both *RNG*(*UDG*) and *GG*(*UDG*) are not length spanners; *GG*(*UDG*) is power efficient.

The *Yao graph* [28] with an integer parameter $k > 6$, denoted by $\overrightarrow{YG}_k(UDG)$, is defined as follows. At each node u, any k equally-separated rays originating at u define k cones. In each cone, choose the shortest edge $uv \in UDG(V)$ among all edges emanated from u, if there is any, and add a directed link $\overrightarrow{uv}$. Ties are broken arbitrarily or by ID (see figure 1(c)). The resulting directed graph is called the *Yao graph*. Let $YG_k(UDG)$ be the undirected graph by ignoring the direction of each link in $\overrightarrow{YG}_k(UDG)$. Some researchers used a similar construction named θ-graph [28]. The difference is that it chooses the edge which has the shortest projection on the axis of each cone instead of the shortest edge in each cone.

In [3,10], relative neighborhood graph and Gabriel graph are used as underlying network topologies. However, Bose et al. [1] proved that the length stretch factors of these two graphs are $\Theta(n)$ and $\Theta(\sqrt{n})$ respectively. Actually, they are at most $n-1$ and $\sqrt{n-1}$ [24]. Moreover, in [18], Li et al.

showed that the power stretch factor of RNG is $n-1$ while the power stretch factor of GG is 1. Recently, some researchers [18,27] proposed to construct the wireless network topology based on Yao graph. It is known that the length/power stretch factor and the node out-degree of Yao graph are bounded by some positive constants. But as Li et al. mentioned in [18], all these three graphs can not guarantee node degree bounded (for Yao graph, the node in-degree could be as large as $\Theta(n)$). In [18,19], 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/power stretch factor. However, all these graphs [18,19,27], are not guaranteed to be planar. In [17] Li et al. proposed a planar spanner *localized Delaunay triangulations* (LDel), and in [8] Gao et al. proposed a planar spanner *Restricted Delaunay Graph* for wireless ad hoc networks. Unfortunately, both of them might result in an unbounded node degree.

Bose et al. [2] proposed a centralized method with running time $O(n \log n)$ to build a degree-bounded planar spanner for a two-dimensional point set. They construct 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 [6,11,12]. 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, Wang and Li [25] proposed the first efficient localized algorithm to build a degree-bounded planar spanner $BPS(UDG)$ for wireless ad hoc networks. It has a length spanning ratio $t = \max\{\frac{\pi}{2}, \pi \sin\frac{\alpha}{2} + 1\} \cdot C_{del}(1+\epsilon)$, and each node has degree at most $19 + \lceil \frac{2\pi}{\alpha} \rceil$. Here $0 < \alpha \leq \pi/3$ is an adjustable parameter, and $C_{del} \leq \frac{4\sqrt{3}}{9}\pi$ is the spanning ratio of the Delaunay triangulation. Though their method can achieve all these three desirable features: planar, degree-bounded, and power efficient, the theoretical bound on the node degree of their structure is a large constant. For example, when $\alpha = \pi/6$, the theoretical bound on node degree is 25. In addition, the communication cost of their method can be very high, although it is $O(n)$ theoretically, because it needs to collect the 2-hop information for every wireless node. Even as mentioned in [25], the method by Calinescu [5] to collect 2-hop neighbors information takes $O(n)$ messages, however the hidden constant can be as high as several hundreds. Concerning this large communication cost and the possible large node degree, we propose two communication efficient methods to construct small degree-bounded planar power efficient structures, which are more practical in wireless ad hoc networks. The construction of our second structure only needs at most $3n$ messages, the tradeoff is that theoretically our structures do not have constant length spanning ratio.

3. Proposed approaches

We propose two novel methods to build power efficient planar structures with much less communication costs and lower node degree bounds compared with previously best known planar power efficient structures [25] called *BPS* (see figure 2(b)). Before presenting our methods, we first present a localized construction of Gabriel graph structure for homogeneous wireless ad hoc networks.

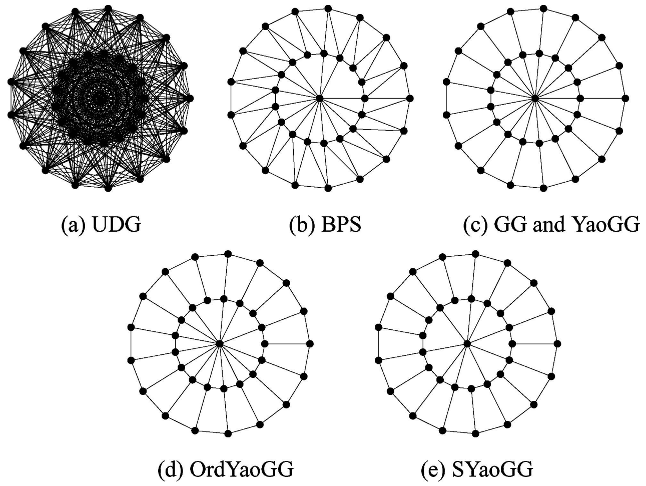


Figure 2. Several planar power spanners on the UDG shown in (a). Here $k = 9$ for Yao related construction.

Algorithm 1. Construct gabriel graph

1. In the beginning, each node u locally broadcasts a message with ID_u, and its position (x_u, y_u) to all nodes in its transmission region. Each node u initiates sets $E_{UDG}(u)$ and $E_{GG}(u)$ to be empty. Here $E_{UDG}(u)$ and $E_{GG}(u)$ are the set of links known by u in UDG and GG respectively.
2. At the same time, each node u processes the incoming messages. Assume that node u gets a message from some new node v, then it adds a link uv to $E_{UDG}(u)$.
   Node u checks whether there is another link $uw \in E_{UDG}(u)$ where $w \in$ disk(u, v). If no such link uw exists, then it adds uv to $E_{GG}(u)$. On the other hand, for any link uw $uw \in E_{GG}(u)$, node u checks whether v $v \in$ disk(u, w), if the condition holds, then u removes link uw from $E_{GG}(u)$.
   Node u repeats this step until no new messages are received.
3. All links uv in $E_{GG}(u)$ are the final links in $GG(UDG)$ incident on u.

We first show that Algorithm 1 builds the structure $GG(UDG)$ correctly. For any link $uv \in GG(UDG)$, clearly, we cannot remove them in Algorithm 1. For a link $uv \notin GG(UDG)$, assume that a node w is inside disk(u,v) and both links uw and wv belong to UDG. If node u gets the message from w first, and then gets the message from v, clearly, uv cannot be added to $E_{GG}(u)$. If node u gets the message from v first, then node u will remove link uv from $E_{GG}(u)$ (if it is there) when u gets the information of node w.

It is not difficult to prove that structure $GG(UDG)$ is connected by induction if UDG is connected. In addition, since we remove a link uv only if there are two links uw and wv with w inside disk(u,v), it is easy to show that the power stretch factor of structure $GG(UDG)$ is exactly 1 [19]. In other words, the minimum power consumption path for any two nodes u and v in UDG is still kept in $GG(UDG)$. Remember that here we assume the power needed to support a link uv is uv is $\|uv\|^\beta$, for $\beta \in [2, 5]$. Notice that, as mentioned in the literature, $GG(UDG)$ is not degree bounded. For example, when all $n-1$ nodes are uniformly distributed on a unit circle with the nth node u as center, the node degree of u is $n-1$. Figure 2(a) shows another example, where $(n-1)/2$ nodes are uniformly distributed on a unit circle, another $(n-1)/2$ nodes are on a half unit circle, and both circles have the nth node u as center. The node degree of center is $(n-1)/2 = O(n)$ as shown in figure 2(c).

The following result is folklore.

Theorem 2 ([17]). $GG(UDG)$ is a planar power spanner, whose power stretch factor is 1.

Hereafter, if it is clear that these structures are constructed on $UDG(V)$, we omit the (UDG) in the representation of all structures. For instance, we will use GG to denote Gabriel Graph instead of $GG(UDG)$.

3.1. Degree(k+5) planar power spanner (OrdYaoGG)

One natural way to construct a degree-bounded planar power spanner is to apply the Yao structure on Gabriel graph. In [19], Li et al. showed that the final structure by directly applying the Yao structure on GG is a planar power spanner, called *YaoGG*, however its in-degree can be as large as $O(n)$, as in the example shown in figure 2(c). In this paper, we present a new method by applying the ordered Yao structures on Gabriel graph to bound node degree. The idea is similar with the method in [25] where they apply Yao structures on the localized Delaunay triangulations to build a degree-bounded planar length spanner based on a locally computed ordering of nodes. The major differences are (1) here we only use 1-hop information instead of two hop information, which reduces communication cost significantly; (2) we use Gabriel graph instead of the localized Delaunay triangulation, which makes the localized method much simpler and more efficient; (3) the method used to bound the degree is also different.

Since Gabriel graph is power efficient, we will then bound the node degree of the Gabriel graph by removing some edges without destroying the power spanner property. We will process the nodes in a special order. When we process a node u, we use the Yao structure to decide which unprocessed neighbors will be selected, while keeping already processed neighbors. Our special order makes sure that when processing a node, it only has at most 5 processed neighbors. The algorithm detail is as follows.

Algorithm 2. Construct degree-(K+5) planar power spanner *OrdYaoGG*

1. First, each node self-constructs the Gabriel graph GG locally based on the strategy described in Algorithm. Let $N_{GG}(u)$ be the neighbors set of node u in GG.
2. Second, each node u decides its order π locally as follows. Two data structures at each node u are used in this algorithm:
   (1) $\pi[\,]$: the list of the local orders of all neighbors of u (including itself) in GG, where each element is initially set as 0, i.e., unordered. Hence $\pi[v]$ denotes the order of node v, which is a neighbor of node u or node u itself.
   (2) $d(u)$: the number of its unordered neighbors known by node u so far, which is initially set as its degree in GG.
   (3) DOQUERY: a flag indicating whether this node need perform a query to its neighbors now. Initially, the flag is set as FALSE if its degree $d(u) > 5$ and TRUE otherwise. Notice that when the node is ordered (i.e., $\pi[u] > 0$), this flag DOQUERY is always set to FALSE.

   The strategy of finding a local ordering is as follows:

   (a) If DOQUERY is true, then node u queries all its unordered neighboring nodes by sending a message QUERY. The query message QUERY contains only the ID of node u.
   (b) When an unordered node v receives a message QUERY from a neighboring node u in GG, it checks whether $d(v) \le 5$ and $ID(v) < ID(u)$. If so, node v replies node u a message FAILEDQUERY with the IDs of itself and u. Otherwise, node v replies node u a message PASSEDQUERY with the IDs of itself and u.
   (c) If node u received a message FAILEDQUERY, node u sets DOQUERY to FALSE. Node u will not perform such query until its degree is decreased later, so there are at most 5 rounds of queries.
   (d) If node u receives message PASSEDQUERY from all its unordered neighbors, node u sets

   $$\pi[u] = \max\{\pi[v] \mid v \in N_{GG}(u)\} + 1,$$

   sets DOQUERY to FALSE, and broadcasts $\pi[u]$ to its neighbors $N_{GG}(u)$ through message MYORDER.
   (e) If node v receives a MYORDER message from its neighbor u in GG saying that $\pi[u] = k$, it records $\pi[u]$ locally, and updates its $d(u) = d(v) - 1$. If $\pi[v] = 0$ and $d(v) \le 5$, then node v sets DOQUERY to TRUE.
   (f) When node u finds that $d(u) = 0$ and $\pi[u] > 0$, it can go to next step to bound its degree in the final structure.
3. All nodes self-form the final topology based on the local ordering π as follows. Initially, all nodes are marked with WHITE color, i.e., unprocessed. Let $N_{OYGG}(u)$ be the set of neighbors of u in the final topology, which is initialized as $N_{GG}(u)$.

(a) If node u is unprocessed (marked WHITE) and has the largest order $\pi[u]$ among all its WHITE neighbors in $N_{GG}(u)$, it divides its transmission region (which is a unit disk centered at the node u) into k equal-sized cones, keeps one nearest WHITE neighbor $v \in N_{OYGG}(u)$ (if available) in each cone and deletes others. Node u marks itself BLACK, i.e., processed, and notifies all nodes in $N_{GG}(u)$ of the deleted edges through a broadcast message UPDATEN. The message UPDATEN includes all unselected neighbors.

(b) Once the node u receives the message UPDATEN for deleting edge from its neighbor v, it deletes the node v from its local list $N_{OYGG}(u)$.

4. When all nodes are processed, all the remaining edges $\{uv | v \in N_{OYGG}(u), \forall v \in GG\}$ form the final network topology *OrdYaoGG*. Each node then can shrink its transmission range as long as it sufficiently reaches its farthest neighbor in the final topology.

Lemma 3. The final topology *OrdYaoGG* is a planar graph, whose node degree is bounded by $k+5$ where $k > 6$ is an adjustable parameter.

Proof: The Yao graph construction does not add any edges to original graphs, on the contrast, it only deletes edges. Hence the planar property is inherited from *GG* graph.

We then show that each node degree is bounded by $k+5$ in *OrdYaoGG*. To prove this, we first review one important property for planar graph: there always exists a node with degree at most 5. Clearly, our local ordering is able to start, since there is at least one node with degree at most 5 initially. Once a node is ordered, the neighboring nodes will update their node degree accordingly. We clearly can repeat this procedure until all nodes are ordered, since the Gabriel graph induced on all unordered nodes is always planar. Let P_u be the neighbors of node u in *GG* that are ordered *after* u. From our processing order of nodes, these nodes will be marked BLACK before node u, i.e., being processed before u. We will then call P_u predecessors of node u. Clearly, in the local ordering π, every node u has at most 5 edges to its predecessors P_u in *GG*, that is to say, before it is marked with BLACK, it has at most 5 processed neighbors.

When node u is processed, it could select at most k other unprocessed neighbors into final structure, thus, its degree is bounded by $k+5$. Once a node is marked with BLACK color, its degree will be kept unchanged according to our algorithm. This finishes our proof. □

In figure 2, we show that *GG* and *YaoGG* cannot bound the node degree, while our structure *OrdYaoGG* is indeed degree-bounded by $k+5 = 14$, here k is set as 9 in our experiment. We then prove that the final structure is also power efficient.

Lemma 4. *OrdYaoGG* is a power spanner of *UDG*, and its power spanning ratio is $\rho = \frac{1}{1-(2\sin\frac{\pi}{k})^\beta}$, where $k > 6$ is an adjustable parameter and $\beta \in [2, 5]$ is a constant depending on the transmission environment.

Proof: Since the *GG* is a power spanner with spanning ratio 1, we only need prove that *OrdYaoGG* is a power spanner of *GG* with spanning ratio $\rho = \frac{1}{1-(2\sin\frac{\pi}{k})^\beta}$. The proof is similar to the proof for Yao on UDG [18] and the later proof of Theorem 7. Due to space limitation, we omit the details here. □

We then analyze the total communication cost of Algorithm 2. (1) Clearly, the first step of building *GG* can be done using only n messages: each message contains the ID and geometry position of a node. (2) The second step of computing local ordering can be done in $21n$ messages: First, an unordered node u sends out at most 5 query messages containing its ID. Each such query message is replied by $d(u)$ neighbors. Since we perform a new query only if $d(u)$ decreases from last failed query, the total messages used for queries is at most $n \cdot \sum_{i=1}^{5}(i+1) = 20n$ messages. Second, an ordered node u sends a message containing its ID and the local ordering π_u computed. The second step can thus be done in at most $21n$ messages. (3) In the third step, a processed node u will inform all its WHITE neighbors v about the deletion of the edge uv from Gabriel Graph (which has at most $3n$ edges). In the final topology OrdYaoGG, at least $n-1$ edges were kept to guarantee the connectivity, thus, the total number of such messages is at most $2n$. In summary, the following lemma directly follows.

Lemma 5. Assuming that both the ID and the geometry position can be represented by $\log n$ bits each, the total number of messages of Algorithm 2 is then at most $24n$, where each message has at most $2\log n$ bits.

Additional communication and computation cost can be saved, if the degree is expected to be bounded by $k+5$ only. The modification is to let all nodes with degree at most $k+5$ be initially marked as BLACK, that is to say, they do not participate in the third step in Algorithm 2.

Remember that the total messages of our Algorithm 2 is bounded by $O(n)$. This implies that the average number of messages per node is a constant, which is verified in our simulations presented later. However, in the worst case, the number of messages sent by some node could be as large as $O(n)$. Algorithm 2 can be modified to further bound the communication cost of each node. During the Yao construction in the third step, instead of using message UPDATEN to delete the unselected links, each node will notify its neighbors of the kept edges. In other words, the message UPDATEN contains the selected neighbor IDs instead of the deleted neighbor IDs. The communication cost of each node can be bounded since at most k neighbors are kept during Yao construction. It is easy to show that each node sends at most 31 messages during constructing GG and computing the local order: at most 5 QUERY messages are sent, and at most 25 PASSEDQUERY or FAILEDQUERY messages are sent. The tradeoff is that the total communication cost could be higher than that used in Algorithm 2 if the final topology is denser.

3.2. Degree-k planar power spanner (SYaoGG)

Algorithm 2 constructs a planar power efficient structure using at most $O(n \log n >)$ bits communications, and the final structure has a theoretical degree bound k+5, where $k > 6$ is a parameter. In this section, we propose a more efficient method with much less communication cost during construction. Notice that, the majority communication cost of Algorithm 2 is spent on computing a local ordering of nodes so a bounded node degree is achieved. Our second method will eliminate this step while still achieving a bounded node degree. We still process the nodes in a local order, which can be obtained easily. Each node u uses the Yao structure to decide which neighbors will be kept: always keep the closest processed neighbor if exists, otherwise keep the closest unprocessed neighbor. Clearly, this will bound the node degree, but, as will see later, it is much tricky to prove the final structure is power efficient. The second method works as follows.

Algorithm 3. Construct degree-k planar power spanner *SYaoGG*

1. First, each node self-constructs the Gabriel graph GG locally based on the strategy described in Algorithm 1.
2. All nodes together self-form the final topology as follows. Initially, each node u is marked with WHITE color, i.e., unprocessed, and initializes $N_{\text{SYGG}}(u)$ as the set of all the neighbor nodes in GG.
   (a) If a WHITE node u has the smallest ID among its WHITE neighbors in GG, it divides its transmission region into k equal-sized cones where $k > 8$ is an adjustable parameter. In each cone, node u checks whether there are some BLACK nodes in $N_{\text{SYGG}}(u)$ within same cone:
      (i) Yes. Node u keeps the closest BLACK neighbor $v \in N_{\text{SYGG}}(u)$ among them and deletes all the other links in the cone;
      (ii) No. Node u keeps a closest WHITE neighbor $v \in N_{\text{SYGG}}(u)$ (if available) among them and deletes all the other links in the cone.

      After processing all k cones, node u marks itself BLACK, i.e. processed, then notifies each deleted neighboring node v in GG by a broadcasting message UPDATEN.
   (b) Once a WHITE node v receives the message UPDATEN from a neighbor u in GG, it checks whether itself is in the nodes set for deleting: if so, it deletes the sending node u from $N_{\text{SYGG}}(v)$, otherwise, marks u as BLACK in its local list $N_{\text{SYGG}}(v)$.
   (c) Once a BLACK node v receives the message UPDATEN from a neighbor belonging to $N_{\text{SYGG}}(v)$, it checks whether itself is in the nodes set for deleting: if so, it deletes the sending node u from $N_{\text{SYGG}}(v)$, otherwise, marks u as BLACK in its local list $N_{\text{SYGG}}(v)$.
3. When all nodes are processed, all selected edges $\{uv | v \in N_{OYGG}(u), \forall v \in \text{GG}\}$ form the final network topology, denoted by *SYaoGG*. Each node then can shrink its transmission range as long as it sufficiently reaches its farthest neighbor in the final topology.

Algorithm 3 further reduces the communication cost during constructing a degree-bounded planar power spanner, because we do not demand the local ordering before construction.

Our analysis of the structure *SYaoGG* relies on the following simple observation.

Lemma 6. In GG graph, if two edges uv and uw emanates from a single node u, then both the angle $\angle uvw$ and $\angle uvw$ must be acute.

Proof: We prove it by inducing contradiction. Suppose the angle $\angle uvw$ is an obtuse angle, then $||wv|| < ||uw||$, hence, all the three edges uv, vw and uw are in the UDG graph. Thus, the circle with diameter uw contains the node v inside. According to the property of GG graph, edge uw can not be kept during GG construction. The contradiction is induced. This finishes the proof. □

Theorem 7. The structure *SYaoGG* is k degree-bounded planar power spanner, whose power stretch factor is at most $\rho = \frac{\sqrt{2}^{\beta}}{1-(2\sqrt{2}\sin\frac{\pi}{k})^{\beta}}$, where $k \geq 9$ is an adjustable parameter and $\beta\ \varepsilon[2,5]$ is a constant factor depending on the communication environment.

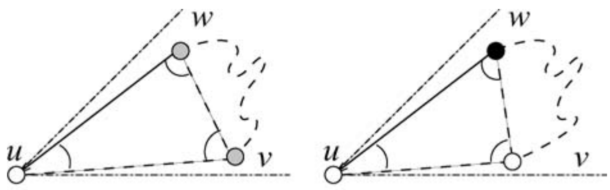
Proof: First, the node degree is obviously bounded by k because each node only keeps one undirected edge in each cone. Figure 2(e) illustrates the *SYaoGG* structure self-constructed on the *UDG* graph in figure 2(a). The node degree is indeed at most k=9.

Second, the graph *SYaoGG* is planar, because the Gabriel graph GG is planar and Algorithm 3 does not add any more edges, thus, the planar property is inherited.

In the following, we show that the structure *SYaoGG* is a power spanner. According to Theorem 2, GG has power spanning ratio 1. Hence, from Lemma 1, it is sufficient to show that for any nodes u and v with an edge $uv \in GG$, there is a path connecting u and v in *SYaoGG* with power cost at most $\rho \cdot ||uv||^{\beta}$.

Given any edge $uv \in GG$, we will construct a path $u \leftrightsquigarrow v$ connecting u and v in *SYaoGG*. If edge uv is kept in the final structure, then $u \leftrightsquigarrow v$ is just uv. Otherwise, assume that uv is removed[2] when processing node u. There must exist a link uw selected by node u in the same cone. Then $u \leftrightsquigarrow v$ is the concatenation of uw with $w \leftrightsquigarrow v$ (see figure 3). Notice that node u is marked as processed in this stage. It is possible that the link uw could then be removed by node w later on since node w is not processed when process node u. If so, we replace link uw by $u \leftrightsquigarrow w$, see figure 4 for illustration, details will be explained later.

[2]Notice that an edge $uv\varepsilon\ GG$ can only be removed while processing its **endpoint** node u or node v.



(a) Both w and v are WHITE or BLACK

(b) w is BLACK and v is WHITE

Figure 3. The link uw is kept in the final structure.

We then prove by induction, on the number of its edges, that the path $u \leftrightsquigarrow v$ has power cost, denoted by $p(u \leftrightsquigarrow v)$, at most $\rho||uv||^{\beta}$.

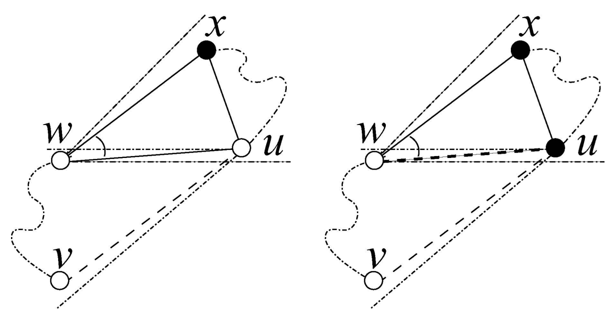
Obviously, if there is only one edge in $u \leftrightsquigarrow v$, $p(u \leftrightsquigarrow v) = ||uv||^{\beta} < \rho||uv||^{\beta}$. Assume that the claim is true for any path with l edges. Then consider a path $u \leftrightsquigarrow v$ with $l+1$ edges, which is the concatenation of edge uw (or path $u \leftrightsquigarrow w$) and the path $w \leftrightsquigarrow v$ with at most l edges.

Without loss of generality, we always assume that the link uv is removed after node u is processed and link uw is selected in the cone. Notice that the link uw could be removed later by node w if w is processed after u, so there are two cases that need to be discussed carefully:

1. The first case is that link uw is kept in the final structure. Remember that, as described in the algorithm, we always select the nearest BLACK neighbor in a cone if it exists; otherwise the nearest WHITE neighbor is selected if it exists.

   Figure 3 illustrates the situations that a WHITE node u starts Yao construction in the cone. Suppose, we delete uv in the cone and choose edge uw, which is also kept in the final structure. Again, there are two subcases that need to be analyzed:

   Subcase 1. $||uw|| \leq ||uv||$. This subcase happens only when both nodes v and w are processed (or unprocessed), and node u deletes link uv since the existence of closer processed (or unprocessed) neighbor w. Figure 3(a) illustrates the situation.



(a) processing u

(b) processing w

Figure 4. Link uv is removed when processing node u (illustrated in the left figure) and link uw is then removed by node w later (illustrated in the right figure).

We bound the length $||wv||$ respecting to $||uv||$. Notice that $||uw|| \leq ||uv||$ and $\angle wuv < \theta = \frac{2\pi}{k}$. The maximum length of vw is achieved when $||uw|| = ||uv||$ because the angle $\angle uwv$ is acute according to Lemma 6. Therefore

$$||wv|| \leq 2\sin\frac{\theta}{2}||uv|| = 2\sin\frac{\pi}{k}||uv||.$$

By induction, we have

$$\begin{aligned} p(u \leftrightsquigarrow v) &= ||uw||^{\beta} + p(w \leftrightsquigarrow v) \\ &\leq ||uw||^{\beta} + \rho||wv||^{\beta} \\ &\leq ||uv||^{\beta} + \rho \cdot \left(2\sin\frac{\pi}{k}\right)^{\beta} ||uv||^{\beta} \\ &\leq \rho||uv||^{\beta}, \end{aligned}$$

when $\rho \geq \frac{1}{1-(2\sin\frac{\pi}{k})^{\beta}}$.

Subcase 2. $||uw|| > ||uv||$ This case happens only when node w is processed while node v is not processed yet, and node u deletes link uv since any processed neighbor has higher priority in our algorithm. Figure 3(b) illustrates the situation.

We bound the length $||wv||$ respecting to $||uw||$. Notice that $||uw|| > ||uv||$ and $\angle wuv < \theta = \frac{2\pi}{k} < \frac{\pi}{4}$ according to Lemma 6. So we have $\frac{\pi}{4} < \angle uwv < \angle uvw < \frac{\pi}{2}$. Consequently, $||uw|| < \frac{\sin\frac{\pi}{2}}{\sin\frac{\pi}{4}}||uv|| = \sqrt{2}||uv||$. The maximum length of vw is achieved when $||uw|| = ||uv||$ because the angle $\angle\, uwv$ is acute. Therefore

$$||wv|| \leq 2\sin\frac{\pi}{k}||uw|| \leq 2\sqrt{2}\sin\frac{\pi}{k}||uv||.$$

By induction, we have

$$\begin{aligned} p(u \leftrightsquigarrow v) &= ||uw||^{\beta} + p(w \leftrightsquigarrow v) \leq ||uw||^{\beta} + \rho||wv||^{\beta} \\ &\leq (\sqrt{2})^{\beta}(1 + \rho(2\sin\frac{\pi}{k})^{\beta})||uv||^{\beta} \leq \rho||uv||^{\beta}, \end{aligned}$$

when $\rho \geq \frac{\sqrt{2}^{\beta}}{1-(2\sqrt{2}\sin\frac{\pi}{k})^{\beta}}$.

2. The second case is that link uw is later removed by node w. We show that the spanning ratio is still kept. Notice that, this case could only succeed *Subcase 1*. The link uw in *Subcase 2*, see figure 3(b), can never be removed in our algorithm, since both node u and w have processed and kept this edge. An edge can only be removed by its endpoints. This is the tricky case in this algorithm.

Figure 4(a) shows the situation that a WHITE node u selects a link uw in a cone, where the neighbor node w is not processed. Figure 4(b) illustrates the scenario when node w processes its neighbors: since it has two processed[3] neighbors

[3] Node x must also be a processed node, otherwise w will definitely select u instead of x according to our rule.

u and x in the cone, it will select the nearest processed neighbor in that cone, which is node x. Observe that after node w decided to keep link wx and remove link uw, the link wx will be kept in the final structure since both end nodes w and x are processed and only an unprocessed node can remove its incident links later. Obviously, from the selection procedure, we know that

$$||uv|| \geq ||uw|| \geq ||wx||.$$

Notice that, both nodes u and x select the node w in one of their cones when they are processed before node w starts its processing. Node w then selects x instead of u because wx is shorter. Consequently, node u does not have any neighbors kept in the node u's cone shown in figure 4(b). This is a sharp contrast to our first structure *OrdYaoGG*, in which every node always keep an edge in each cone if it originally has one neighbor from Gabriel graph. Then the path $v \leftrightsquigarrow u$ connecting nodes u and v is composed of path $v \leftrightsquigarrow w$, link wx and path $x \leftrightsquigarrow u$. The total power cost of the path $v \leftrightsquigarrow u$ is

$$\begin{aligned} p(u \leftrightsquigarrow v) &= ||wx||^\beta + p(w \leftrightsquigarrow v) + p(u \leftrightsquigarrow x) \\ &\leq ||wx||^\beta + \rho||wv||^\beta + \rho||ux||^\beta \\ &\leq ||wx||^\beta + \rho\left(2\sin\frac{\pi}{k}\right)^\beta (||uv||^\beta + ||uw||^\beta) \\ &\leq ||uv||^\beta\left(1 + 2\rho\left(2\sin\frac{\pi}{k}\right)^\beta\right) \\ &\leq \rho||uv||^\beta, \end{aligned}$$

when $\rho \geq \frac{1}{1-2(2\sin\frac{\pi}{k})^\beta}$.

All conditions about p are satisfied when $\rho = \frac{\sqrt{2}^\beta}{1-(2\sqrt{2}\sin\frac{\pi}{k})^\beta}$. This finishes the proof. □

We then analyze the communication cost of Algorithm 3. (1) Clearly, the first step of building *GG* can be done using only n messages: each message contains the ID and geometry position of a node. (2) In the second step of the algorithm, initially, the number of edges in Gabriel Graph is less than $3n$ since it is a planar graph. Clearly, there are at most $2n$ such removed edges since we keep at least n–1 edges from the connectivity of the final structure. Thus the total messages used to inform the deleted edges from *GG* is at most $2n$. Then the following lemma directly follows.

Lemma 8. Assuming that both the ID and the geometry position can be represented by $2\log n$ bits each, the total number of messages by Algorithm 3 is at most $3n$, where each message has at most $2\log n$ bits.

Similarly, if the message UPDATEN contains the selected neighbor IDs instead of the deleted neighbor IDs, then the communication cost of each node also can be bounded by $k+1$ since at most k neighbors will be kept during Yao construction.

Theoretically, compared with *OrdYaoGG*, the topology *SYaoGG* has lower node degree bound while higher power spanning ratio bound. Worth to mention that, our simulation later shows the power spanning ratios of *OrdYaoGG* and *SYaoGG* are very close in practice.

4. Performance evaluation on random networks

We evaluated the performance of our new degree-bounded and planar spanners by conducting simulations on randomly deployed wireless ad hoc networks. In our experiments, we randomly generated a set V of n wireless nodes and $UDG(V)$, then tested the connectivity of $UDG(V)$. If it is connected, we construct different localized topologies on $UDG(V)$, including our new topologies *OrdYaoGG* and *SYaoGG*, some well-known planar spanner topologies *GG* [3,10]. *YaoGG* [19], and *BPS* [25]. Then we measure the sparseness, the power efficiency and the communication cost during construction of these topologies.

In the experimental results presented here, we generated n random wireless nodes in a 20×20 square; the parameter k, i.e., the number of cones, is set to 9 when we construct *BPS*, *OrdYaoGG* and *SYaoGG*; the transmission range is set to 8. We tested all preferred properties described in Section 2.2 of these planar structures by varying node number from 30 to 300, where 100 node sets are generated for each case to smooth the possible peak effects caused by some exception examples. The average and the maximum were computed over all these 100 node sets.

4.1. Power efficiency

The most important design metric of wireless network topology 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? First, we test power stretch factors of all structures. In our simulations, we set power attenuation constant $\beta = 2$. Setting larger β, from our proofs, we expect to see smaller spanning ratios of our structures. In figure 5, we summarize our experimental results of power stretch factors of all these topologies. It shows all of the power stretch factors are small in practice, just around 1.002, except *GG* has power stretch factor 1. In other words, the path remaining in the sparse planar structures can estimate the shortest path in the original communication graph without too higher power consumption. It is not surprising that the average/maximum power stretch factors of *OrdYaoGG* and *SYaoGG* are at the same level of those of *GG* while they are much sparser.

Another interesting thing to notice is that *OrdYaoGG* has smaller power spanning ratio than *YaoGG*, even though *OrdYaoGG* is sparser than *YaoGG* theoretically and practically (Refer figure 7). One reason is that *OrdYaoGG* is more uniform than *YaoGG*. Hence, the proper ordering scheme can conserve more energy.

Notice that after constructing the sparse structures, a node can shrink its transmission energy as long as it is enough to cover the longest adjacent link in the structure. By this way,

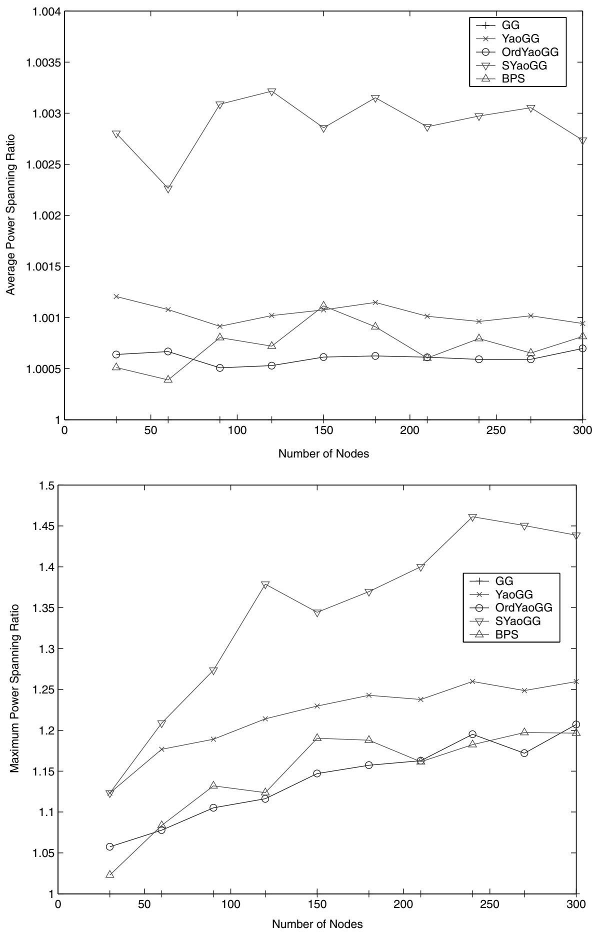


Figure 5. Average and maximum power spanning ratio of different topologies.

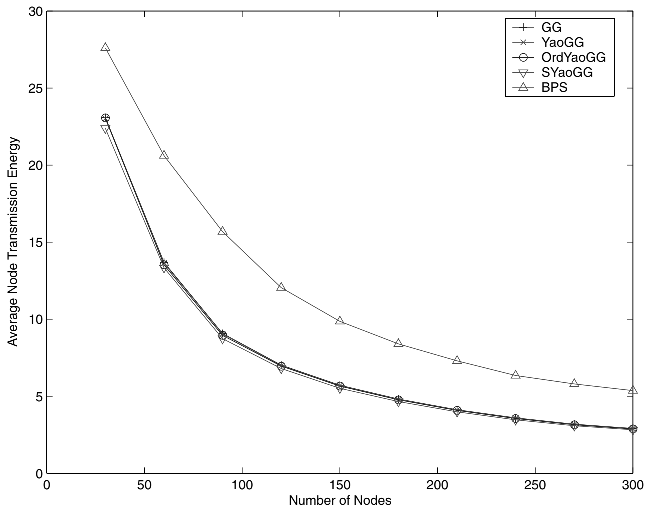


Figure 6. Average and maximum node degree of different topologies.

we define the node transmission power for each node u in a constructed structure as follows. If u has a longest link, say uv, in the structure, then the node transmission energy of u is $\|uv\|^{\beta}$. As expected, Figure 6 shows the average node transmission energy of each topology decreases as the network density increases. The power needed by each node in our new structures *OrdYaoGG* and *SYaoGG* is almost same with that by *GG*, which is much less than its maximum transmission energy (which is 8^{β} here $\beta = 2$ in our experiment). Each node in *BPS* need to set higher transmission energy since it has more neighbors. Specifically, BPS is a supergraph of the Gabriel graph and our new structures are subgraphs of the Gabriel graph.

4.2. Node degree

The node degree is an important performance metric in wireless ad hoc networks, since the degree of each node directly relates to its power consumption and the global network performance.

The average and maximum node degrees of each topology are shown in figure 7. It shows that *OrdYaoGG* and *SYaoGG* have less number of edges (average node degrees) than *YaoGG*, *GG* and *BPS*. In other words, these graphs are sparser. Notice that the node degree of *BPS* is much higher than those of other graphs, since *BPS* uses many edges from *LDel* which is a supergraph (thus much denser than) of *GG*, see figures 2(b) and (c), while all the other structures discussed here are subgraphs of the Gabriel graph. Recall that theoretically, only *BPS*, *OrdYaoGG* and *SYaoGG* have bounded node degree (both for in-degree and out-degree). In [18,19], Li et al. gave an example to show that *RNG*, *GG*, and *LDel* could have large node degree (in-degree for *YG* and *YaoGG*). Notice that, in our experiments, since the wireless nodes are randomly distributed in two dimensional space, it is easy to understand that the maximum node degree of *GG* and *YaoGG* are not as big as the extreme example, however, it can happen. Recall that we proved *OrdYaoGG* and *SYaoGG* have bounded node degree $k{+}5$ and k respectively. In figure 2, we give a special example to show the theoretical node degree bound for *OrdYaoGG* and *SYao*, where two group wireless nodes, with size 17 each, are uniformly distributed on a unit disk and a half-unit disk respectively. Both disks are centered at one node u with $ID = 0$. Figure 2 shows the unit disk graph, which is a complete graph, and other structures built on it. Notice that *GG* and *YaoGG* keep all the links to u in the inner cycle, while *BPS* and *OrdYaoGG* can remove some links to bound node degree, and *SYaoGG* has the best node degree bound $k = 9$. Notice that *BPS* is constructed based on *LDel*, and it also added some edges to keep the length spanner property, so it is the densest among them.

Beside the node degree of all these structures, we are also interested in another kind of node degree, called *physical node degree*. For each node u, it has a longest link, say uv, in a constructed structure. Then the physical degree of u is defined as all nodes w such that $\|uw\| \leq \|uv\|$. This is the total number of nodes that can cause direct interference with u. The average and maximum physical node degrees of each topology are shown in figure 8. They are higher than the node degrees in figure 7 as expected, however they follow the same pattern of curves. Moreover, the possible interference

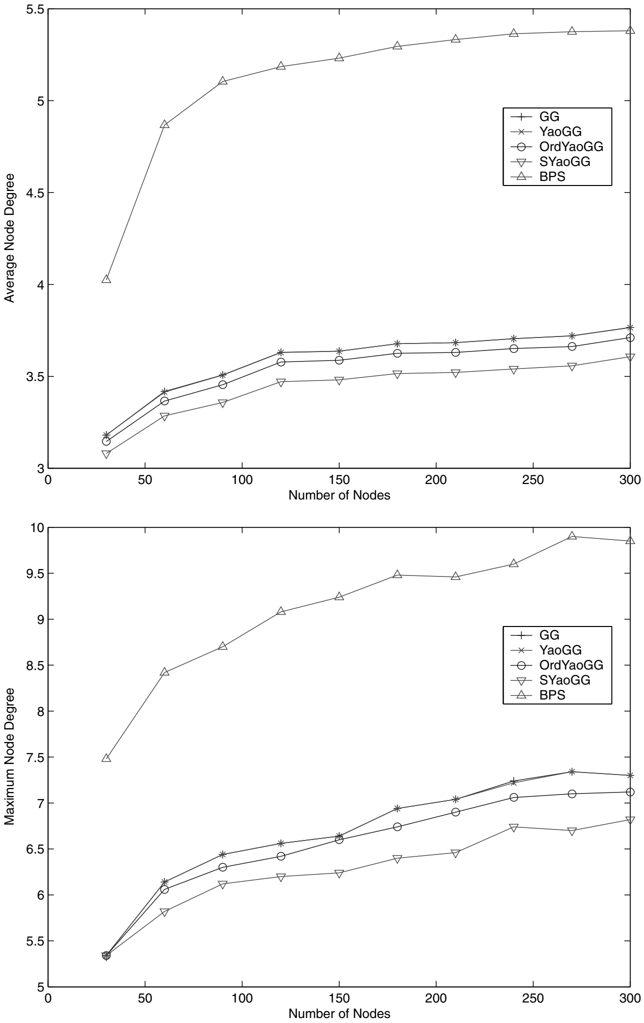


Figure 7. Average and maximum node degree of different topologies.

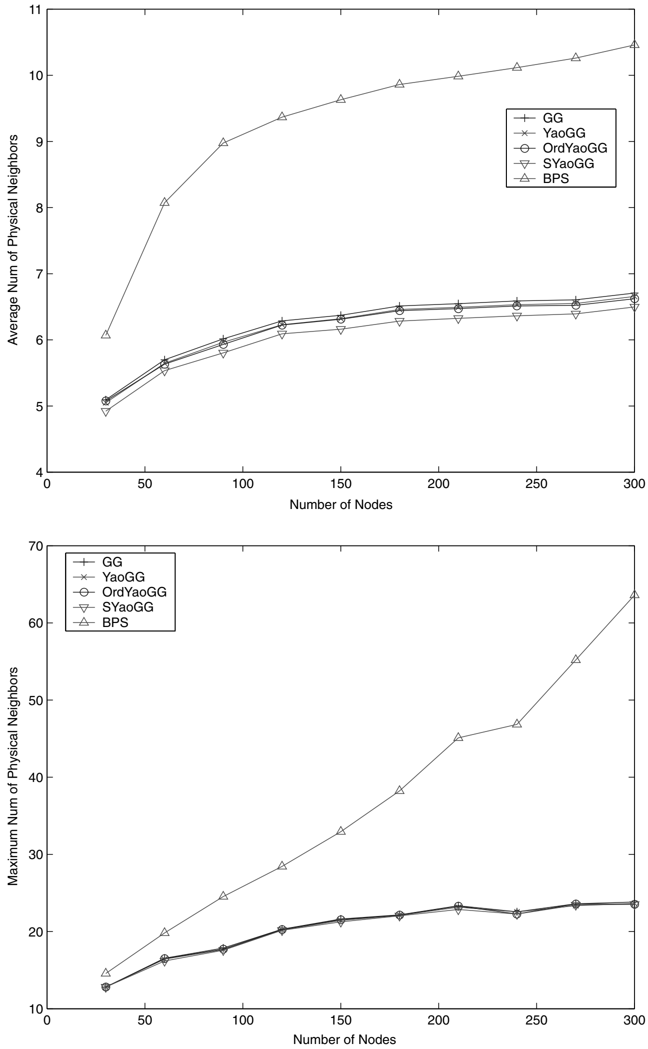


Figure 8. Average and maximum physical node degree of different topologies.

increases slightly while the number of wireless nodes grows. This is tolerable because each node also decreases its transmission range as shown in figure 6 and the average number of actual physical neighbors of a node is around 6 in our simulations.

Notice that for randomly deployed wireless networks, the simulation results do not show big difference between the proposed structures and the structures GG and YaoGG. The reason behind it is that, for randomly deployed networks, the structures GG and YaoGG will have small node degrees with a high probability. Then, the additional steps in our methods to bound the node degree will do nothing in this case. However, our structures can *theoretically* bound the *worst case* performance with only a sufficiently small communication overhead, e.g., the structure SYaoGG can be constructed with at most $3n$ messages.

4.3. Communication cost during construction

In Section 3 we proved that the localized algorithms constructing *OrdYaoGG* and *SYaoGG* use at most $O(n)$ messages. We found that when the number of wireless nodes increases the average messages used by each node for constructing them is still in the same level. Figure 9 summarizes our experimental results of the communication costs in each node during the construction of *OrdYaoGG* and *SYaoGG*. Here we do not compare our communication costs with that of *BPS*, since it uses 2-hop neighbors information and needs to build $LDel^{(2)}$ (*UDG*) which costs much more messages for sure. It is clear that the network becomes more and more dense while the number of wireless nodes increases. However, experiment shows that the localized method does not cost more messages on each node even when the graph becomes denser. An interesting observation is that the average number of messages per node for structures *OrdYaoGG* is around 8 though the theoretical bound is 24. It is reasonable because nodes do not always query 5 times in local ordering in practice. Notice that *SYaoGG* costs much less messages than *OrdYaoGG* does, so it is indeed a very efficient topology construction method. This is expected and consistent with our theoretical analysis.

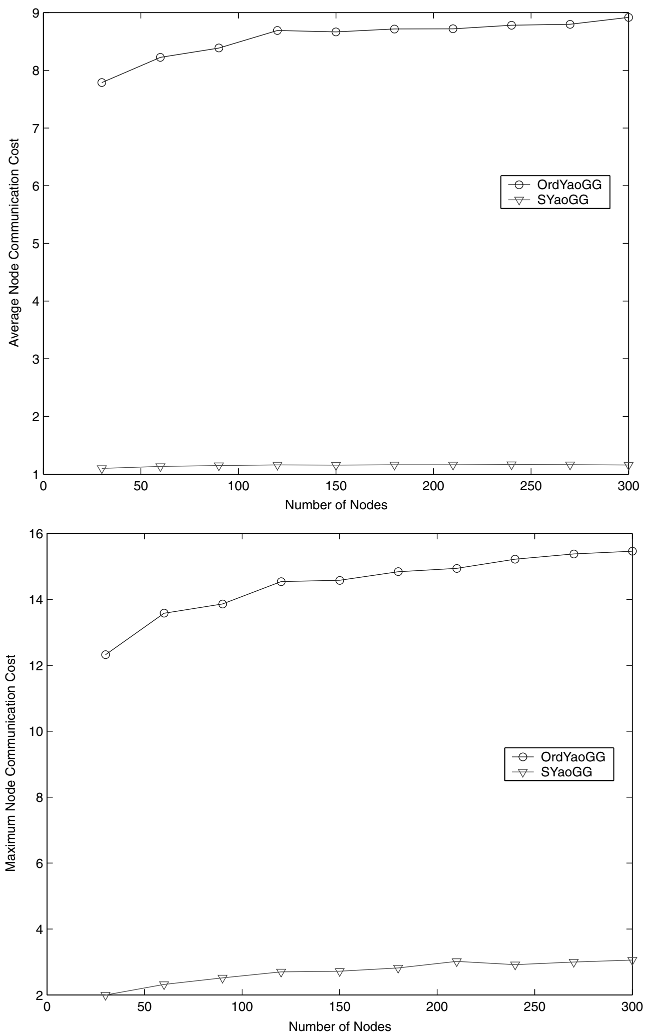


Figure 9. Communication cost during construction of *OrdYaoGG* and *SYaoGG*.

Moreover, simulations results in all charts also show that the performances of our new topologies *OrdYaoGG* and *SYaoGG* are stable when number of nodes changes.

5. Conclusion

We proposed several novel localized algorithms that construct energy efficient routing structures, where each node has a bounded degree and the structures are planar, for wireless ad hoc networks modelled by unit disk graph (UDG). Our first structure has a bounded node degree $k+5$ where integer $k > 6$ is an adjustable parameter; its power stretch factor is no more than $\rho = \frac{1}{1-(2\sin\frac{\pi}{k})^{\beta}}$ it is planar; and it can be constructed locally in 24 n messages, where each message has $O(\log n)$ bits for a wireless network of n nodes.

Our second method improves the degree bound to k, and keeps all other properties, except that the theoretical power spanning ratio is relaxed to $\rho = \frac{\sqrt{2}^{\beta}}{1-(2\sqrt{2}\sin\frac{\pi}{k})^{\beta}}$, where $k > 8$ is an adjustable parameter. We showed that the second structure can be constructed using at most $3n$ messages, where each message has $O(\log n)$ bits.

We conducted extensive simulations to study these new sparse network topologies and compared them with previously known efficient structures. Theoretical results are corroborated by the simulations.

Notice that the bounded node degree of a structure cannot guarantee that the structure has a small link interference. We would like to study how to construct efficient topologies with small interferences while being power efficient and having a bounded node degee.

References

[1] 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)* (2002a).

[2] P. Bose, J. Gudmundsson and M. Smid, Constructing plane spanners of bounded degree and low weight, in: *Proceedings of European Symposium of Algorithms* (2002b).

[3] 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. *3rd Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, (1999) pp. 48–55.

[4] M. Burkhart, P. V. Rickenbach, R. Wattenhofer and A. Zollinger, Does topology control reduce interference, in: *ACM Int. Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc)* (2004).

[5] G. Călinescu, Computing 2-Hop neighborhoods in ad hoc wireless networks, in: *AD-HOC Networks and Wireless (AdHoc-Now)* (2003).

[6] D. Dobkin, S. Friedman and K. Supowit, Delaunay Graphs are Almost as Good as Complete Graphs, Discr. Comp. Geom. (1990) 399–407.

[7] K. Gabriel and R. Sokal, A new statistical approach to geographic variation analysis, Systematic Zoology 18 (1969) 259–278.

[8] 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)* (2001).

[9] M. Grünewald, T. Lukovszki, C. Schindelhauer and K. Volbert, Distributed maintenance of resource efficient wireless network topologies, in: *Proceedings of the 8th International Euro-Par Conference on Parallel Processing* (2002) pp. 935–946.

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

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

[12] J.M. Keil and C.A. Gutwin, Classes of graphs which approximate the complete Euclidean graph, Discr. Comp. Geom. 7 (1992) 13–28.

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

[14] F. Kuhn, R. Wattenhofer and A. Zollinger, Asymptotically Optimal Geometric Mobile Ad-Hoc Routing, in: *International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM)* (2002).

[15] F. Kuhn, R. Wattenhofer and A. Zollinger, Worst-case optimal and average-case efficient geometric ad-hoc routing, in: *ACM Int. Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc)* (2003).

[16] 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: *PODC:ACM Symposium on Principle of Distributed Computing* (2001a).

[17] X.-Y. Li, G. Calinescu and P.-J. Wan, Distributed construction of planar spanner and routing for ad hoc wireless networks, in: *21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, (2002a) Vol. 3.

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

[19] 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)*, (2002b).

[20] T. Lukovszki, New results on geometric spanners and their applications. Ph.D. thesis, University of Paderborn, (1999).

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

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

[22] G.T. Toussaint, The relative neighborhood graph of a finite planar set. Pattern Recognition 12(4) (1980) 261–268.

[24] W. Wang, X.-Y. Li, K. Moaveninejad, Y. Wang and W.-Z. Song, The spanning ratios of *beta*-skeleton, in: *Canadian Conference on Computational Geometry (CCCG)* (2003).

[25] Y. Wang and X.-Y. Li, Efficient Construction of bounded degree and planar spanner for wireless networks, in: *ACM DIALM-POMC Joint Workshop on Foundations of Mobile Computing* (2003).

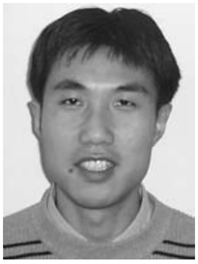
[26] Y. Wang, X.-Y. Li and O. Frieder, Distributed spanner with bounded degree for wireless networks, in: *International Journal of Foundations of Computer Science* (2002).

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

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

Wen-Zhan Song received Ph.D. from Illinois Institute of Technology in 2005, BS and MS from Nanjing University of Science and Technology in 1997 and 2000. He is currently an assistant professor in Washington State University. His current research interest is mainly focus on network protocol and algorithm design, especially in wireless networks, sensor networks and Peer-to-Peer networks. He is a member of the IEEE.
E-mail: song@vancouver.wsu.edu

Yu Wang received the Ph.D. degree in computer science from Illinois Institute of Technology in 2004, the BEng degree and the MEng degree in computer science from Tsinghua University, China, in 1998 and 2000. He has been an assistant professor of computer science at the Univeristy of North Carolina at Charlotte since 2004. His current research interests include wireless networks, mobile computing, algorithm design, and artificial intelligence. He is a member of the ACM, IEEE, and IEEE Communication Society.
E-mail: wangyu@ieee.org



Xiang-Yang Li has been an Assistant Professor of Computer Science at the Illinois Institute of Technology since 2000. He hold MS (2000) and PhD (2001) degree at Computer Science from University of Illinois at Urbana-Champaign. He received his Bachelor degree at Computer Science and Bachelor degree at Business Management from Tsinghua University, P.R. China in 1995. His research interests span the computational geometry, wireless ad hoc networks, game theory, cryptography and network security. He is a Member of the ACM, IEEE, and IEEE Communication Society.
E-mail: xli@cs.iit.edu

Ophir Frieder is the IITRI Professor of Computer Science at the Illinois Institute of Technology. His research interests span the general area of distributed information systems. He is a Member of ACM and a Fellow of the IEEE.
E-mail: ophir@cs.iit.edu