Wireless Sensor Networks and Computational Geometry

Xiang-Yang Li [*] Yu Wang[*]

August 13, 2003

1 Introduction

Wireless Sensor Networks Due to its potential applications in various situations such as battlefield, emergency relief, environment monitoring, and so on, wireless sensor networks [50, 75, 118, 130] have recently emerged as a premier research topic. Sensor networks consist of a set of sensor nodes which are spread over a geographical area. These nodes are able to perform processing as well as sensing and are additionally capable of communicating with each other by means of a wireless ad hoc network. With coordination among these sensor nodes, the network together will achieve a larger sensing task both in urban environments and in inhospitable terrain. The sheer numbers of these sensors and the expected dynamics in these environments present unique challenges in the design of wireless sensor networks. Many excellent researches have been conducted to study problems in this new field [50, 69, 75, 118, 119, 130].

In this chapter, we consider a wireless sensor network consisting of a set V of n wireless sensor nodes distributed in a two-dimensional plane. Each wireless sensor node has an omni-directional antenna. This is attractive because a single transmission of a node can be received by many nodes within its vicinity which, we assume, is a disk centered at the node. We call the radius of this disk the *transmission range* of this sensor node. In other words, node v can receive the signal from node u if node v is within the transmission range of the sender u. Otherwise, two nodes communicate through multi-hop wireless links by using intermediate nodes to relay the message. Consequently, each node in the sensor network also acts as a router, forwarding data packets for other nodes. By a proper scaling, we assume that all nodes have the maximum transmission range equal to one unit. These wireless sensor nodes define a *unit disk graph* $\mathrm{UDG}(V)$ in which there is an edge between two nodes if and only if their Euclidean distance is at most one.

In addition, we assume that each node has a low-power Global Position System (GPS) receiver, which provides the position information of the node itself. If GPS is not available, the distance between neighboring nodes can be estimated on the basis of incoming signal strengths. Relative co-ordinates of neighboring nodes can be obtained by exchanging such information between neighbors [26]. With the position information, we can apply computational geometry techniques to solve some challenging questions in sensor networks.

Computational Geometry Computational geometry emerged from the field of algorithms design and analysis in the late 70s. It studies various problems [49, 58, 131] from computer graphics, geographic information system, robotics, scientific computing, wireless networks recently, and others, in which geometric algorithms could play some fundamental roles. Most geometric algorithms are designed for studying the structural properties, searching, inclusion or exclusion relations, of a set of points, a set of hyperplanes, or both. For example, the structural properties include the convex hull, intersections, hyperplane arrangement, triangulation (Delaunay, regular, and so on), Voronoi diagram, and so on. The query operations often include point location, range searching (orthogonal, unbounded, or some variations) and so on.

In this chapter, we concentrate on how to apply some structural properties of a point set for wireless sensor networks as we treat wireless sensor devices as two-dimensional points.

[*] Department of Computer Science, Illinois Institute of Technology, 10 W. 31st Street, Chicago, IL 60616. Email: `xli@cs.iit.edu`, `wangyu1@iit.edu`

Networking and Routing It is common to separate the network design problem from the management and control of the network in the communication network literature. The separation is very convenient and helps to significantly simplify these two tasks, which are already very complex on its own. Nevertheless, there is a price to be paid for this modularity as the decisions made at the network design phase may strongly affect the network management and control phase. In particular, if the issue of designing efficient routing schemes is not taken into account by the network designers, then the constructed network might not suited for supporting a good routing scheme. Wireless sensor network needs some special treatment as it intrinsically has its own special characteristics and some unavoidable limitations compared with traditional wired networks. Wireless sensor nodes are often powered by batteries only and they often have limited memories. Therefore, it is more challenging to design a network topology for wireless sensor networks, which is suitable for designing an efficient routing scheme to save energy and storage memory consumption, than the traditional wired networks. Also since there maybe several thousand sensors which move often, the construction of the network topology should be easy to operated and updated in a dynamic way.

In technical terms, the question we deal with is therefore whether it is possible (if possible, then how) to design a network, which is a subgraph of the unit disk graph, such that it can be constructed or updated efficiently, and ensures both attractive network features such as bounded node degree, low-stretch factor, and linear number of links, and attractive routing schemes such as localized routing with guaranteed performances.

Topology Control The size of the unit disk graph could be as large as the square order of the number of network nodes. So we want to construct a subgraph of the unit disk graph $\mathrm{UDG}(V)$, which is sparse, can be constructed locally in an efficient way, and is still relatively good compared with the original unit disk graph for routes' quality.

Unlike the wired networks that typically have fixed network topologies, each node in a sensor network can potentially change the network topology by adjusting its transmission range and/or selecting specific nodes to forward its messages, thus, controlling its set of neighbors. The primary goal of topology control in wireless sensor networks is to maintain network connectivity, optimize network lifetime and throughput, and make it possible to design power-efficient routing. Not every connected subgraph of the unit disk graph plays the same important role in network designing. One of the perceptible requirements of topology control is to construct a subgraph such that the shortest path connecting any two nodes in the subgraph is not much longer than the shortest path connecting them in the original unit disk graph. This aspect of path quality is captured by the *stretch factor* of the subgraph. A subgraph with constant stretch factor is often called a *spanner* and a spanner is called a *sparse spanner* if it has only a linear number of links. In this chapter, we review and study how to construct a spanner (a sparse network topology) efficiently for a set of static sensor nodes.

The other imperative requirement for network topology control in sensor networks is the fault tolerance. To guarantee a good fault tolerance, the underlying network structure must be k-connected for some $k > 1$, i.e., given any pair of wireless sensor nodes, there need to be at least k disjoint paths to connect them.

Restricting the size of the network has been found to be extremely important in reducing the amount of routing information. The notion of establishing a subset of nodes which perform the routing has been proposed in many routing algorithms [44, 140, 146, 163]. These methods often construct a virtual backbone by using the connected dominating set [4, 150, 154], which is often constructed from dominating set or maximal independent set.

Routing Many routing algorithms were proposed recently for wireless ad hoc networks. Most of them can be used in wireless sensor networks. The routing protocols proposed may be categorized as table-driven protocols or demand-driven protocols. A good survey may be found in [136].

Table-driven routing protocols maintain up-to-date routing information between every pair of nodes. The changes to the topology are maintained by propagating updates of the topology throughout the network. Destination-sequenced Distance-Vector Routing (DSDV) [127] and Zone-Routing Protocol (ZRP) [74, 166] are two of the table driven protocols proposed recently. The mobility nature of the wireless sensor networks prevent these table-driven routing protocols from being widely used in large scale wireless ad hoc networks. Thus, on-demand routing protocols are preferred.

Source-initiated on-demand routing creates routes only when desired by the source node. The methodologies that have been proposed include the Ad-Hoc On-Demand Distance Vector Routing (AODV) [128], the Dynamic Source Routing (DSR) [23], and the Temporarily Ordered Routing Algorithm (TORA) [122]. In addition, the Associativity Based Routing (ABR) [147] and Signal Stability Routing (SSR) use various criteria for selecting routes.

Introducing a hierarchical structure into routing have also been used in many protocols such as the Clusterhead Gateway Switch Routing (CGSR) [33], the Fisheye Routing [123, 124], and the Hierarchical State Routing [70]. Dominating set based methods were also adopted by several researchers [44, 146, 163]. To facilitate this, several methods [4, 111, 150, 161] were proposed to approximate the minimum dominating set or the minimum connected dominating set problems in centralized and/or distributed ways.

Route discovery can be very expensive in communication costs, thus reducing the response time of the network. On the other hand, explicit route maintenance can be even more costly in the explicit communication of substantial routing information and the usage of scarcity memory of wireless sensor nodes. The geometric nature of the multi-hop wireless sensor networks allows a promising idea: localized routing protocols.

Localized routing does not require the nodes to maintain routing tables, a distinct advantage given the scarce storage resources and the relatively low computational power available to the wireless nodes. More importantly, given the numerous changes in topology expected in sensor networks, no re-computation of the routing tables is needed and therefore we expect a significant reduction in the overhead. Thus localized routing is scalable. Localized routing is also uniform, in the sense that all the nodes execute the same protocol when deciding to which other node to forward a packet.

But localized routing is challenging to design, as even guaranteeing the successful arrival at the destination of the packet is a non-trivial task. This task was successfully solved by Bose et al. [20] (see also [78]) thus opening the way for a second stage of research, focusing on improving the *efficiency* of localized routings. Localized routing also has no built-in mechanism to avoid congestion by overloading nodes. Mauve *et al.* [117] conducted an excellent survey of position-based localized routing protocols.

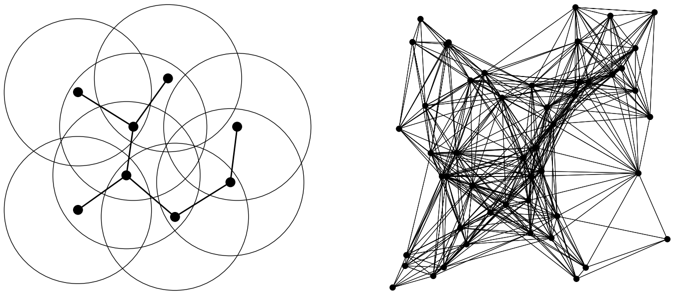
Organization The rest of the chapter is organized as follows. In Section 2, we first review some definitions necessary for more detailed review of current progress of applying computational geometry techniques to wireless snesor networks. Specifically, we specify how the sensor network is modeled in this chapter, review some geometry structures, define the graph spanners, and introduce the localized algorithm concept. In Section 3, we review in detail the geometry structures that are suitable for the topology control in wireless sensor networks, especially the structures with bounded stretch factor, or with bounded node degree, or planar structures. Then we briefly discuss the fault tolerant and interference problems in topology control. We also review the current status of controlling the transmission power so the total or the maximum transmission power is minimized without sacrificing the network connectivity. State of the art of constructing virtual backbone for wireless networks is reviewed. As there are many heuristics proposed in this area, we concentrate on the ones that have theoretic performance guarantees or are popular. After reviewing the geometric structures, we review the so called localized routing methods in Section 4. Many routing algorithms were proposed in the literature. We concentrate on the localized routing protocols as they utilize the geometry nature of the wireless sensor networks. Location service protocols are also discussed. Section 5 reviews the broadcasting protocols that apply the geometry nature to guarantee the performance. Finally, we conclude the chapter in Section 6 by pointing out some possible future research questions.

2 Preliminaries

In this section, we first review some definitions and concepts necessary for later discussion in this chapter. We specify how the sensor network is modeled in a geometry view, review some well known geometry structures, and define spanners and low-weight graphs.

2.1 Unit Disk Graph

We consider a sensor network consisting of a set V of sensor nodes distributed in a two-dimensional plane. By a proper scaling, we assume that all nodes have the maximum transmission ranges equal to one unit. These sensor 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. See Figure 1 (a). Hereafter, we always assume that $UDG(V)$ is a connected graph. Given a set of points uniformly and randomly distributed in an area if the transmission range satisfies some value then the $UDG(V)$ is connected with high probability [63, 134, 137]. We call all nodes within a constant k hops of a node u in the unit disk graph $UDG(V)$ as the ***k-local nodes*** of u, denoted by $N_k(V)$. Usually, here the constant k is 1 or 2. The size of the unit disk graph could be as large as the square order of the number of sensor nodes, such as Figure 1 (b). So in topology contorl which will be discussed in next section, we try to construct a subgraph (spanner) for the unit disk graph $UDG(V)$, and the spanner is sparse and can be constructed locally in an efficient way.



(a) (b)

Figure 1: Examples of unit disk graphs.

2.2 Power-Attenuation Model

Energy conservation is a critical issue in sensor network for the node and network life, as the nodes are powered by batteries only. Each sensor node typically has a portable set with transmission and reception processing capabilities. To transmit a signal from a node to the other node, the power consumed by these two nodes consists of the following three parts. First, the source node needs to consume some power to prepare the signal. Second, in the most common power-attenuation model, the power needed to support a link uv is $\|uv\|^\beta$, where $\|uv\|$ is the Euclidean distance between u and v, β is a real constant between 2 and 5 dependent on the transmission environment. This power consumption is typically called *path loss*. Finally, when a node receives the signal, it needs consume some power to receive, store and then process that signal. For simplicity, this overhead cost can be integrated into one cost, which is almost the same for all nodes. Thus, we will use c to denote such constant overhead. In most results surveyed here, it is assumed that $c = 0$, i.e., the path loss is the major part of power consumption to transmit signals. The power cost $p(e)$ of a link $e = uv$ is then defined as the power consumed for transmitting signal from u to node v.

2.3 Spanners

Spanners have been studied intensively in recent years [7, 8, 9, 17, 28, 45, 76, 89, 164]. Let $G = (V, E)$ be a n-vertex connected weighted graph. The distance in G between two vertices $u, v \in V$ is the total weight (length) of the shortest path between u and v and is denoted by $d_G(u, v)$. A subgraph $H = (V, E')$, where $E' \subseteq E$, is a *t-spanner* of G if for every $u, v \in V$, $d_H(u, v) \leq t \cdot d_G(u, v)$. The value of t is called the *stretch factor*.

Spanners for Euclidean graphs is called *geometric spanners* or *Euclidean spanners*. It means the distance $d_G(u, v)$ in graph G between u and v is the Euclidean distance between vertices u and v. All previous algorithms that construct a t-spanner of the Euclidean complete graph $K(V)$ in computational geometry are centralized methods. The rapid development of the wireless communication presents a new challenge for algorithm designing and analysis. Distributed algorithms are favored than the more traditional centralized algorithms.

Consider any unicast $\Pi(u, v)$ in G (could be directed) from a node $u \in V$ to another node $v \in V$:

$$\Pi(u, v) = v_0 v_1 \cdots v_{h-1} v_h, \text{ where } u = v_0,\ v = v_h.$$

Here h is the number of hops of the path Π. The total *transmission power* $p(\Pi)$ consumed by this path Π is defined as $p(\Pi) = \sum_{i=1}^{h} \|v_{i-1}v_i\|^{\beta}$. Let $p_G(u, v)$ be the least energy consumed by all paths connecting nodes u and v in G. Let H be a subgraph of G. The *power stretch factor* of the graph H with respect to G is then defined as

$$\rho_H(G) = \max_{u,v \in V} \frac{p_H(u, v)}{p_G(u, v)}$$

If G is a unit disk graph, we use $\rho_H(V)$ instead of $\rho_H(G)$. For any positive integer n, let

$$\rho_H(n) = \sup_{|V|=n} \rho_H(V).$$

Similarly, we define the length stretch factors $\ell_H(G)$ and $\ell_H(n)$. When the graph H is clear from the context, it is dropped from notations.

It was proved in [102] that, for a constant δ, $\rho_H(G) \leq \delta$ iff for any link v_iv_j in graph G but not in H, $p_H(v_i, v_j) \leq \delta\|v_iv_j\|^{\beta}$. It is then sufficient to analyze the power stretch factor of H for each link in G but not in H. It is not difficult to show that, for any $H \subseteq G$ with a length stretch factor δ, its power stretch factor is at most δ^{β} for any graph G. In particular, a graph with a constant bounded length stretch factor must also have a constant bounded power stretch factor, but the reverse is not true. Finally, the power stretch factor has the following monotonic property: If $H_1 \subset H_2 \subset G$ then the power stretch factors of H_1 and H_2 satisfy $\rho_{H_1}(G) \geq \rho_{H_2}(G)$.

2.4 Low Weight Structures

The power stretch factor we previously discussed is defined for the unicasting communications. However, in practice, we also have to consider the broadcast or multicast communications. Wan *et al.*[151] showed that the minimum energy cost of broadcasting or multicasting is related to the total energy cost of all links in the Euclidean minimum spanning tree MST. They proved that a broadcasting method based on the Euclidean minimum spanning tree rooted at the sender uses energy no more than 12 times the minimum energy cost of any broadcasting scheme. Therefore, we want the network topology is a low weight structure. Given a structure G over a set of points, let $\omega(G)$ be the total length of the links in G and $\omega_{\beta}(G)$ be the total power needed to support all links in G, i.e., $\omega_{\beta}(G) = \sum_{uv \in G} \|uv\|^{\beta}$. Then, a structure G is called *low weight* if $\omega(G)$ is within a constant factor of $\omega(MST)$.

2.5 Geometry Structures

Several geometrical structures have been studied recently both by computational geometry scientists and network engineers. Here we review the definitions of some of them which could be used in the wireless sensor networking applications. Let $G = (V, E)$ be a geometric graph defined on vertex set V with edge set E.

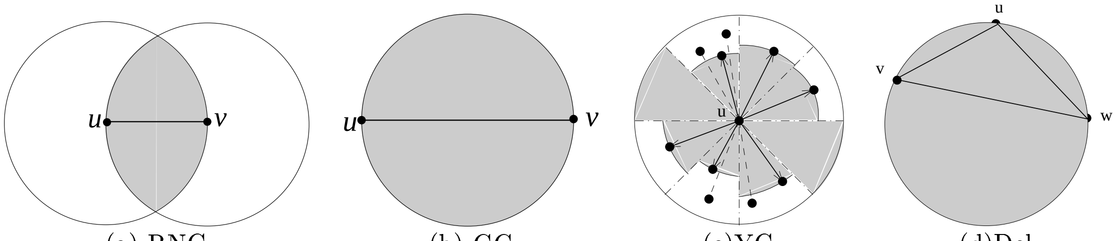
2.5.1 Minimum spanning tree, relative neighborhood graph and Gabriel graph

The *minimum spanning tree* of G, denoted by MST(G), is the tree belong to E that connects all nodes and whose total edge length is minimized. MST(G) is obviously one of the sparsest connected subgraphs, but its stretch factor can be as large as $n - 1$.

The *relative neighborhood graph*, denoted by RNG(G), is a geometric concept proposed by Toussaint [148]. It consists of all edges $uv \in E$ such that there is no point $w \in V$ with edges uw and wv in E satisfying $\|uw\| < \|uv\|$ and $\|wv\| < \|uv\|$. See Figure 2 (a) for an illustration. Notice if G is a directed graph, then edges uw and wv also are directed in the above definition, i.e., we have $\overrightarrow{uw}$ and $\overrightarrow{wv}$ instead of uw and wv.

Let $disk(u, v)$ be the disk with diameter uv. Then, the *Gabriel graph* [53] GG(G) contains an edge uv from G if and only if $disk(u, v)$ contains no other vertex $w \in V$ such that there exist edges uw and wv from G. See Figure 2 (b) for an illustration. Same to the definition of RNG(G), if G is a directed graph, then edges uw and wv also are directed in the above definition of GG(G). GG(G) is a planar graph (that is, no two edges cross each other) if G is the complete graph or UDG. It is easy to show that RNG(G) is a subgraph of the Gabriel graph GG(G). For an undirected and connected graph G, both GG(G) and RNG(G) are connected and contain the minimum spanning tree of G.

Gabriel graph was used as a planar subgraph in the Face routing protocol [20, 47, 143] and the GPSR routing protocol [78] that guarantee the delivery of the packet. Relative neighborhood graph RNG was used for efficient broadcasting (minimizing the number of retransmissions) in one-to-one broadcasting model in [139].

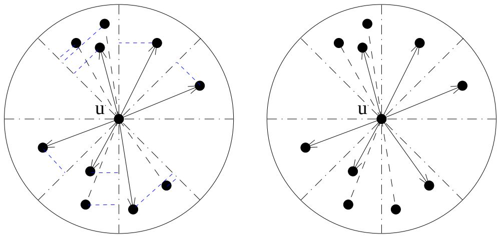


(a) RNG (b) GG (c)YG (d)Del

Figure 2: The definitions of RNG, GG, YG, and Del on point set. The shaded area is empty of nodes inside. (a): The lune using uv is empty for RNG. (b): The diametric circle using uv is empty for GG. (c): The shortest edge in each cone is added as a neighbor of u for Yao. (d): The circumcircle of uvw is empty for Del.

2.5.2 Yao graph and θ-graph

The *Yao graph* [164] with an integer parameter $k \geq 6$, denoted by $\overrightarrow{YG}_k(G)$, is defined as follows. At each node u, any k equally-separated rays originated at u define k cones. In each cone, choose the shortest edge uv among all edges from u, if there is any, and add a directed link $\overrightarrow{uv}$. Ties are broken arbitrarily. The resulting directed graph is called the Yao graph. See Figure 2 (c) for an illustration. Let $YG_k(G)$ be the undirected graph by ignoring the direction of each link in $\overrightarrow{YG}_k(G)$. If we add the link $\overrightarrow{vu}$ instead of the link $\overrightarrow{uv}$, the graph is denoted by $\overleftarrow{YG}_k(G)$, which is called the *reverse* of the Yao graph. Some researchers used a similar construction named θ-graph [80, 81, 114], the difference is that, in each cone, it chooses the edge which has the shortest projection on the axis of the cone instead of the shortest edge. (See Figture 3.) Here the axis of a cone is the angular bisector of the cone. For more detail, please refer to [80, 81, 114].



(a) θ-graph (b) Yao Graph

Figure 3: Illustrations of the difference between θ-graph and Yao graph

Notice all these definitions are exactly the conventional definitions [52, 79, 98, 164] when graph G is the completed Euclidean graph $K(V)$. We will use MST(V), RNG(V), GG(V), and Yao(V) to denote the corresponding resulting graph if G is the complete graph $K(V)$.

2.5.3 Delaunay triangulation and Voronoi diagram

We continue with the definition of Delaunay triangulation. Assume that there are no four vertices of V that are co-circular. A triangulation of V is a *Delaunay triangulation*, denoted by *Del*(V), if the circumcircle of each of its triangles does not contain any other vertices of V in its interior. A triangle is called the *Delaunay triangle* if its circumcircle is empty of vertices of V. See Figure 2 (d) for an illustration. The *Voronoi region*, denoted by $Vor(p)$, of a vertex $p \in V$ is a collection of two dimensional points such that every point is closer to p than to any other vertex of V. The *Voronoi diagram* for V is the union of all Voronoi regions $Vor(p)$, where $p \in V$. The Delaunay triangulation $Del(V)$ is also the dual of the Voronoi diagram: two vertices p and q are connected in $Del(V)$ if and only if $Vor(p)$ and $Vor(q)$ share a common boundary. The shared boundary of two Voronoi regions $Vor(p)$ and $Vor(q)$ is on the perpendicular bisector line of segment pq. The boundary segment of a Voronoi region is called the *Voronoi edge.* The intersection point of two Voronoi edge is called the *Voronoi vertex.* The Voronoi vertex is the circumcenter of some Delaunay triangle. See Figure 4 for an illustration of the relation between $Vor(V)$ and $Del(V)$.

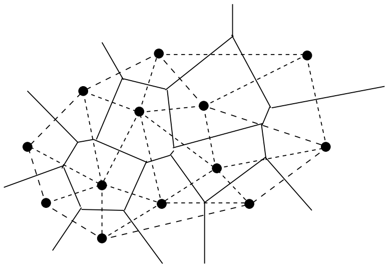


Figure 4: The Voronoi diagram and Delaunay triangulation of a set of two dimensional nodes. The Delaunay triangulation is represented by dash lines.

2.5.4 Connected dominating set and independent set

Besides these geometric structures, some graph notations will also be used in this chapter. A subset S of V is a *dominating set* if each node u in V is either in S or is adjacent to some node v in S. Nodes from S are called dominators, while nodes not is S are called dominatees. A subset C of V is a *connected dominating set* (CDS) if C is a dominating set and C induces a connected subgraph. Consequently, the nodes in C can communicate with each other without using nodes in $V - C$. A dominating set with minimum cardinality is called *minimum dominating set*, denoted by MDS. A connected dominating set with minimum cardinality is denoted by *minimum connected dominating set* (MCDS).

A subset of vertices in a graph G is an *independent set* if for any pair of vertices, there is no edge between them. It is a *maximal independent set* if no more vertices can be added to it to generate a larger independent set. It is a *maximum independent set* (MIS) if no other independent set has more vertices.

2.6 Localized Algorithms

In sensor networks, the large numbers of sensors and the expected dynamics present unique challenges in the design of sensor networks. We believe there are significant robustness and scalability advantages to designing applications using *localized algorithms* - where sensors only interact with other sensors in a restricted vicinity, but nevertheless collectively achieve a desired global objective (such as spanner or low weight). In [50], the authors gave two attractive properties of localized algorithms: 1) because each node communicates only with other nodes in some neighborhood, the communication overhead *scales* well with increase in network size; 2) for a similar reason these algorithms are *robust* to network partitions and node failures.

Specifically for topology control, it is preferred that the underlying network topology can be constructed in a localized manner. Here a distributed algorithm constructing a graph G is a *localized algorithm* if every node u can exactly decide all edges incident on u based only on the information of all nodes within a constant hops of u (plus a constant number of additional nodes' information if necessary). It is easy to see that the Yao graph YG(V), the relative neighborhood graph RNG(V) and the Gabriel graph GG(V) can be constructed locally. However, the Euclidean minimum spanning tree MST(V) and the Delaunay triangulation Del(V) can not be constructed by any localized algorithm. In next section, we are interested in localized algorithms that construct sparse and power efficient network topologies.

3 Topology Control

In this section, we first study the power stretch factor of several sparse geometric structures for unit disk graph, which can be used as the network topology of sensor networks. Notice that, a trade-off can be made between the sparseness of the topology and the power efficiency. The power efficiency of any spanner is measured by its power stretch factor. We then review some geometric results on topology issues in sensor networks, such as fault tolerance and interference.

3.1 Bounded Degree Structures

Beside the sparseness and spanner properties, 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. Therefore, in this subsection, we review some bounded degree spanners.

3.1.1 Yao Structure

Applying Yao structure to bound node degree is very nature idea. The Yao graph $YG_k(V)$ has length stretch factor $\frac{1}{1-2\sin\frac{\pi}{k}}$. Thus, its power stretch factor is no more than $(\frac{1}{1-2\sin\frac{\pi}{k}})^{\beta}$. Li *et al.* [102] proved a stronger result.

Theorem 1 *[102] The power stretch factor of the Yao graph $YG_k(V)$ is at most $\frac{1}{1-(2\sin\frac{\pi}{k})^{\beta}}$.*

See [102] for a detailed proof of this theorem. Li *et al.* [103] also proposed to apply the Yao structure on top of the Gabriel graph structure (the resulting graph is denoted by $\overrightarrow{YGG}_k(V)$), and apply the Gabriel graph structure on top of the Yao structure (the resulting graph is denoted by $\overrightarrow{GYG}_k(V)$). These structures are sparser than the Yao structure and the Gabriel graph structure and they still have a constant bounded power stretch factor. These two structures are connected graphs if the UDG is connected, which can be proved by showing that RNG is a subgraph of both structures.

The two-phased approach by Wattenhofer *et al.* [156] consists of a variation of the Yao graph followed by a variation of the Gabriel graph. They tried to prove that the constructed spanner has a constant power stretch factor and the node degree is bounded by a constant. Unfortunately, there are some discrepancies in their proof of the constant power stretch factor and their result is erroneous, which was discussed in detail in [102].

Li *et al.* [93] proposed a structure that is similar to the Yao structure for topology control. Each node u finds a power $p_{u,\alpha}$ such that in every cone of degree α surrounding u, there is some node that u can reach with power $p_{u,\alpha}$. Here, nevertheless, we assume that there is a node reachable from u by the maximum power in that cone. Then the graph G_α contains all edges uv such that u can communicate with v using power $p_{u,\alpha}$. It was proved in [93] that, if $\alpha \leq \frac{5\pi}{6}$ and the UDG is connected, then graph G_α is a connected graph. On the other hand, if $\alpha > \frac{5\pi}{6}$, they showed that the connectivity of G_α is not guaranteed by giving some counter-example [93].

Notice that although the directed graphs $\overrightarrow{YG}_k(V)$, $\overrightarrow{GYG}_k(V)$ and $\overrightarrow{YGG}_k(V)$ have a bounded power stretch factor and a bounded out-degree k for each node, some nodes may have very large in-degrees. The nodes configuration given in Figure 5 will result a very large in-degree for node u. Bounded out-degree gives us advantages when apply several routing algorithms. However, unbounded in-degree at node u will often cause large overhead at u. Therefore it is often imperative to construct a sparse network topology such that both the in-degree and the out-degree are bounded by a constant while it is still power-efficient.

3.1.2 Sink Structure

Arya *et al.* [7] gave an ingenious technique to generate a bounded degree graph with constant length stretch factor. In [102], Li *et al.* applied the same technique to construct a sparse network topology with a bounded degree and a bounded power stretch factor from $YG(V)$. The technique is to replace the directed star consisting of all links toward a node u by a directed tree $T(u)$ of a bounded degree with u as the sink. Tree $T(u)$ is constructed recursively. The algorithm is as follows.

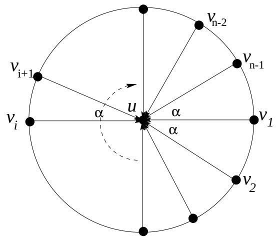


Figure 5: Node u has degree (or in-degree) $n-1$.

Algorithm 1 *Constructing-YG**

1. First, construct the graph $\overrightarrow{YG}_k(V)$. Each node u will have a set of in-coming nodes $I(u) = \{v \mid \overrightarrow{vu} \in \overrightarrow{YG}_k(V)\}$.
2. For each node u, use the following algorithm Tree(u,$I(u)$) to build tree $T(u)$.

Algorithm 2 *Constructing-$T(u)$ Tree(u,$I(u)$)*

1. To partition the unit disk centered at u, choose k equal-sized cones centered at u: $C_1(u)$, $C_2(u)$, $\cdots$, $C_k(u)$.
2. Node u finds the nearest node $y_i \in I(u)$ in $C_i(u)$, for $1 \leq i \leq k$, if there is any. Link $\overrightarrow{y_i u}$ is added to $T(u)$ and y_i is removed from $I(u)$. For each cone $C_i(u)$, if $I(u) \cap C_i(u)$ is not empty, call Tree(y_i,$I(u) \cap C_i(u)$) and add the created edges to $T(u)$.

Figure 6 (a) illustrates a directed star centered at u and Figure 6 (b) shows the directed tree $T(u)$ constructed to replace the star with $k = 8$. The union of all trees $T(u)$ is called the *sink structure* $\overrightarrow{YG}^*_k(V)$.

Notice that, node u constructs the tree $T(u)$ and then broadcasts the structure of $T(u)$ to all nodes in $T(u)$. Since the total number of edges in the Yao structure is at most $k \cdot n$, where k is the number of cones divided, the total number of edges of $T(u)$ of all nodes u is also at most $k \cdot n$. Thus, the total communication cost of broadcasting the $T(u)$ to all its neighbors is still at most $k \cdot n$. Recall that k is a small constant.

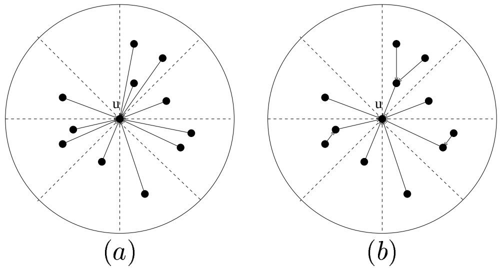


Figure 6: (a) Star formed by links toward to u. (b) Directed tree $T(u)$ sinked at u.

The algorithm uses a directed tree $T(u)$ to replace the directed star for each node u. Therefore, if nodes u and v are connected by a path in $\overrightarrow{YG}_k$, they are also connected by a path in $\overrightarrow{YG}^*_k$. It is already known that $\overrightarrow{YG}_k$ is strongly connected if UDG(V) is connected, so does $\overrightarrow{YG}^*_k$.

Theorem 2 *[102] The power stretch factor of the graph $\overrightarrow{YG}^*_k(V)$ is at most $(\frac{1}{1-(2\sin\frac{\pi}{k})^\beta})^2$. The maximum degree of the graph $\overrightarrow{YG}^*_k(V)$ is at most $(k+1)^2 - 1$. The maximum out-degree is k.*

Notice that the sink structure and the Yao graph structure do not have to have the same number of cones, and the cones do not need to be aligned. For setting up a power-efficient wireless networking, each node u finds all its neighbors in $YG_k(V)$, which can be done in linear time proportional to the number of nodes within its transmission range.

3.1.3 YaoYao Structure

In this section, we review another algorithm proposed by Li *et al.* [103] that constructs a sparse and power efficient topology. Assume that each node v_i of V has a unique identification number $ID(v_i) = i$. The identity of a directed link $\overrightarrow{uv}$ is defined as $ID(\overrightarrow{uv}) = (||uv||, ID(u), ID(v))$.

Node u chooses a node v from each cone, if there is any, so the directed link $\overrightarrow{vu}$ has the smallest $ID(\overrightarrow{vu})$ among all directed links $\overrightarrow{wu}$ in $YG(V)$ in that cone. The union of all chosen directed links is the final network topology, denoted by $\overrightarrow{YY}_k(V)$. If the directions of all links are ignored, the graph is denoted as $YY_k(V)$.

Theorem 3 *[103] Graph $\overrightarrow{YY}_k(V)$ is strongly connected if UDG(V) is connected and $k > 6$.*

It was proved in [153] that $\overrightarrow{YY}_k(V)$ is a spanner in civilized graph. Here a unit disk graph is civilized graph if the distance between any two nodes in this graph is larger than a positive constant λ. In [68], they called the civilized unit disk graph as the λ-precision unit disk graph. Notice the sensor devices in wireless sensor networks can not be too close or overlapped. Thus, it is reasonable to model the wireless sensor networks as a civilized unit disk graph.

Theorem 4 *[153] The power stretch factor of the directed topology $\overrightarrow{YY}_k(V)$ is bounded by a constant in civilized graph.*

The experimental results by Li *et al.* [103] showed that this sparse topology has a small power stretch factor in practice. They [103, 153] conjectured that $\overrightarrow{YY}_k(V)$ also has constants bounded length spanning ratio and power stretch factor theoretically in any unit disk graph. Recently Jia *et al.* [72] claim that they prove that $\overrightarrow{YY}_k(V)$ also has a constant bounded power stretch factor theoretically in general graphs. For the detailed proof please refer to [72].

3.1.4 Symmetric Yao Graph

In [99, 100], Li *et al.* also considered another undirected structure, called *symmetric Yao graph* $YS_k(V)$, which guarantees that the node degree is at most k. Each node u divides the region into k equal angular regions centered at the node, and chooses the closest node in each region, if any. An edge uv is selected to graph $YS_k(V)$ if and only if both directed edges $\overrightarrow{uv}$ and $\overrightarrow{vu}$ are in the *Yao graph* $\overrightarrow{YG}_k(V)$. Then it is obvious that the maximum node degree is k.

Theorem 5 *[99, 153] The graph $YS_k(V)$ is strongly connected if UDG(V) is connected and $k \geq 6$.*

The experiment by Li *et al.* also showed that it has a small power stretch factor in practice. However, it was shown in [59] recently that $YS_k(V)$ is not a spanner theoretically. See [96] for more detail.

3.2 Planar Structures

Some routing algorithms ask the network topology be planar, such as right hand routing, *Greedy Perimeter Stateless Routing* (GPSR) [78], *Greedy Face Routing* (GFG) [20], *Adaptive Face Routing*(AFR) [87]. and *Gready Other Adaptive Face Routing* (GOAFR) [88]. Therefore, we want the constructed topology to be a planar graph, i.e., no two edges cross each other in the graph. In this subsection, we study three planar structures which can be used in sensor networks.

3.2.1 RNG and GG

Remember that the relative neighborhood graph and Gabriel graph are planar graph, and they can be constructed easily using localized methods.

Since the relative neighborhood graph has the length stretch factor as large as $n - 1$, then obviously its power stretch factor is at most $(n - 1)^2$. Li *et al.* [102] showed that it is actually $n - 1$.

The Gabriel graph has length stretch factor between $\frac{\sqrt{n}}{2}$ and $\frac{4\pi\sqrt{2n-4}}{3}$[17]. Wang *et al.* [152] showed that it is actually exactly $\sqrt{n-1}$. Then its power stretch factor is at most $\left(\frac{4\pi\sqrt{2n-4}}{3}\right)^2$. In [102], Li *et al.* also prove that the power stretch factor of any Gabriel graph is one.

Notice that both RNG and GG do not have bounded node degree. The nodes configuration given in Figure 5 will also result a very large degree for node u.

3.2.2 Localized Delaunay Triangulation

Given a set of nodes V, it is well-known that the Delaunay triangulation $Del(V)$ is a planar t-spanner of the completed graph $K(V)$. However, it is not appropriate to require the construction of the Delaunay triangulation in the wireless communication environment because of the possible massive communications it requires. Given a set of points V, let UDel(V) be the graph of removing all edges of $Del(V)$ that are longer than one unit, i.e., $UDel(V) = Del(V) \cap UDG(V)$. Li *et al.* [98] considered the *unit Delaunay triangulation* UDel(V) for planar spanner of UDG, which is a subset of the Delaunay triangulation. It was proved in [98] that UDel(V) is a t-spanner of the unit disk graph UDG(V).

Theorem 6 *[98] For any two vertices u and v of V, $||\Pi_{UDel(V)}(u,v)|| \le \frac{1+\sqrt{5}}{2}\pi \cdot ||\Pi_{UDG(V)}(u,v)||$.*

Notice that, Kevin and Gutwin [80] showed that the Delaunay triangulation is a t-spanner for a constant $t \approx 2.42$. This was proved by induction on the order of the lengths of all pair of nodes (from the shortest to the longest). It can be shown that the path connecting nodes u and v constructed by the method given in [80] also satisfies that all edges of that path is shorter than $\|uv\|$. Consequently, we know that the unit Delaunay triangulation $UDel(V)$ is a $\frac{4\sqrt{3}}{9}\pi$-spanner of the unit disk graph $UDG(V)$.

Li *et al.* [98] gave a localized algorithm that constructs a sequence graphs, called *localized Delaunay* $LDel^{(k)}(V)$, which are supergraphs of UDel(V). We begin with some necessary definitions before presenting the algorithm. Triangle $\triangle uvw$ is called a *k-localized Delaunay triangle* if the interior of the circumcircle of $\triangle uvw$, denoted by $disk(u,v,w)$ hereafter, does not contain any vertex 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. The *k-localized Delaunay graph* over a vertex set V, denoted by $LDel^{(k)}(V)$, has exactly all Gabriel edges in UDG and edges of all k-localized Delaunay triangles.

When it is clear from the context, we will omit the integer k in our notation of $LDel^{(k)}(V)$. As shown in [98], the graph $LDel^{(1)}(V)$ may contain some edges intersecting. On the other hand, $LDel^{(2)}(V)$ is a planar graph.

Theorem 7 *[98] $LDel^{(k)}(V)$ is a planar graph for any $k \ge 2$.*

Notice that, although $LDel^{(1)}(V)$ is not a planar graph, the following theorem proved in [98] guarantees that it is sparse.

Theorem 8 *Graph $LDel^{(1)}(V)$ has thickness 2.*

Although the graph $UDel(V)$ is a t-spanner for $UDG(V)$, it is unknown how to construct it locally. We can construct $LDel^{(2)}(V)$, which is guaranteed to be a planar spanner of $UDel(V)$, but a total communication cost of a simple approach is $O(m \log n)$ bits, where m is the number of edges in $UDG(V)$ and could be as large as $O(n^2)$. In order to reduce the total communication cost to $O(n \log n)$ bits, they do not construct $LDel^{(2)}(V)$, and instead they extract a planar graph $PLDel(V)$ out of $LDel^{(1)}(V)$. They provided a novel algorithm to construct $LDel^{(1)}(V)$ using linear communications and then make it planar in linear communication cost. The final graph still contains $UDel(V)$ as a subgraph. Thus, it is a t-spanner of the unit-disk graph $UDG(V)$.

In the following, the order of three nodes in a triangle is immaterial.

Algorithm 3 *Localized Unit Delaunay Triangulation*

1. Each wireless node u broadcasts its identity and location and listens to the messages from other nodes.

2. Assume that node u gathered the location information of $N_1(u)$. It computes the Delaunay triangulation $Del(N_1(u))$ of its 1-neighbors $N_1(u)$, including u itself.

3. For each edge uv of $Del(N_1(u))$, 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 $\pi/2$. Node u marks all *Gabriel edges* uv, which will never be deleted.

4. Each node u finds all triangles $\triangle uvw$ from $Del(N_1(u))$ such that all three edges of $\triangle uvw$ have length at most one unit. If angle $\angle wuv \geq \frac{\pi}{3}$, node u broadcasts a message $\mathsf{proposal}(u, v, w)$ to form a 1-localized Delaunay triangle $\triangle uvw$ in $LDel^{(1)}(V)$, and listens to the messages from other nodes.

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

6. A node u adds the edges uv and uw to its set of incident edges if the triangle $\triangle uvw$ is in the Delaunay triangulation $Del(N_1(u))$ and both v and w have sent either $\mathsf{accept}(u, v, w)$ or $\mathsf{proposal}(u, v, w)$.

It was proved that the graph constructed by the above algorithm is $LDel^{(1)}(V)$. Indeed, for each triangle $\triangle uvw$ of $LDel^{(1)}(V)$, one of its interior angles is at least $\pi/3$ and $\triangle uvw$ is in $Del(N_1(u))$, $Del(N_1(v))$ and $Del(N_1(w))$. So one of the nodes amongst $\{u, v, w\}$ will broadcast the message $\mathsf{proposal}(u, v, w)$ to form a 1-localized Delaunay triangle $\triangle uvw$.

As $Del(N_1(u))$ is a planar graph, and a proposal is made only if $\angle wuv \geq \frac{\pi}{3}$, node u broadcasts at most 6 proposals. And each proposal is replied by at most two nodes. Therefore, the total communication cost is $O(n \log n)$ bits. The above algorithm also shows that $LDel^{(1)}(V)$ has $O(n)$ edges, which we know from Theorem 8. Putting together the arguments above, we have:

Theorem 9 *[98] Algorithm 3 constructs $LDel^{(1)}(V)$ with total communication cost $O(n \log n)$ bits.*

We then review the algorithm to extract from $LDel^{(1)}(V)$ a planar subgraph.

Algorithm 4 *Planarize $LDel^{(1)}(V)$*

1. Each wireless node u broadcasts the Gabriel edges incident on u and the triangles $\triangle uvw$ of $LDel^{(1)}(V)$ and listens to the messages from other nodes.

2. Assume node u gathered the Gabriel edge and 1-local Delaunay triangles information of all nodes from $N_1(u)$. For two intersected triangles $\triangle uvw$ and $\triangle xyz$ known by u, node u removes the triangle $\triangle uvw$ if its circumcircle contains a node from $\{x, y, z\}$.

3. Each wireless node u broadcasts all the triangles incident on u which it has not removed in the previous step, and listens to the broadcasting by other nodes.

4. Node u keeps the edge uv in its set of incident edges if it is a Gabriel edge, or if there is a triangle $\triangle uvw$ such that u, v, and w have all announced they have not removed the triangle $\triangle uvw$ in Step 2.

They denoted the graph extracted by the algorithm above by $PLDel(V)$. Note that any triangle of $LDel^{(1)}(V)$ not kept in the last step of the Planarization Algorithm is not a triangle of $LDel^{(2)}(V)$, and therefore $PLDel(V)$ is a supergraph of $LDel^{(2)}(V)$. Thus,

$$UDel(V) \subseteq LDel^{(2)}(V) \subseteq PLDel(V) \subseteq LDel^{(1)}(V)$$

Similar to the proof that $LDel^{(2)}(V)$ is a planar graph, they showed that the algorithm does generate a planar graph. The total communication cost to construct the graph $PLDel(V)$ is a $O(\log n)$ times the number of edges of the graph $LDel^{(1)}(V)$, which by Theorem 8 is $O(n)$. Putting together all the arguments above and Theorem 6,

Theorem 10 *$PLDel(V)$ is planar $\frac{4\sqrt{3}}{9}\pi$-spanner of $UDG(V)$, and can be constructed with total communication cost $O(n \log n)$ bits.*

They [98] cannot construct $LDel^{(2)}$ in $O(n)$ messages due to the difficulty of collecting the 2-hop neighbors for every node in $O(n)$ messages. 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. Recently, Gruia [25] proposed an approach (using $O(n)$ messages total) which is based on the specific connected dominating set introduced by Alzoubi, Wan, and Frieder [2]. 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. For detailed algorithm and proofs, please refer to [25]. Using this approach, we can build $LDel^{(2)}$ in $O(n)$ messages. There is a such algorithm proposed in [155].

3.2.3 Partial Delaunay Triangulation

Stojmenovic and Li [99] also proposed a geometry structure, namely the partial Delaunay triangulation (PDT), that can be constructed in a localized manner. Partial Delaunay triangulation contains Gabriel graph as its subgraph, and itself is a subgraph of the Delaunay triangulation, more precisely, the subgraph of the unit Delaunay triangulation UDel(V). The algorithm for the construction of PDT goes as follows.

Let u and v be two neighboring nodes in the network. Edge uv belongs to $Del(V)$ if and only if there exists a disk with u and v on its boundary, which does not contain any other point from the set V. First test whether $disk(u, v)$ contains any other node from the network. If it does not, the edge belongs to GG and therefore to PDT. If it does, check whether nodes exist on both sides of line uv or on only one side. If both sides of line uv contain nodes from the set inside $disk(u, v)$ then uv does not belong to $Del(V)$.

Suppose now that only one side of line uv contains nodes inside the circle $disk(u, v)$, and let w be one such point that maximizes the angle $\angle uwv$. Let $\alpha = \angle uwv$. Consider now the largest angle $\angle uxv$ on the other side of the mentioned circle $disk(u, v)$, where x is a node from the set S. If $\angle uwv + \angle uxv > \pi$, then edge uv is definitely not in the Delaunay triangulation $Del(V)$. The search can be restricted to common neighbors of u and v, if only one-hop neighbor information is available, or to neighbors of only one of the nodes if 2-hop information (or exchange of the information for the purpose of creating PDT is allowed) is available. Then whether edge uv is added to PDT is based on the following procedure.

Assume only $N_1(u)$ is known to u, and there is one node w from $N_1(u)$ that is inside $disk(u, v)$ with the largest angle $\angle uwv$. Edge uv is added to PDT if the following conditions hold: (1) there is no node from $N_1(u)$ that lies on the different side of uv with w and inside the circumcircle passing through u, v, and w, (2) $\sin \alpha > \frac{d}{R}$, where R is the transmission radius of each wireless node, d is the diameter of the circumcircle $disk(u, v, w)$, and $\alpha = \angle uwv$ (here $\alpha \geq \frac{\pi}{2}$).

Assume only 1-hop neighbors are known to u and v, and there is one node w from $N_1(u) \cup N_1(v)$ that is inside $disk(u, v)$ with the largest angle $\angle uwv$. Edge uv is added to PDT if the following conditions hold: (1) there is no node from $N_1(u) \cup N_1(v)$ that lies on the different side of uv with w and inside the circumcircle passing u, v, and w, (2) $\cos \frac{\alpha}{2} > \frac{d}{2R}$, where R is the transmission radius of each wireless node and $\alpha = \angle uwv$.

Obviously, PDT is a subgraph of $UDel(V)$. It is not difficult to construct an example such that the spanning ratio of the partial Delaunay triangulation could be very large.

3.2.4 Restricted Delaunay Graph

Gao *et al.* [55] also proposed another structure, called *restricted Delaunay graph* RDG and showed that it has good spanning ratio properties and can be maintained locally. A restricted Delaunay graph of a set of points in the plane is a planar graph and contains all the Delaunay edges with length at most one. In other other words, they call *any* planar graph containing $UDel(V)$ as a restricted Delaunay graph. They described a distributed algorithm to construct a RDG such that at the end of the algorithm, each node u maintains a set of edges $E(u)$ incident to u. Those edges $E(u)$ satisfy that (1) each edge in $E(u)$ has length at most one unit; (2) the edges are consistent, i.e., an edge $uv \in E(u)$ if and only if $uv \in E(v)$; (3) the graph obtained is planar; (4) The graph $UDel(V)$ is in the union of all edges $E(u)$.

The algorithm works as follows. First, each node u acquires the position of its 1-hop neighbors $N_1(u)$ and computes the Delaunay triangulation $Del(N_1(u))$ on $N_1(u)$, including u itself. In the second step, each node u sends $Del(N_1(u))$ to all of its neighbors. Let $E(u) = \{uv \mid uv \in Del(N_1(u))\}$. For each edge $uv \in E(u)$, and for each $w \in N_1(u)$, if u and v are in $N_1(w)$ and $uv \notin Del(N_1(u))$, then node u deletes edge uv from $E(u)$.

When the above steps are finished, the resulting edges $E(u)$ satisfy the four properties listed above. However, unlike the local Delaunay triangulation, the computation cost and communication cost of each node needed to obtain $E(u)$ is not optimal within a small constant factor.

3.3 Bounded Degree, Planar Structures

The structures discussed so far either have bounded degree, or are planar, or are spanners, but none of the structures has all these three properties together. We then review one recent result [155] that can construct a bounded degree planar spanner in a localized manner (total communication cost is $O(n \log n)$ bits). No localized method is known before this result for constructing a planar spanner with bounded node degree. Their method rigorously combines (localized) Delaunay triangulation $LDel^{(2)}(V)$ and the ordered Yao structure [21, 164].

Algorithm 5 *Localized Construction of Planar Spanner with Bounded Degree for* $UDG(V)$

1. First, compute the planar localized Delaunay triangulation $LDel^{(2)}(V)$ (using the method in [25] to collect the location information of $N_2(u)$), 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 $\pi_v = k$, 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 a unprocessed node u has the highest local order in its unprocessed neighbors N_u in $LDel^{(2)}(V)$, let k be the number of processed neighbors [2] of u in $LDel^{(2)}(V)$. Assume that $v_1, v_2, \cdots, v_k$ be the processed neighbors of u in $LDel^{(2)}(V)$ (see Figure 7). 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 $s_1, s_2, \cdots, s_m$ be the ordered unprocessed neighbors of u in $N_{LDel^{(2)}}(u)$. For this cone, node u first adds an edge us_i, where s_i is the nearest neighbor among $s_1, s_2, \cdots, s_m$. Node u then tells $s_1, s_2, \cdots, s_m$ to add all the edges $s_j s_{j+1}$, $1 \le j < m$. Node u marks itself processed, and tells all nodes in $N_{LDel^{(2)}}(u)$ that it is processed.

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

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

[1] If some unordered neighbor 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 since graph $LDel^{(2)}(V)$ is planar.

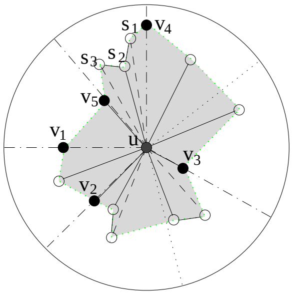


Figure 7: Constructing Planar Spanner with Bounded Degree for $UDG(V)$: Process node u.

Notice that in the algorithm we use *open* sectors, which means that in the algorithm we do not consider adding the edges on the boundaries (any edge involved previously processed neighbors). For example, in Figure 7, the cones do not include any edges uv_i. This guarantees the algorithm does not add any edges to node v_i after v_i has been processed. This approach bounds the node degree.

Theorem 11 *[155] The maximum node degree of the graph $BPS(V)$ is at most* $19 + \lceil \frac{2\pi}{\alpha} \rceil$.

For example, when $\alpha = \pi/3$, the maximum node degree is at most 25.

Notice that the ordering computed by this method is not a total ordering. Some nodes may have the same order. However, no two neighboring nodes in $LDel^{(2)}(V)$ receive the same order. Thus, after all nodes are ordered, the algorithm will process all nodes. Observe that the algorithm do 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 the ordering method.

The algorithms in [22, 105] 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. However Algorithm 5 may add some edges (such as edges $s_i s_{i+1}$ added in step 4(b) that do not belong to the $UDel(V)$. Therefore, the proof of the following theorem is much complex.

Theorem 12 *[155] Graph $BPS(V)$ is a planar graph.*

Then in [155], they prove that the structure is also a spanner.

Theorem 13 *[155] Graph $BPS(V)$ is a t-spanner, where*

$$t = \max\{\frac{\pi}{2}, \pi \sin\frac{\alpha}{2} + 1\} \cdot C_{del}.$$

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

Theorem 14 *[155] Algorithm 5 uses at most $O(n)$ messages, where each message has $O(\log n)$ bits.*

Proof. Notice that it was shown in [25] that we can collect the 2-hop neighbor information for all nodes using total $O(n)$ messages. The communication cost of building $LDel^{(2)}$ is $O(n)$ since every node only has to propose at most 6 triangles and each propose is replied by two nodes.

The second step (local ordering) takes $O(n)$ messages, since every node only query 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: (1) tell its neighbors to add some edges, and (2) claims that it is processed. The total messages of telling neighbors to add some edges is $O(n)$ since the total added edges is $O(n)$ from the planar property of the final topology. So the total communication cost is bounded by $O(n)$. □

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 UDG. 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 UDG. The improvement is based on the fact that we only need the triangles $\triangle wuv$ in $LDel^{(2)}(V)$ that has angle $\angle wuv \geq \pi/3$. All such triangles are definitely in $LDel^{(1)}(V)$. Thus, we can construct the Delaunay triangulation $Del(N_1(u))$ instead. Then check each candidate triangle $\triangle wuv$ from $LDel^{(1)}(V)$ to see if they contain any node from $N_2(u)$ inside its circumcircle. If it does not, then it belongs to $Del(N_2(u))$.

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

3.4 Bounded Degree, Planar, Low Weight Structures

Remember that low weight is also a desirable feature for network topology in sensor networks. However, the total weight of any graph structures mentioned above (except MST) could be arbitrarily larger than MST theoretically [103, 104]. Figure 8 gives such an example of wireless sensor nodes. Here $\|u_i v_i\| = 1$ and $\|u_i u_{i+1}\| = \|v_i v_{i+1}\| = \epsilon$ for a very small positive real number ϵ. The graph shown in the example is the relative neighborhood graph RNG(V). It is easy to show that

$$\frac{\sum_{e \in RNG} \|e\|^\beta}{\sum_{e \in MST} \|e\|^\beta} = \frac{n + 2(n-1)\epsilon^\beta}{1 + 2(n-1)\epsilon^\beta} \to 0$$

when $\epsilon \to 0$. Notice that all other graph structures (except MST) contain RNG as a subgraph for this node configuration. It then implies our previous claim.

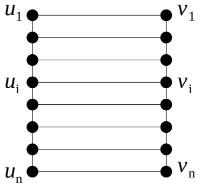


Figure 8: An instance of wireless sensor nodes that every network structures described previously (except MST) have an arbitrarily large total weight.

In this section we will discuss how to design algorithms achieving low weight, and (possibly) in additional to some other properties such as spanner, bounded degree, and planar. Unfortunately, until now, there is no efficient localized algorithm that can achieve all there properties. Arya *et al.* [7, 8] gave a centralized algorithm to construct a spanner with bounded node degree and the total edge length is no more than a constant factor of that of $MST(V)$. However, it is very complicated to transform their algorithms to a distributed algorithm. And the spanner is not guaranteed a planar graph.

3.4.1 Centralized Low Weight Bounded Degree Planar Spanners

Recently, Bose *et al.* [22] proposed an algorithm which constructs a bounded degree and planar spanner for a given points set V. They show that the length stretch factor of the final graph is $\frac{(\pi+1)2\pi}{(3\cos\pi/6)(1+\epsilon)}$ and node degree is at most 27. The running time of their algorithm is $O(n \log n)$. However, their method is impossible to have a localized even distributed version, since they use BFS and many operations on polygons (such as degree-3 partitions). Notice that breadth-first-search may take $O(n^2)$ communications.

Then borrowing some ideas from their method, in [105], we proposed another method for constructing a low weight bounded degree planar spanner.

1. First, it computes the Delaunay triangulation of a set V of n nodes, $Del(V)$. Let $N_{Del}(u)$ be the neighbors of node u in the Delaunay triangulation $Del(V)$, and d_u be the degree of node u in $Del(V)$. By proper data structure, $N_{Del}(u)$ and d_u can be achieved in time $O(n)$.

2. Find an order π of V as follows. Let $G_1 = Del(V)$ and $d_{G,u}$ be the node degree of u in graph G. Find the node u_1 with the smallest value of $(d_{G_1,u}, ID(u))$, let $\pi_{u_1} = n$. Then remove u_1 and its adjacent edges from G_1, the remaining graph G_2 is still a planar graph. Find the node u_2 with smallest value of $(d_{G_2,u}, ID(u))$, let $\pi_{u_2} = n - 1$. Repeat this procedure, until G_n only has one node u_n, let $\pi_{u_n} = 1$. Let P_v denote the predecessors of v in π, i.e., $P_v = \{u \in V : \pi_u < \pi_v\}$. Notice since G_i is always a planar graph, we know that the smallest value of $d_{G_i,u}$ is at most 5. Then, in ordering π, node u at most have 5 edges to its predecessors P_u in $Del(V)$.

3. Let E be the edge set of $Del(V)$, E' be the edge set of the desired spanner. Initialize E' to be empty set and all nodes in V are unprocessed. Then, for each node u in V, following the increasing order π, run the following steps to add some edges from E to E' (we only consider the Delaunay neighbors $N_{Del}(u)$ of u):

   (a) We use $v_1, v_2, \cdots, v_k$ to denote the predecessors of node u. Notice that u can have at most 5 edges to its predecessors (processed Delaunay neighbors) in E, i.e., $k \leq 5$. Then there are $k \leq 5$ *open* sectors at node u whose boundaries are rays emanated from u to the processed neighbors v_i of u in $Del(V)$. For each such sector at u, we divide it into a minimum number of *open* cones of degree at most α, where $\alpha \leq \pi/2$ is a parameter.

   (b) For each such cone, let $s_1, s_2, \cdots, s_m$ be the geometrically ordered neighborhood $N_{Del}(u)$ of u in this cone. That is, $s_1, s_2, \cdots, s_m$ are all *unprocessed* nodes that are connected by some edges of E to u in this cone. For this cone, we first add the shortest edge in E that is connected to u to the edge set E', then add to E' all the edges (s_j, s_{j+1}), $1 \leq j < m$.

   (c) Mark node u processed.

4. Repeat this procedure in the increasing order of π, until all nodes are processed. The final graph is formed by all edges in E'.

5. Run the greedy spanner algorithm by [60] to bound the weight of the graph.

We proved the following theorem:

Theorem 15 *[105] Given a set V of n points in a two-dimensional plane, the above $O(n \log n)$-time algorithm constructs a graph*

1. *that is planar,*

2. *that is a t-spanner, for $t = \max\{\frac{\pi}{2}, \pi \sin\frac{\alpha}{2} + 1\} \cdot C_{del}(1 + \epsilon)$,*

3. *in which each point of V has degree at most $19 + \lceil\frac{2\pi}{\alpha}\rceil$,*

4. *and whose total edge weight is bounded from above by a constant factor of the weight of the Euclidean minimum spanning tree of V. Here the constant factor depends on ϵ.*

Here $0 < \alpha < \pi/2$ is an adjustable parameter.

Notice that we can build Delaunay triangulation in $O(n \log n)$, and do ordering in time $O(n \log n)$ (using heap for the ordering based on degrees), and Yao structure in $O(n)$ (each edge is processed at most a constant times and there are $O(n)$ edges to be processed). Thus, the time complexity of the algorithm is $O(n \log n)$, same with the method by Bose *et al.* [22]. However, our algorithm has smaller bounded node degree, and (more importantly) our algorithm has potential to become a localized version for wireless sensor networks application. The only problem here is the last step – greedy method can not be performed in a local way.

3.4.2 Localized Low Weight Bounded Degree Planar Structures

Recently, Li *et al.* [97, 107] proposed three localized structures which are low weight, planar and have bounded node degree. However, they are not spanners.

A: Structure based on RNG'

In [97], Li gave the first localized method to construct a structure H with weight $O(\omega(MST))$ using total $O(n)$ local-broadcast messages. The method is based on a modified relative neighborhood graph. Notice that, traditionally, the relative neighborhood graph will always select an edge uv even if there is some node on the boundary of $lune(u,v)$. Thus, RNG may have unbounded node degree, e.g., considering $n-1$ points equally distributed on the circle centered at the nth point v, the degree of v is $n-1$. Notice that for the sake of lowing the weight of a structure, the structure should contain as less edges as possible without breaking the connectivity. Li [97] then naturally extended the traditional definition of RNG as follows.

The *modified relative neighborhood graph* consists of all edges uv such that (1) the *interior* of $lune(u,v)$ contains no point $w \in V$ and, (2) there is no point $w \in V$ with $ID(w) < ID(v)$ on the boundary of $lune(u,v)$ and $\|wv\| < \|uv\|$, and (3) there is no point $w \in V$ with $ID(w) < ID(u)$ on the boundary of $lune(u,v)$ and $\|wu\| < \|uv\|$, and (4) there is no point $w \in V$ on the boundary of $lune(u,v)$ with $ID(w) < ID(u)$, $ID(w) < ID(v)$, and $\|wu\| = \|uv\|$. See Figure 9 for an illustration when an edge uv is *not* included in the modified relative neighborhood graph. Li called such structure by RNG'. Obviously, RNG' is a subgraph of traditional RNG. It was proved in [97] that RNG' has a maximum node degree 6 and still contains a MST as a subgraph. However, RNG' is still not a low weight structure.

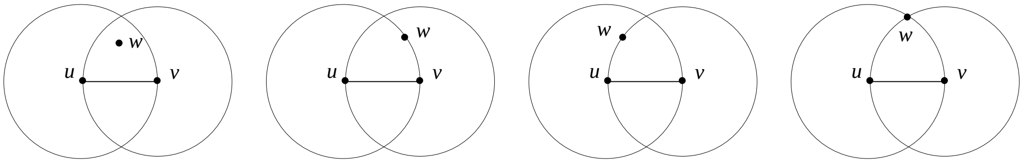


Figure 9: Four cases when edges are not in the modified RNG.

Obviously, graph RNG' still can be constructed using n messages. Each node first locally broadcasts its location and ID to its one-hop neighbors. Then every node decides which edge to keep solely based on the one-hop neighbors' location information collected. Since the definition is still symmetric, the edges constructed by different nodes are consistent, i.e., an edge uv is kept by a node u iff it is also kept by node v. The computational cost of a node u is still $O(d \log d)$, where d is its degree in UDG. A simple edge by edge testing method has time complexity $O(d^2)$.

Notice that it is well-known that the communication complexity of constructing a minimum spanning tree of a n-vertex graph G with m edges is $O(m+n\log n)$; while the communication complexity of constructing MST for UDG is $O(n \log n)$ even under the local broadcasting communication model in wireless networks. It was shown in [97] that it is *impossible* to construct a low-weighted structure using only one hop neighbor information.

The localized algorithm given in [97] that constructs a low-weighted structure using only some two hops information is as follows.

Algorithm 6 *Construct Low Weight Structure H*

1. All nodes together construct the graph RNG' in a localized manner.
2. Each node u locally broadcasts its incident edges in RNG' to its one-hop neighbors. Node u listens to the messages from its one-hop neighbors.
3. Assume node u received a message informing existence of edge $xy \in RNG'$ from its neighbor x. For each edge $uv \in RNG'$, if uv is the longest among uv, xy, ux, and vy, node u removes edge uv. Ties are broken by the label of the edges. Here assume that $uvyx$ is the convex hull of u, v, x, and y.
4. Let H be the final structure formed by all remaining edges in RNG'.

Obviously, if an edge uv is kept by node u, then it is also kept by node v.

Theorem 16 *[97] The total edge weight of H is within a constant factor of that of the minimum spanning tree.*

This was proved by showing that the edges in H satisfy the *isolation property* (defined in [46]). They [97] also showed that the final structure contains MST of UDG as a subgraph.

Clearly, the communication cost of Algorithm 6 is at most $7n$: initially each node spends one message to tell its one-hop neighbors its position information, then each node uv tells its one-hop neighbors all its incident edges $uv \in RNG'$ (there are at most total $6n$ such messages since RNG' has at most $3n$ edges). The computational cost of Algorithm 6 could be high since for each link $uv \in RNG'$, node u has to test whether there is an edge $xy \in RNG'$ and $x \in N_1(u)$ such that uv is the longest among uv, xy, ux, and vy. Then [107] presents some new algorithms that improve the computational complexity of each node while still maintains low communication costs.

B: Structure based on LMST

The first new method in [107] uses a structure called *local minimum spanning tree*, let us first review its definition. It is first proposed by Li, Hou and Sha [95]. Each node u first collects its one-hop neighbors $N_1(u)$. Node u then computes the minimum spanning tree $MST(N_1(u))$ of the induced unit disk graph on its one-hop neighbors $N_1(u)$. Node u keeps a directed edge uv if and only if uv is an edge in $MST(N_1(u))$. They call the union of all directed edges of all nodes the *local minimum spanning tree*, denoted by $LMST_1$. If only symmetric edges are kept, then the graph is called $LMST_1^-$, i.e., it has an edge uv iff both directed edge uv and directed edge vu exist. If ignoring the directions of the edges in $LMST_1$, they call the graph $LMST_1^+$, i.e., it has an edge uv iff either directed edge uv or directed edge vu exists. They prove that the graph is connected, and has bounded degree 6. In [107], Li *et al.* also showed that graph $LMST_1^-$ and $LMST_1^+$ are actually planar. Then they extend the definition to k-hop neighbors, the union of all edges of all minimum spanning tree $MST(N_k(u))$ is the *k local minimum spanning tree*, denoted by $LMST_k$. For example, the 2 local minimum spanning tree can be constructed by the following algorithm.

Algorithm 7 *Construct Low Weight Structure $LMST_2$ by 2-hop Neighbors*

1. Each node u collects its two hop neighbors information $N_2(u)$ using a communication efficient protocol described in [25].
2. Each node u computes the Euclidean minimum spanning tree $MST(N_2(u))$ of all nodes $N_2(u)$, including u itself.
3. For each edge $uv \in MST(N_2(u))$, node u tells node v about this directed edge.
4. Node u keeps an edge uv if $uv \in MST(N_2(u))$ or $vu \in MST(N_2(v))$. Let $LMST_2^+$ be the final structure formed by all edges kept. [3]

In [107], they prove that structures $LMST_2$ ($LMST_2^+$ and $LMST_2^-$) are connected, planar, low-weighted, and have bounded node degree at most 6. In general, we can prove following theorems:

Theorem 17 *[107] Structure $LMST_k$ is connected, planar graph and with bounded node degree at most 6, for all $k \geq 1$. Structure $LMST_k$ is low-weighted, for all $k \geq 2$.*

Specifically, MST is a subgraph of $LMST_k$, $LMST_k \subseteq RNG'$.

Although the constructed structure $LMST_2$ has several nice properties such as being bounded degree, planar, and low-weighted, the communication cost of Algorithm 7 could be very large to save the computational cost of each node. The large communication costs are from collecting the two hop neighbors information $N_2(u)$ for each node u. Although the total communication of the protocol described in [25] is $O(n)$, the hidden constant is large.

[3] It keeps an edge if either node u or node v wants to keep it. Another option is to keep an edge only if both nodes want to keep it. Let $LMST_2^-$ be the structure formed by such edges.

C: Structure based on combining RNG' and LMST

We could improve the communication cost of collecting $N_2(u)$ by using a subset of two hop information without sacrificing any properties. Define $N_2^{RNG'}(u) = \{w \mid vw \in RNG' \text{ and } v \in N_1(u)\} \cup N_1(u)$. We describe our modified algorithm as follows.

Algorithm 8 *Construct Low Weight Structure IMRG by 2-hop Neighbors in RNG'*

1. Each node u tells its position information to its one-hop neighbors $N_1(u)$ using a local broadcast model. All nodes together construct the graph RNG' in a localized manner.

2. Each node u locally broadcasts its incident edges in RNG' to its one-hop neighbors. Node u listens to the messages from its one-hop neighbors.

3. Each node u computes the Euclidean minimum spanning tree $MST(N_2^{RNG'}(u))$ of all nodes $N_2^{RNG'}(u)$, including u itself.

4. For each edge $uv \in MST(N_2^{RNG'}(u))$, node u tells node v about this directed edge.

5. Node u keeps an edge uv if $uv \in MST(N_2^{RNG'}(u))$ or $vu \in MST(N_2^{RNG'}(v))$. Let $IMRG^+$ be the final structure formed by all edges kept. Similarly, the final structure is called $IMRG^-$ when edge $uv \in RNG'$ is kept iff $uv \in MST(N_2^{RNG'}(u))$ and $uv \in MST(N_2^{RNG'}(v))$. Here IMRG is the abbreviation of *Incident MST and RNG Graph.*

Notice that in the algorithm, node u constructs the local minimum spanning tree $MST(N_2^{RNG'}(u))$ based on the induced UDG of the point sets $N_2^{RNG'}(u)$. It is obvious that the communication cost of Algorithm 8 is at most $7n$.

We can show that structures $IMRG^+$ and $IMRG^-$ are still connected, planar, bounded degree, and low-weighted. They are obviously planar, and with bounded degree since both structures are still subgraphs of the modified relative neighborhood graph RNG'. Clearly, the constructed structures are a supergraph of the previous structures, i.e., $LSMT_2+ \subseteq IMRG^+$ and $LSMT_2^- \subseteq IMRG^-$, since Algorithm 8 uses less information than Algorithm 7 in constructing the local minimum spanning tree. Thus, these two structures $IMRG^+$ and $IMRG^-$ are still connected. We also prove the following lemma.

Lemma 18 *[107] Structures $IMRG^-$ and $IMRG^+$ are still low-weighted.*

Theorem 19 *[107] Algorithm 8 constructs structures $IMRG^-$ or $IMRG^+$ using at most $7n$ messages. The structures $IMRG^-$ or $IMRG^+$ are connected, planar, bounded degree, and low-weighted. Both $IMRG^-$ and $IMRG^+$ have node degree at most 6.*

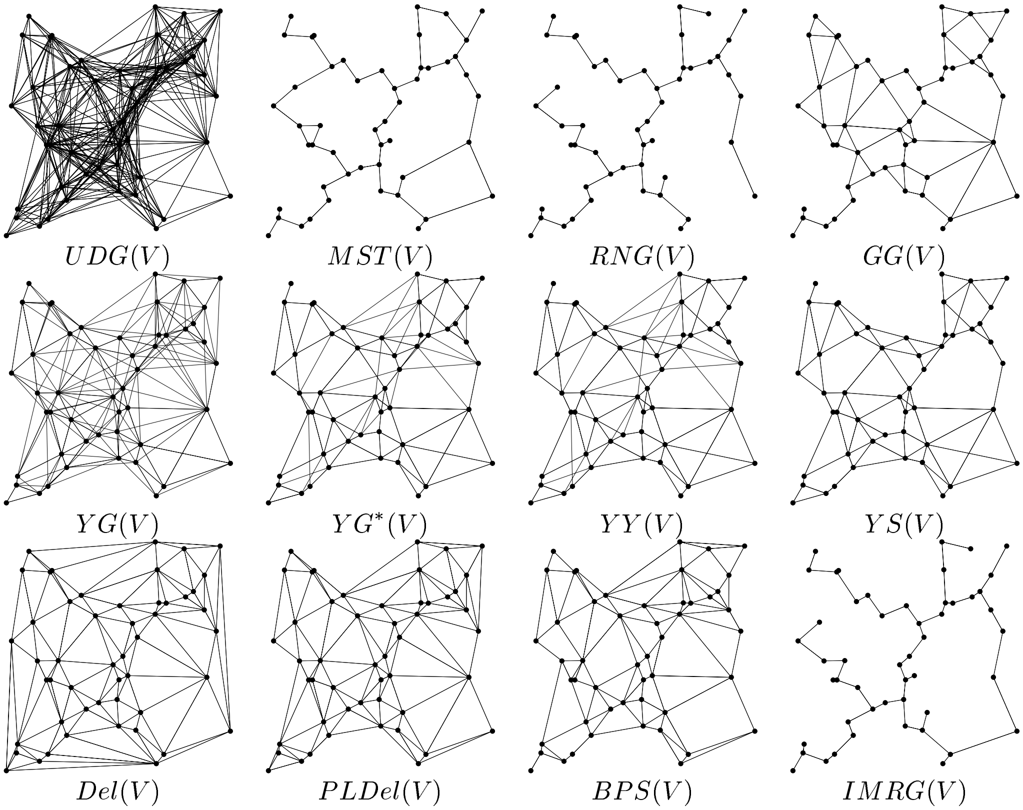
Recall that until now there is no efficient localized algorithm that can achieve all following desirable features: bounded degree, planar, low weight and spanner. It is still an open problem.

We then gave some concrete examples of the geometry structures introduced in the previous sections.

3.5 Fault Tolerance

Fault tolerance is one of the central challenges in designing the wireless sensor networks. Sensor node may be battery constrained or subject to hostile environments, so individual *node failure* will be a regular or common event. To make fault tolerance possible, first of all, the underlying network topology must have multiple disjoint paths to connect any two given wireless sensor devices. Here the path could be vertex disjoint or edge disjoint. We use the vertex disjoint multiple paths in this chapter considering the communication nature of the wireless sensor networks.

By setting the transmission range sufficiently large, the induced unit disk graph will be k-connected without doubt. However, since energy conservation is important to increase the life of the wireless sensor device, then the question is how to find the minimum transmission range such that the induced unit disk graph is multiply connected. Recently, applying stochastic geometry, Penrose [125, 126], Bettstetter [15], Li *et al* [108] studied



$UDG(V)$ $MST(V)$ $RNG(V)$ $GG(V)$

$YG(V)$ $YG^*(V)$ $YY(V)$ $YS(V)$

$Del(V)$ $PLDel(V)$ $BPS(V)$ $IMRG(V)$

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

how to set the transmission radius to achieve the k-connectivity with certain probability for a network when wireless nodes or sensors are uniformly and randomly distributed over a two-dimensional region. Due to space limit, we skip their results here, only concentrate on the topology control problem.

Remember that, in topology control, we try to maintain only a linear number of links using a localized construction method. However, this sparseness of the constructed network topology should not compromise on the fault tolerance and compromise too much on the power consumptions for communications. Therefore, in this section we study a localized method to control the network topology given a k-faults tolerant deployment of wireless sensor nodes such that the resulting topology is still fault tolerant but with much fewer communication links maintained. We show that the constructed topology has only linear number of links and is a length spanner.

Levcopoulos *et al.* [90] proposed some algorithms for constructing fault tolerant geometric spanners. Their algorithm can construct a spanner of degree $O(c^k)$, whose total edge length is bounded by $O(c^k)$ times the weight of a MST, and that is resilient to k edge or vertex faults. However, their algorithms are too complex to have a localized version.

Lukovszki [114] gave a method to construct a spanner that can sustain k-nodes or links failures for complete graph. Our topology control method [108] is based on this method and the Yao structure [164]. It is obvious that the Yao structure does not sustain k faults in a neighborhood of any node since each node only has at most p neighbors and one neighbor is selected in each cone at most. However, we can modify the Yao structure as follows such that the structure is k-fault tolerant.

Each node u defines any p equal-separated rays originated at u, thus defines p equal cones, where $p > 6$. In each cone, node u chooses the $k+1$ closest nodes in that cone, if there is any, and add directed links from u to these nodes. Ties are broken arbitrarily. Let $YG_{p,k+1}$ be the final topology formed by all nodes.

Obviously, the following theorem holds.

Theorem 20 *[108] The structure $YG_{p,k+1}$ can sustain k nodes faults if original unit disk graph is k node faults tolerant.*

We also show that the above structure approximates the original unit disk graph well. More specifically, we will show that it is a spanner even with k fault nodes.

Theorem 21 *[108] The structure $Y_{p,k+1}$ is a length spanner even with k nodes faults.*

Due to limited power and resource of wireless sensor nodes, wireless topologies always prefer to have bounded node degree, such that every wireless sensor nodes only keep constant neighbors. The node degree of the structure $Y_{p,k+1}$ is at most $p(k+1)$, where $p \geq 6$.

Another related problem is how to find small transmission range (power) for each node such that the resulted communication graph is k-connected. In [65], they called it *power-opitmal k-fault tolerance*. This problem is known to be NP-hard and related to the problem of *transimission power control* which we will discuss in Section 3.7. Some herristics [10, 134] for this problem have been proposed. Ramanathan and Rosales-Hain [134] consider the special case of 2-fault tolerance and provide a centralized spanning tree heruristic for minimizing the maximum transmit power.

Recently, Bahramgiri *et al.* [10] generalized the cone-based local heuristic of Wattenhofer *et al.* [93, 156] to slove the k-fault tolerance. We can prove that their resulted graph is also a length spanner even with k nodes faults (the proof is similar to ours [108]). However, their method does not bound the node degree. Figure 11(a) shows an example in which node u can have as many as neighbors even after applying their method. Then we give a careful enhancement of their protocol to bound the node degree. In Bahramgiri's method, they increase the power step by step until there is no gap greater than α between the successive neighbors or the power reaches the maximum power. They proved that if $\alpha \leq \frac{2\pi}{3k}$ then the resulted graph is k-connected. After applying their method, we can remove some links by the following method. For a node u, we divide its transmission range into $\frac{4\pi}{\alpha}$ equal cones (each cone have an angle $\alpha/2$). We select only one neighbor in each cone c if there is any, delete all other links. However, if for a cone c, one of its adjacent cones, say b, does not have any neighbors of u, we select the boundary neighbor v such that vu forms the smallest angle with cone b; if both adjacent cones of c are empty, we select *two* neighbors in c (close to the two boundary of cone c respectively); if c does not have empty adjacent cones, we can select any one of the neighbors. See Figure 11(b) for illustration. Since the gap between any two successive remaining neighbors is still not greater than α (except the empty cones), it is easy to

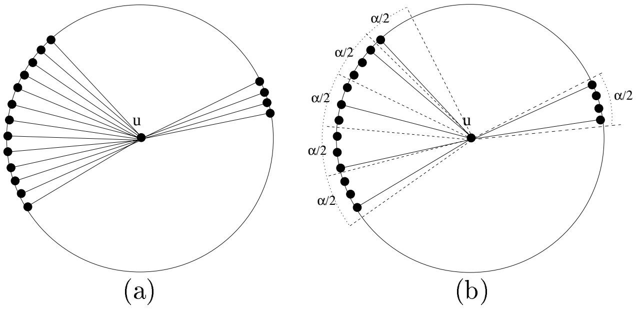


Figure 11: (a) node u does not have bounded degree in graph generated by Bahramgiri's protocol; (b) new method to bound node degree for Bahramgiri's protocol.

show that the constructed graph is still k-connected if $\alpha \leq \frac{2\pi}{3k}$. The node degree is bounded by $\frac{2\pi}{\frac{\alpha}{2}} = \frac{4\pi}{\alpha}$. When $\alpha = \frac{2\pi}{3k}$, the node degree is bounded by $6k$, which is almost the same as ours.

Both heuristics [10, 134] do not have provable bounds on the solution cost for power-opitmal k-fault tolerance problem. In [65], Haijaghayi *et al.* showed there are examples for which these heuristics perfrom arbitrarily worse than the opimal solution. Recently Lloyd *et al.* [113] presented a result which they prove gives an 8-approximation for 2-fault tolerance. Haijaghayi *et al.* then presented a more general result, some algorithms minimize power while maintaining k-fault tolerance with guaranteed approximation facotrs, in [65]. We will review them in Section 3.7.

3.6 Interference

In addition to spanner (which means connectivity and energy-efficiency) and other properties we discussed previously, we would like to have a topology with high capacity or throughput, such that it can route *as much as traffic* in the topology. One of the important issue affecting the throughput is interference. Modeling interference in a wireless environment is a complex task. The wireless medium is susceptible to path loss, noise, interference and blockages due to physical obstructions. Rajaraman [133] reviewed several models from path loss, bit-error rate to interference. In [62], Gupta and Kumar analyze the throughput of ad hoc networks under both the physical and protocol models of interference. For the detail definitions of these models, please refer to [62, 133].

In Rajaraman's review [133], he claimed the throughput of a topology depends on, among other factors, the level of interference inherent to the topology. Define the *interference number* of an edge e in a graph G to be the maximum number of other edges in G that interfere with e, in the sense of the interference model. Define the interference number of the topology to be the maximum interference number of all edges in G. A plausible goal then is to seek a topology with a small interference number. The particular interference number achievable, however, depends on the relative positions of the wireless nodes and their transmission radii.

Most of the proposed topology control protocols did not study the interference number of their topology theoretically, instead of, some of them showed the simulation results on network throughput. Recently, Jia, Rajaraman and Scheideler [72] showed the interference analysis of YaoYao structure ($YY_k(V)$). They used the protocol model from [62] as the interference model. First, they proved the following theorem to show the throughput achievable on $YY_k(V)$ is essentially limited only by its interference number, when compared with an optimal schedule on $UDG(V)$.

Theorem 22 *[72] Let I be the interference number of $YY_k(V)$. Let W denote a set of packets that are successfully delivered by an arbitrary schedule of packet transmissions in $UDG(V)$ in t steps. Then there exists a schedule of transmissions in $YY_k(V)$ that delivers W in $O(tI+n^2)$ steps. Thus, for sufficiently large t and W, the throughput achievable on $YY_k(V)$ is an $\Omega(1/I)$ fraction of the optimal.*

Then they established an upper bound on the interference number of $YY_k(V)$ for a random node distribution.

Theorem 23 *[72] If the n nodes are placed independently and uniformly at random in a unit square, the the interference number of $YY_k(V)$ is $O(logn)$ whp.*

For other topologies, the interference analysis are still open problems.

3.7 Transmission Power Control

In the previous sections, we have assumed that the transmission power of every node is equal and is normalized to one unit. We relax this assumption for a moment in this section. In other words, we assume that each node can adjust its transmission power according to its neighbors' positions. A natural question is then how to assign the transmission power for each node such that the wireless sensor network is connected with optimization criteria being minimizing the maximum (or total) transmission power assigned.

A transmission power assignment on the vertices in V is a function f from V into real numbers. The *communication graph*, denoted by G_f, associated with a transmission power assignment f, is a directed graph with V as its vertices and has a directed edge $\overrightarrow{v_iv_j}$ if and only if $||v_iv_j||^\beta \leq f(v_i)$. We call a transmission power assignment f *complete* if the communication graph G_f is strongly connected. Recall that a directed graph is strongly connected if, for any given pair of ordered nodes s and t, there is a directed path from s to t.

The *maximum-cost* of a transmission power assignment f is defined as $mc(f) = \max_{v_i \in V} f(v_i)$. And the *total-cost* of a transmission power assignment f is defined as $sc(f) = \sum_{v_i \in V} f(v_i)$. The min-max assignment problem is then to find a complete transmission power assignment f whose cost $mc(f)$ is the least among all complete assignments. The min-total assignment problem is to find a complete transmission power assignment f whose cost $sc(f)$ is the least among all complete assignments.

Given a graph H, we say the power assignment f is induced by H if

$$f(v) = \max_{(v,u) \in E} ||vu||^\beta,$$

where E is the set of edges of H. In other words, the power assigned to a node v is the largest power needed to reach all neighbors of v in H.

Transmission power control has been well-studied by peer researchers in the recent years. Monks *et al.* [120] conducted simulations which show that implementing power control in a multiple access environment can improve the throughput performance of the non-power controlled IEEE 802.11 by a factor of 2. Therefore it provides a compelling reason for adopting the power controlled MAC protocol in wireless network.

The min-max assignment problem was studied by several researchers [134, 137]. Let MST(V) be the Euclidean minimum spanning tree over a point set V. Both [134] and [137] use the power assignment induced by MST(V). The correctness of using minimum spanning tree is proved in [134]. Both algorithms compute the minimum spanning tree from the fully connected graph. Notice that Kruskal's or Prim's minimum spanning tree algorithm has time complexity $O(m + n \log n)$, where m is the number of edges of the graph. Thus, the approach by [134] and [137] has time complexity $O(n^2)$ in the worst case. In addition, different distributed implementation of this algorithm is not feasible because of the information each node has to store and process. In contrast, we can give a simple $O(n \log n)$ time complexity centralized algorithm to construct MST from RNG, which can also be implemented efficiently for distributed computation.

For an optimum transmission power assignment f_{opt}, call a link uv the *critical link* if $||uv||^\beta = mc(f_{opt})$. It was proved in [134] that the longest edge of the Euclidean minimum spanning tree MST(V) is always the critical link.

The best distributed algorithm [51, 54, 56] can compute the minimum spanning tree in $O(n)$ rounds using $O(m+n \log n)$ communications for a general graph with m edges and n nodes. The relative neighborhood graph, the Gabriel graph and the Yao graph all have $O(n)$ edges and contain the Euclidean minimum spanning tree. This implies the following theorem.

Theorem 24 *The distributed min-max assignment problem can be solved in $O(n)$ rounds using $O(n \log n)$ communications.*

The min-total assignment problem was studied by Kiroustis *et al.* [82] and by Clementi *et al.* [37, 38, 40]. Kiroustis *et al.* [82] first proved that the min-total assignment problem is *NP-hard* when the mobile nodes are deployed in a three-dimensional space. A simple 2-approximation algorithm based on the Euclidean minimum spanning tree was also given in [82]. The algorithm guarantees the same approximation ratio in any dimensions.

Then Clementi *et al.* [37, 38, 40] proved that the min-total assignment problem is still NP-hard when the mobile nodes are deployed in a two dimensional space.

Recently, Călinescu *et al.* gave a method that achieves better approximation ratio than the approach by the minimum spanning tree by using idea from the minimum Steiner tree.

A natural generalization of the connectivity requirement is k-fault tolerance or k-connectivity. As mentioned in Section 3.5, some researchers studied the power assignemnts of wireless nodes that minimize power while maintaining k-fault tolerance. As power-optimal connectivity is NP-hard, power-optimal k-fault tolerance is NP-hard as well.

Many of the best known approximation algorithms (such as [31]) are based on linear programming (LP) approaches. However, in [65], Haijaghayi *et al.* showed that for the min-total k-connectivity assignment problem, the natural integer LP formulation has an itergrality gap of $\Omega(\frac{n}{k})$, implying that there is no approximation algorithm based on LP with an approximation factor better than $\Omega(\frac{n}{k})$.

Some heursitics [10, 134] are proposed. Bahramgiri *et al.* [10] showed that the cone-based topology control (CBTC) algorithm of Wattenhofer *et al.* [93, 156] can be extended to slove the k-fault tolerance. In [65], the authors also constructed examples which demonstrate that the approximation factor for CBTC algorithm is at least $\Omega(\frac{n}{k})$.

Recently, Lloyd *et al.* [113] presented a result which they prove gives an centralized 8-approximation for min-total 2-fault tolerance assignment . Haijaghayi *et al.* [65] then presented a more general result, three algorithms minimize power while maintaining k-fault tolerance. The first algorithm gives an $O(k\alpha)$ approximation where α is the best approximation factor for the related prolbem in wired networks (the best α so far is in $O(\log k)$ [31]). The second algorithm is based on an approximation algorithm introduced by Kortsarz and Nutov [84]. It is more complicated and achieves $O(k)$ approximation for general graphs. Their first two algorithms are centralized algorithms. Then they presented two distributed approximation algorithms for the cases 2- and 3-connectivity in geometric graphs with approximation factors $2(4\times 2^{\beta-1}+1)$ and $2(1+7\times 2^{\beta-1}+12\times 4^{\beta-1})$. Both these algorithms use the distributed minimum spanning tree algorithm. In addition, they demostrated how to generalize these algorithms for k-connectivity in geometric graphs. However, their methods do not work for unit disk graphs, i.e., then the nodes transmission radius is bounded from above by a constant. It is still an open problem to achieve k-connectivity for UDG with objective of minimizing the total edge length.

3.8 Clustering, Virtual Backbone

While all the structures discussed so far are flat structures, there are another set of structures, called hierarchical structures, are used in wireless networks. Instead of all nodes are involved in relaying packets for other nodes, the hierarchical routing protocols pick a subset of nodes that server as the routers, forwarding packets for other nodes. The structure used to build this virtual backbone is usually the connected dominating set.

3.8.1 Centralized Methods

Guha and Khuller [61] studied the approximation of the connected dominating set problem for general graphs. They gave two different approaches, both of them guarantee approximation ratio of $\Theta(H(\Delta))$, where H is the harmonic function and Δ is the maximum node degree. As their approaches are for general graphs and thus do not utilize the geometry structure if applied to the wireless ad hoc networks. One approach is to grow a spanning tree that includes all nodes. The internal nodes of the spanning tree is selected as the final connected dominating set. This approach has approximation ratio $2(H(\Delta)+1)$. The other approach is first approximating the dominating set and then connecting the dominating set to a connected dominating set. They [61] proved that this approach has approximation ratio $\ln\Delta+3$.

One can also use the Steiner tree algorithm to connect the dominators. This straightforward method gives approximation ratio $c(H(\Delta)+1)$, where c is the approximation ratio for the unweighted Steiner tree problem. Currently, the best ratio is $1+\frac{\ln 3}{2}\simeq 1.55$, due to Robins and Zelikovsky [135].

By definition, any algorithm generating a maximal independent set is a clustering method. We first review the methods that approximates the maximum independent set, the minimum dominating set, and the minimum connected dominating set.

Hunt *et al.* [68] and Marathe *et al.* [115] also studied the approximation of the maximum independent set and the minimum dominating set for unit disk graphs. They gave the first PTASs for MDS in UDG. The method is based on the following observations: a maximal independent set is always a dominating set; given a square Ω with a fixed area, the size of any maximal dominating set is bounded by a constant C. Assume that there are n nodes in Ω. Then, we can enumerate all sets with size at most C in time $\Theta(n^C)$. Among these enumerated sets, the smallest dominating set is the minimum dominating set. Then, using the shifting strategy proposed by Hochbaum [67], they derived a PTAS for the minimum dominating set problem.

Since we have PTAS for minimum dominating set and the graph $VirtG$ connecting every pair of dominators within at most 3 hops is connected [154], we have an approximation algorithm (constructing a minimum spanning tree $VirtG$) for MCDS with approximation ratio $3+\epsilon$. Notice that, Berman *et al.* [14] gave an $\frac{4}{3}$ approximation method to connect a dominating set and Robins *et al.* [135] gave an $\frac{4}{3}$ approximation method to connect an independent set. Thus, we can easily have an $\frac{8}{3}$ approximation algorithm for MCDS, which was reported in [3]. Recently, Cheng *et al.* [30] designed a PTAS for MCDS in UDG. However, it is difficult to distributize their method efficiently.

3.8.2 Distributed Methods

Many distributed clustering (or dominating set) algorithms have been proposed in the literature [4, 5, 6, 34, 110, 111]. All algorithms assume that the nodes have distinctive identities (denoted by ID hereafter).

In the rest of section, we will interchange the terms cluster-head and dominator. The node that is not a cluster-head is also called *dominatee*. A node is called *white* node if its status is yet to be decided by the clustering algorithm. Initially, all nodes are white. The status of a node, after the clustering method finishes, could be *dominator* with color *black* or *dominatee* with color *gray*. The rest of this section is devoted for the distributed methods that approximates the minimum dominating set and the minimum connected dominating set for unit disk graph.

Clustering without Geometry Property

For general graphs, Jia *et al.* [73] described and analyzed some randomized distributed algorithms for the minimum dominating set problem that run in polylogarithmic time, independent of the diameter of the network, and that return a dominating set of size within a logarithmic factor from the optimum with high probability. Their best algorithm runs in $O(\log n \log \Delta)$ rounds with high probability, and every pair of neighbors exchange a constant number of messages in each round. The computed dominating set is within $O(\log \Delta)$ in expectation and within $O(\log n)$ with high probability. Their algorithm works for weighted dominating set also.

The method proposed by Das *et al.* [44, 141] contains three stages: approximating the minimum dominating set, constructing a spanning forest of stars, expanding the spanning forest to a spanning tree. Here the *stars* are formed by connecting each dominatee node to one of its dominators. The approximation method of MDS is essentially a distributed variation of the the centralized Chvatal's greedy algorithm [35] for set cover. Notice that the dominating set problem is essentially the set cover problem which is well-studied. It is then not surprise that the method by Das *et al.* [44, 141] guarantees a $H(\Delta)$ for the MDS problem, where H is the harmonic function and Δ is the maximum node degree.

While the algorithm proposed by Das *et al.* [44, 141] finds a dominating set and then grows it to a connecting dominating set, the algorithm proposed by Wu and Li [162, 163] takes an opposite approach. They first find a connecting dominating set and then prune out certain redundant nodes from the CDS. The initial CDS $\mathbb{C}$ contains all nodes that have at least two non-adjacent neighbors. A node u is said to be *locally redundant* if it has either a neighbor in $\mathbb{C}$ with larger ID which dominate all other neighbors of u, or two adjacent neighbors with larger ID which together dominates all other neighbors of u. Their algorithm then keeps removing all locally redundant nodes from $\mathbb{C}$. They showed that this algorithm works well in practice when the nodes are distributed uniformly and randomly, although no any theoretical analysis is given by them both for the worst case and for the average approximation ratio. However, it was shown by Alzoubi *et al.* [4] that the approximation ratio of this algorithm could be as large as $\frac{n}{2}$.

Stojmenovic *et al.* [146] proposed several synchronized distributed constructions of connecting dominating set. In their algorithms, the connecting dominating set consists of two types of nodes: clusterhead and border-nodes (also called gateway or connectors elsewhere). The clusterhead nodes are just a maximal independent set, which is constructed as follows. At each step, all white nodes which have the lowest *rank* among all white neighbors are colored black, and the white neighbors are colored gray. The ranks of the white nodes is updated if necessary. Here, the following rankings of a node are used in various methods: the ID only [34, 110], the ordered pair of degree and ID [29], and an ordered pair of degree and location [146]. After the clusterhead nodes are selected, border-nodes are selected to connect them. A node is a border-node if it is not a clusterhead and there are at least two clusterheads within its 2-hop neighborhood. It was shown by [4] that the worst case approximation ratio of this method is also $\frac{n}{2}$, although it works well in practice.

In [11, 12], Basagni *et al.* studied how to maintain the clustering in mobile wireless ad hoc networks. It uses a general *weight* as a criterion for selecting the node as the clusterhead, where the weight could be any criteria used before.

Clustering with Geometry Property

Notice that none of the above algorithm utilizes the geometry property of the underlying unit disk graph. Recently, several algorithms were proposed with a constant worst case approximation ratio by taking advantage of the geometry properties of the underlying graph. These methods typically use two messages similar to IamDominator and IamDominatee, and typically have the following procedures: a white node claims itself to be a dominator if it has the smallest ID among all of its white neighbors, if there is any, and broadcasts IamDominator to its 1-hop neighbors. A white node receiving IamDominator message marks itself as dominatee and broadcasts IamDominatee to its 1-hop neighbors. The set of dominators generated by the above method is actually a maximal independent set. Here, we assume that each node knows the IDs of all its 1-hop neighbors, which can be achieved by asking each node to broadcast its ID to its 1-hop neighbors initially. This approach of constructing MIS is well-known. For example, Stojmenovic *et al.* [146] also used this method to compute the MIS.

The second step of backbone formation is to find some *connectors* (also called *gateways*) among all the dominatees to connect the dominators. Then the connectors and the dominators form a *connected dominating set*. Recently, Wan, *et al.* [150] proposed a communication efficient algorithm to find connectors based on the fact that there are only a constant number of dominators within k-hops of any node. The following observation is a basis of several algorithms for CDS. After clustering, one dominator node can be connected to many dominatees. However, it is well-known that a dominatee node can only be connected to at most *five* dominators in the unit disk graph model. Generally, it was shown in [150, 154] that for each node (dominator or dominatee), there are at most a constant number of dominators that are at most k units away.

Lemma 25 *For every node v, the number of dominators inside the disk centered at v with radius k-units is bounded by a constant $\ell_k < (2k+1)^2$.*

Lemma 26 *Given a dominating set S, let $VirtG$ be the graph connecting all pairs of dominators u and v if there is a path in UDG connecting them with at most* 3 *hops. $VirtG$ is connected.*

It is natural to form a connected dominating set by finding connectors to connect any pair of dominators u and v if they are connected in $VirtG$. This strategy is also adopted by Wan, *et al.* [150]. Notice that, in the approach by Stojmenovic *et al.* [146], they set any dominatee node as the connector if there are two dominators within its 2-hop neighborhood. This approach is very pessimistic and results in very large number of connectors in the worst case [4]. Instead, Wan *et al.* [2]suggested to find only one unique shortest path to connect any two dominators that are at most three hops away. Wang and Li [154] and Alzoubi *et al.* [150] discussed in detail some approaches to optimize the communication cost and the memory cost. In [2, 154], they proved the following theorem.

Theorem 27 *The number of connectors found by this algorithm is at most ℓ_3 times of the minimum. The size of the connected dominating set found by this algorithm is within a small constant factor of the minimum.*

The graph constructed by this algorithm is called a CDS graph (or *backbone* of the network). If we also add all edges that connect all dominatees to their dominators, the graph is called extended CDS, denoted by CDS'. It was shown in [2, 154] that the CDS' graph is a sparse spanner in terms of both hops and length with factors 3 and 6, meanwhile CDS has a bounded node degree $\max(\ell_3, 5 + \ell_2)$. See [154] for detailed proofs.

Several routing algorithms require the underlying topology be planar. Notice in the formation algorithm of CDS, we do not use any geometry information. The resulting CDS maybe non-planar graph. Even using some geometry information, the CDS still is not guaranteed to be a planar graph. Then Li *et al.* [154] proposed a method to make the graph CDS planar without losing the spanner property of the backbone. Their method applies the localized Delaunay triangulation [98] on top of the induced graph from CDS, denoted by ICDS. It was proved in [98] that $LDel(G)$ is a spanner if G is a unit disk graph. Notice that ICDS is a unit disk graph defined over all dominators and connectors. Consequently, $LDel(ICDS)$ is a spanner in terms of length.

4 Localized Routing

The geometric nature of the multi-hop wireless sensor networks allows a promising idea: localized geometric routing (or localized routing) protocols. A routing protocol is *localized* if the decision to which node to forward a packet is based only on:

- The information in the header of the packet. This information includes the source and the destination of the packet, but more data could be included, provided that its total length is bounded.
- The local information gathered by the node from a small neighborhood. This information includes the set of 1-hop neighbors of the node, but a larger neighborhood set could be used provided it can be collected efficiently.

Randomization is also used in designing the protocols. A routing is said to be *memory-less* if the decision to which node to forward a packet is solely based on the destination, current node and its neighbors within some constant hops. Localized routing is sometimes called in the literature *stateless* [77, 78], *online* [16, 18], or *distributed* [145].

In order to make the localized geometric routing work, the source node has to learn the current (or approximately current) location of the destination node. Notice that, for sensor networks collecting data, the destination node is often fixed, thus, location service is not needed in these applications. However, the help of a *location service* is needed in most application scenarios. Mobile nodes register their locations to the location service. When a source node does not know the position of the destination node, it queries the location service to get that information. In cellular networks, there are dedicated position severs. It will be difficult to implement the centralized approach of location services in wireless sensor networks. First, for centralized approach, each node has to know the position of the node that provides the location services, which is a chicken-and-egg problem. Second, the dynamic nature of the wireless sensor networks makes it very unlikely that there is at least one location server available for each node. Recently, algorithms for distributed location services are studied in [13, 64, 92, 142]. Due to space limit, we omit the location service problem here. See [96] for detailed review.

4.1 Simple Heuristics

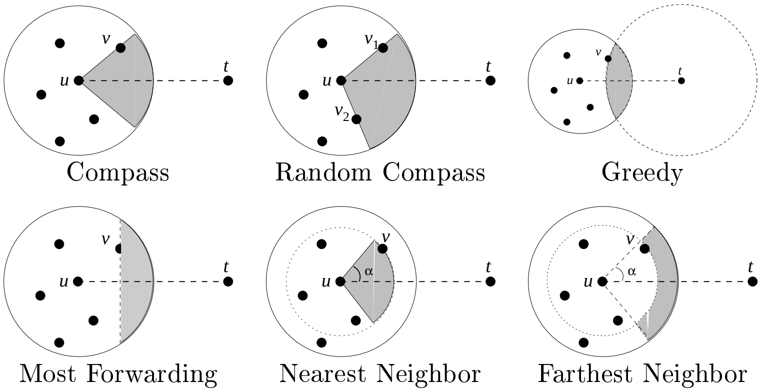
We summarize some localized geometric routing protocols proposed in the networking and computational geometry literature.

Compass Routing Let t be the destination node. Current node u finds the next relay node v such that the angle $\angle vut$ is the smallest among all neighbors of u in a given topology. See[85].

Random Compass Routing Let u be the current node and t be the destination node. Let v_1 be the node on the above of line ut such that $\angle v_1ut$ is the smallest among all such neighbors of u. Similarly, we define v_2 to be nodes below line ut that minimizes the angle $\angle v_2ut$. Then node u randomly choose v_1 or v_2 to forward the packet. See[85].

Greedy Routing Let t be the destination node. Current node u finds the next relay node v such that the distance $||vt||$ is the smallest among all neighbors of u in a given topology. See [20].

Most Forwarding Routing (MFR) Current node u finds the next relay node v such that $||v't||$ is the smallest among all neighbors of u in a given topology, where v' is the projection of v on segment ut. See [145].



Compass Random Compass Greedy

Most Forwarding Nearest Neighbor Farthest Neighbor

Figure 12: Various localized routing methods. Shaded area is empty and v is next node.

Nearest Neighbor Routing (NN) Given a parameter angle α, node u finds the nearest node v as forwarding node among all neighbors of u in a given topology such that $\angle vut \leq \alpha$.

Farthest Neighbor Routing (FN) Given a parameter angle α, node u finds the farthest node v as forwarding node among all neighbors of u in a given topology such that $\angle vut \leq \alpha$.

Greedy-Compass Current node u first finds the neighbors v_1 and v_2 such that v_1 forms the smallest counterclockwise angle $\angle tuv_1$ and v_2 forms the smallest clockwise angle $\angle tuv_2$ among all neighbors of u with the segment ut. The packet is forwarded to the node of $\{v_1, v_2\}$ with minimum distance to t. See [18, 121]

Notice that it is shown in [20, 85] that the compass routing, random compass routing and the greedy routing guarantee to deliver the packets from the source to the destination if Delaunay triangulation is used as network topology. They proved this by showing that the distance from the selected forwarding node v to the destination node t is less than the distance from current node u to t. However, the same proof cannot be carried over when the network topology is Yao graph, Gabriel graph, relative neighborhood graph, and the localized Delaunay triangulation. When the underlying network topology is a planar graph, the right hand rule or face routing is often used to guarantee the packet delivery after simple localized routing heuristics fail [20, 145, 78, 88, 86]. We will discuss them in next section.

Theorem 28 *[121] The greedy routing guarantees the delivery of the packets if the Delaunay triangulation is used as the underlying structure. The compass routing guarantees the delivery of the packets if the regular triangulation is used as the underlying structure. There are triangulations (not Delaunay) that defeat these two schemes. The greedy-compass routing works for all triangulations, i.e., it guarantees the delivery of the packets as long as there is a triangulation used as the underlying structure. Every oblivious routing method is defeated by some convex subdivisions.*

Here a triangulation is *regular triangulation* if it is the projection of the lower convex hull of some 3-dimensional polytopes P into the X-Y plane. Delaunay triangulation is a special regular triangulation in which all the vertices of P are on a paraboloid $z^2 = x^2 + y^2$. Another interesting triangulation is *greedy triangulation* which is constructed by adding edges in the increasing order of their lengths to avoid crossing edges. They [121] also study the localized routing for greedy triangulation. As the greedy triangulation can not be constructed locally or very efficiently in a distributed manner. We omit that part in this book. It is easy to see that there is no memoryless routing method that works in the unit disk graph.

4.2 Right Hand Rule and Face Routing

Right hand rule is long-known method for traversing a graph (in analogy to following the right hand wall in a maze). And it has been used in some wireless routing protocols [20, 145, 78, 77]. The rule states that when arriving at node x from node y, the next edge traversed is the next one sequentially counterclockwise about x

from edge xy. In the example shown in Figure 13, x will forward the packet to z following right hand rule, traversing face P. It is known that the right hand rule traverses the interior of a closed polygonal region (a face) in clockwise edge order. And it traverses an exterior region in counterclockwise edge order. In general, right hand rule is applied in planar graphs (in which no edges intersect each other). In [77] give a *no-crossing heuristic* to deal with the case where edges cross.

Applying the right hand rule in planar graphs, a routing protocol called *Face Routing* is proposed by [85] (in the paper they call the algorithm *Compass Routing II*). We consider a planar graph G. The nodes and edges of graph G partition the Euclidean plane into contiguous regions called the *faces* of G. The main idea of the face routing is to walk along the faces which are intersected by the line segment st between the source s and the destination t. In each face, it uses the right hand rule to explore the boundaries. On its way around a face, the algorithm keeps track of the points where it crosses the line st. Having completely surrounded a face, the algorithm returns to one of these intersections lying closest to t, where it proceeds by exploring the next face closer to t. Figure 13 gives an illustration. See [85, 87] for detailed algorithms. They also proved that the face routing algorithm guarantees to reach the destination t after traversing at most $O(n)$ edges where n is the number of nodes.

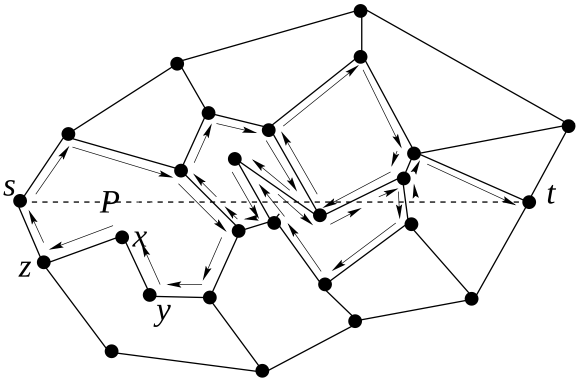


Figure 13: An illustration of the face routing algorithm.

Though face routing terminates in linear time, it is not satisfactory, since already a very simple flooding algorithm will terminate in $O(n)$ steps. Then in [87], Kuhn *et al.* proposed a new method called *Adaptive Face Routing* (AFR), in which, restricted search areas are used to avoid exploring the complete boundary of faces. The idea is as follows. The exploration of faces is restricted to an ellipse area. The ellipse size is set to an initial estimate of the optimal path length. If face routing fails to reach the destination (when it reach the ellipse, it has to turn back), the algorithm will restart with a bounding ellipse of doubled size. They proved that the algorithm will finally find a path to t if s and t are connected. Also the number of steps of AFR is bounded by $O(c^2(p^*))$, where p^* is an optimal path and $c(p^*)$ is the cost of that path. In their proof, they assumed the unit disk graph is a civilized graph. Finally they give a tight lower bound by showing that *any localized* geometric routing algorithm has worst-case cost $O(c^2(p^*))$.

Recently, Kuhn *et al.* [88] extend the *Adaptive Face Routing* to a routing algorithm called *Other Adaptive Face Routing* (OAFR). Instead of changing to the next face at the "best" intersection of the face boundary with st, OAFR returns to the boundary point closest to the destination. They proved the cost of OAFR is also bounded by $O(c^2(p^*))$ which is asymptotically optimal.

4.3 Combine Face Routing with Greedy Routing

Greedy routing was used in early routing protocol for wireless networks. However, it is easy to construct a simple example to show that greedy algorithm will not succeed to reach the destination but fall into a local minimum, a node without any "better" neighbors. Then a natural approach to improve the potential of greedy routing for practical purposes is to combining greedy routing and face routing (or right hand rule) to recover the routing after simple greedy routing fail in local minimum. Many wireless protocols used this approach [20, 78, 88, 86, 145, 165].

Greedy Perimeter Stateless Routing (GPRS) [77, 78] is one of the famous routing protocol for wireless networks. It uses RNG or GG as the planar routing topology, then combining greedy and right hand rule to forward packets in the network. It works as follows. When a node receives a greed-mode packet, it searches its neighbor table for a neighbor who is closer to the destination t. If there is one, it will forward it to that neighbor. When no neighbor is closer, the node marks the packet into perimeter mode. GPSR forwards perimeter-mode packets using a simple planar graph traversal (right-hand rule). When a packet enters perimeter mode, GPSR records in the packet the location L_p. Then when receiving a perimeter-mode packet, GPSR will first compare it with forwarding node's location. GPRS returns a packet to greedy mode of the distance from the forwarding node to t is less than that from L_p to t. For more detail, please refer to [78, 77]. GPRS can guarantee the delivery of the packets when the underlying network topology is a planar graph.

Recently, Kuhn *et al.* [88] proposed a new algorithm to combine greedy routing with their *Other Adaptive Face Routing* (OAFR). They called the new method *Greedy Other Adaptive Face Routing* (GOAFR). The idea is similar to GPSR. When greedy method falls in local minimum, GOAFR uses OAFR to recover the routing. Same as for AFR, they proved the cost of GOAFR is bounded by $O(c^2(p^*))$ which is asymptotically optimal. In addition, they show that the algorithm is also average-case efficient through extensive simulations. In [88], the authors showed simulations of a variety of face routing algorithms and their combinations with a greedy approach. Notice that unlike GPSR, when doing face routing in GOAFR, it does not return to greedy method until OAFR completely finishes the exploration of the face. This may affect the efficiency of routing. In [86], Kuhn *et al.* uses an "early fallback" technique to return to greedy routing as soon as possible. The new algorithm called GOAFR$^+$. It employs two counters p and q to keep track of how many of the nodes visited during the current face routing phase are located closer (counted by p) and how many are not closer (counted by q) to the destination than the starting point of the current face routing phase. When a certain fallback condition holds, GOAFR$^+$ directly falls back to greedy mode. This modification makes an obvious improvement for the average case performance. Their theoretical analysis also proves that GOAFR$^+$ is asymptotically optimal in the worst case.

4.4 Routing on Delaunay Triangulation

With respect to localized routing, there are several ways to measure the quality of the protocol. In Kuhn's analysis, they used the number of steps (hops) in a path to measure the quality of their routing methods. Given the scarcity of the power resources in wireless sensor networks, minimizing the total power used is imperative. A stronger condition is to minimize the total Euclidean distance traversed by the packet. Morin *et al.* [18, 121] also studied the performance ratio of previously studied localized routing methods. They proved that none of the previous proposed heuristics guarantees a constant ratio of the traveled distance of a packet compared with the minimum. They gave the first localized routing algorithm such that the traveled distance of a packet from u to v is at most a constant factor of $\|uv\|$ when the Delaunay triangulation is used as the underlying structure.

Their algorithm is based on the proof of the spanner property of the Delaunay triangulation [48]. Without loss of generality, let $b_0 = u$, b_1, b_2, $\cdots$, b_{m-1}, $b_m = v$ be the vertices corresponding to the sequence of Voronoi regions traversed by walking from u to v along the segment uv. If a Voronoi edge or a Voronoi vertex happens to lie on the segment uv, then choose the Voronoi region lying above uv. See Figure 14. Given two nodes u and v, $tunnel(u, v)$ is defined as the collection of triangles that intersect the segment uv. The sequence of nodes b_i, $0 \leq i \leq m$, defines a path from u to v. In general, they [48] refer to the path constructed this way between some nodes u and v as the *direct DT path* from u to v.

Assume that line uv is the x-axis. The path constructed by Dobkin *et al.* uses the direct DT path as long as it is above the x-axis. Assume that the path constructed so far has brought us to some node b_i such that b_i is above uv, and b_{i+1} is below uv. Let j be the least integer larger than i such that b_j is above uv. Notice that here j exists because $b_m = v$ is on uv. Then the path constructed by Dobkin *et al.* uses either the direct DT path to b_j or takes a *shortcut*. See [48] for more detail about the condition when to choose the direct DT path from b_i to b_j, when to choose the shortcut path from b_i to b_j, and how the short-cut path is defined.

Bose and Morin basically use sort of binary search method to find which path is better. Refer [121] for more detail of finding the path. However, their algorithm needs the Delaunay triangulation as the underlying structure which is expensive to construct in wireless ad hoc networks. In [19], they further extent their method to any triangulations satisfying the diamond property. Here, a triangulation satisfying the diamond property if

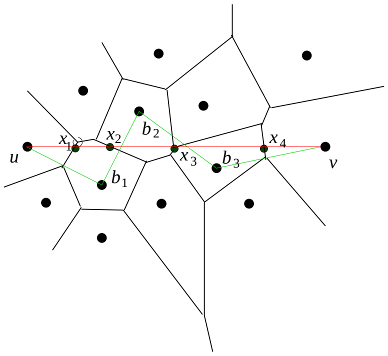


Figure 14: There is a good approximation path using the edges of $tunnel(u, v)$.

for every edge uv in the triangulation, either $\triangle uvw_1$ or $\triangle uvw_2$ is empty of other vertices, where w_i satisfying $\angle w_i uv = \angle w_i vu = \frac{\pi}{6}$, for $i = 1, 2$.

Li *et al.* [106] showed that the local Delaunay triangulation *PLDel* can be used to approximate the Delaunay triangulation *Del* almost always when the network is connected and the sensor nodes are randomly deployed. Consequently, the method by Bose *et al.* [121] can be used on local Delaunay triangulation almost always.

Localized routing protocols support mobility by eliminating the communication intensive task of updating the routing tables. But mobility can affect the localized routing protocols, in both the performance and the guarantee of delivery. There is no work so far to design protocols with guaranteed delivery when the network topology changes during the routing.

5 Broadcasting

Before this section, we only consider the unicast routing protocols, however in wireless networks broadcast is a very important operation, as it provides an efficient way of communication that does not require global information and functions well in the case of changing topologies. Although many broadcast/multicast algorithms [32, 71, 112, 129, 149, 159, 160] have be proposed for wireless ad hoc networks, most of them are not power-aware. Not until recently have research efforts been made to devise power-efficient multicast/broadcast algorithms for wireless networks. In [94], Li and Hou provide a detailed taxonomy of existing work. Here, we simply categorize existing work into two groups: centralized methods and localized methods.

5.1 Centralized Methods

5.1.1 Assumptions

Minimum-energy broadcast/multicast routing in a simple ad hoc networking environment has been addressed by the pioneering work in [36, 39, 83, 157]. To assess the complexities *one at a time*, the nodes in the network are assumed to be randomly distributed in a two-dimensional plane and there is no mobility. Nevertheless, as argued in [157], the impact of mobility can be incorporated into this static model because the transmitting power can be adjusted to accommodate the new locations of the nodes as necessary. In other words, the capability to adjust the transmission power provides considerable "elasticity" to the topological connectivity, and hence may reduce the need for hand-offs and tracking. In addition, as assumed in [157], there are sufficient bandwidth and transceiver resources. Under these assumptions, centralized (as opposed to distributed) algorithms were presented by [157, 109, 42, 91] for minimum-energy broadcast/multicast routing. These centralized algorithms, in this simple networking environment, are expected to serve as the basis for further studies on distributed algorithms in a more practical network environment, with limited bandwidth and transceiver resources, as well as the node mobility.

5.1.2 Centralized Methods

Some centralized methods are based on optimization. The scheme proposed in [116] is built upon an alternate search based paradigm in which the minimum-cost broadcast/multicast tree is constructed by a search process. Two procedures are devised to check the viability of a solution in the search space. Preliminary experimental results show that this method renders better solutions than BIP, though at a higher computational cost. Liang [109] showed that the minimum-energy broadcast tree problem is NP-complete, and proposed an approximate algorithm to provide a bounded performance guarantee for the problem in the general setting. Essentially they reduce the minimum-energy broadcast tree problem to an optimization problem on an auxiliary weighted graph and solve the optimization problem so as to give an approximate solution for the original problem. They also proposed another algorithm that yields better performance under a special case. Das et al. [42] proposed an evolutionary approach using genetic algorithms. The same authors also presented in [43] three different integer programming models which can be used to find the solutions to the minimum-energy broadcast/ multicast problem. The major drawback of optimization based schemes are, however, that they are centralized and require the availability of global topological information.

Some centralized methods are based on greedy heuristics. Three greedy heuristics were proposed in [157] for the minimum-energy broadcast routing problem: MST (minimum spanning tree), SPT (shortest-path tree), and BIP (broadcasting incremental power). The MST heuristic first applies the Prim's algorithm to obtain a MST, and then orient it as an arborescence rooted at the source node. The SPT heuristic applies the Dijkstra's algorithm to obtain a SPT rooted at the source node. The BIP heuristic is the node version of Dijkstra's algorithm for SPT. It maintains, throughout its execution, a single arborescence rooted at the source node. The arborescence starts from the source node, and new nodes are added to the arborescence one at a time on the minimum incremental cost basis until all nodes are included in the arborescence. The incremental cost of adding a new node to the arborescence is the minimum additional power increased by some node in the current arborescence to reach this new node. The implementation of BIP is based on the standard Dijkstra's algorithm, with one fundamental difference on the operation whenever a new node q is added. Whereas the Dijkstra's algorithm updates the node weights (representing the current knowing distances to the source node), BIP updates the cost of each link (representing the incremental power to reach the head node of the directed link). This update is performed by subtracting the cost of the added link pq from the cost of every link qr that starts from q to a node r not in the new arborescence. They have been evaluated through simulations in [157], but little is known about their analytical performances in terms of the approximation ratio. Here, the approximation ratio of a heuristic is the maximum ratio of the energy needed to broadcast a message based on the arborescence generated by this heuristic to the least necessary energy by any arborescence for any set of points.

For a pure illustration purpose, another slight variation of BIP was discussed in detail in [151]. This greedy heuristic is similar to the Chvatal's algorithm [35] for the set cover problem and is a variation of BIP. Like BIP, an arborescence, which starts with the source node, is maintained throughout the execution of the algorithm. However, unlike BIP, many new nodes can be added one at a time. Similar to the Chvatal's algorithm [35], the new nodes added are chosen to have the minimal *average* incremental cost, which is defined as the ratio of the minimum additional power increased by some node in the current arborescence to reach these new nodes to the number of these new nodes. They called this heuristic as the Broadcast Average Incremental Power (BAIP). In contrast to the $1 + \log m$ approximation ratio of the Chvatal's algorithm [35], where m is the largest set size in the Set Cover Problem, they showed that the approximation ratio of BAIP is at least $\frac{4n}{\ln n} - o(1)$, where n is the number of receiving nodes.

Wan *et al.* [151] showed that the approximation ratios of MST and BIP are between 6 and 12 and between $\frac{13}{3}$ and 12 respectively; on the other hand, the approximation ratios of SPT and BAIP are at least $\frac{n}{2}$ and $\frac{4n}{\ln n} - o(1)$ respectively, where n is the number of nodes. We then discuss in detail of their proof techniques in next section.

The Iterative Maximum-Branch Minimization (IMBM) algorithm was another effort [91] to construct power-efficient broadcast trees. It begins with a basic broadcast tree in which the source directly transmits to all other nodes. Then it attempts to approximate the minimum-energy broadcast tree by iteratively replacing the maximum branch with less-power, more-hop alternatives.

Both BIP and IMBM operate under the assumption that the transmission power of each node is unconstrained, i.e., every node can reach every other node. Both algorithms are centralized in the sense that they require: (a) the source node needs to know the position/distance of every other node; and (b) each node needs to know its downstream, on-tree neighbors so as to propagate broadcast messages. As a result, it may be difficult to extend both algorithms into distributed versions, as a significant amount of information is required to be exchanged among nodes.

5.1.3 Theoretical Analysis of Minimum-energy Broadcast

Any broadcast routing is viewed as an arborescence (a directed tree) T, rooted at the source node of the broadcasting, that spans all nodes. Let $f_T(\mathbf{p})$ denote the transmission power of the node $\mathbf{p}$ required by T. For any leaf node $\mathbf{p}$ of T, $f_T(\mathbf{p}) = 0$. For any internal node $\mathbf{p}$ of T,

$$f_T(\mathbf{p}) = \max_{\mathbf{pq} \in T} \|\mathbf{pq}\|^{\beta},$$

in other words, the β-th power of the longest distance between $\mathbf{p}$ and its children in T. The total energy required by T is $\sum_{\mathbf{p} \in P} f_T(\mathbf{p})$. Thus the minimum-energy broadcast routing problem is different from the conventional link-based minimum spanning tree (MST) problem. Indeed, while the MST can be solved in polynomial time by algorithms such as Prim's algorithm and Kruskal's algorithm [41], it is still unknown whether the minimum-energy broadcast routing problem can be solved in polynomial time. In its general graph version, the minimum-energy broadcast routing can be shown to be NP-hard [57], and even worse, it can not be approximated within a factor of $(1-\epsilon)\log\Delta$, unless $NP \subseteq DTIME\left[n^{O(\log\log n)}\right]$, where Δ is the maximal degree and ϵ is any arbitrary small positive constant. However, this intractability of its general graph version does not necessarily imply the same hardness of its geometric version. In fact, as shown later, its geometric version can be approximated within a constant factor. Nevertheless, this suggests that the minimum-energy broadcast routing problem is considerably harder than the MST problem. Recently, Clementi *et al.* [36] proved that the minimum-energy broadcast routing problem is a NP-hard problem and obtained a parallel but weaker result to those of [151].

Wan *et al.* [151] gave some lower bounds on the approximation ratios of MST and BIP by studying some special instances in [151]. Their deriving of the upper bounds relies extensively on the geometric structures of Euclidean MSTs. A key result in [151] is an upper bound on the parameter $\sum_{e \in mst(P)} \|e\|^2$ for any finite point set P of radius one. Note that the supreme of the total edge lengths of $mst(P)$, $\sum_{e \in mst(P)} \|e\|$, over all point sets P of radius one is infinity. However, the parameter $\sum_{e \in mst(P)} \|e\|^2$ is bounded from above by a constant for any point set P of radius one. They use c to denote the supreme of $\sum_{e \in mst(P)} \|e\|^2$ over all point sets P of radius one. The constant c is at most 12; see [151].

Theorem 29 *[151]* $6 \le c \le 12$.

The proof of this theorem involves complicated geometric arguments; see [151] for more detail. Note that for any point set P of radius one, the length of each edge in $mst(P)$ is at most one. Therefore, Theorem 29 implies that for any point set P of radius one and any real number $\beta \ge 2$,

$$\sum_{e \in mst(P)} \|e\|^{\beta} \le \sum_{e \in mst(P)} \|e\|^2 \le c \le 12.$$

The next theorem proved in [151] explores a relation between the minimum energy required by a broadcasting and the energy required by the Euclidean MST of the corresponding point set.

Lemma 30 *[151] For any point set P in the plane, the total energy required by any broadcasting among P is at least $\frac{1}{c}\sum_{e \in mst(P)} \|e\|^{\beta}$.*

Consider any point set P in a two-dimensional plane. Let T be an arborescence oriented from some $mst(P)$. Then the total energy required by T is at most $\sum_{e \in T_{\mathbf{p}}} \|e\|^{\beta}$. From Lemma 30, this total energy is at most c times the optimum cost. Thus the approximation ratio of the link-based MST heuristic is at most c. Together with Theorem 29, this observation leads to the following theorem.

Theorem 31 *[151] The approximation ratio of the link-based MST heuristic is at most c, and therefore is at most 12.*

In addition, they derived an upper bound on the approximation ratio of the BIP heuristic. Once again, the Euclidean MST plays an important role.

Lemma 32 *[151] For any broadcasting among a point set P in a two-dimensional plane, the total energy required by the arborescence generated by the BIP algorithm is at most $\sum_{e \in mst(P)} \|e\|^{\beta}$.*

5.2 Localized Methods

The centralized algorithms do not consider computational and message overheads incurred in collecting global information. Several of them also assume that the network topology does not change between two runs of information exchange. These assumptions may not hold in practice, since the network topology may change from time to time, and the computational and energy overheads incurred in collecting global information may not be negligible. This is especially true for large-scale sensor networks where the topology is changing dynamically due to the changes of position, energy availability, environmental interference, and failures, which implies that centralized algorithms that require global topological information may not be practical.

Santivanez et al. [138] show that flooding is a good solution for the sake of scalability and simplicity. Several flooding techniques for wireless networks have been proposed [66, 132, 144], each with respect to certain optimization criterion. However, none of them takes advantage of the feature that the transmission power of a node can be adjusted.

Some distributed heuristics are proposed, such as [1, 24, 158]. Most of them are based on distributed MST method. A possible drawback of these distributed method is that it may not perform well under frequent topological changes as it relies on information that is multiple hops away to construct the MST. Refer to [94] for more detail. The relative neighborhood graph, the Gabriel graph and the Yao graph all have $O(n)$ edges and contain the Euclidean minimum spanning tree. This implies that we can construct the minimum spanning tree using $O(n \log n)$ messages.

Cartigny et al. [27] proposed a localized algorithm, called RBOP [27] that is built upon the notion of relative neighborhood graph (RNG). In RBOP, the broadcast is initiated at the source and propagated, following the rules of neighbor elimination, on the topology represented by RNG. Simulation results show that the performance degradation could be as high as 100% as compared to BIP. Li and Hou [94] proposed another localized algorithm, called BLMST, which basically uses LMST as the broadcast topology. Their simulations show the performance of BLMST is much better than that of BROP, and comparable to that of BIP.

However, as shown in [97] and Section 3.4 (by Figure 8), the total weights of RNG and LMST could still be as large as $O(n)$ times of the total weight of MST. Given a graph G, let $\omega_b(G) = \sum_{e \in G} \|e\|^b$. Then $\omega_1(RNG) = \Theta(n) \cdot \omega_1(MST)$ and $\omega_1(LMST) = \Theta(n) \cdot \omega_1(MST)$. In Section 3.4.2, we describe three low weight planar graphs: H, $LSMT_2$ and $IMRG$. All of the three low weight planar structures can be constructed by localized methods, and the total communication costs are $O(n)$. It is easy to show that the energy consumption using those structures are within $O(n^{\beta-1})$ of the optimum, i.e., $\omega_\beta(H) = O(n^{\beta-1}) \cdot \omega_\beta(MST)$, $\omega_\beta(LMST_2) = O(n^{\beta-1}) \cdot \omega_\beta(MST)$, $\omega_\beta(IMRG) = O(n^{\beta-1}) \cdot \omega_\beta(MST)$ for any $\beta \geq 1$. This improves the previously known "lightest" structure RNG by $O(n)$ factor since in the worst case $\omega(RNG) = \Theta(n) \cdot \omega(MST)$ and $\omega_\beta(RNG) = \Theta(n^\beta) \cdot \omega_\beta(MST)$.

6 Summary and Open Questions

Wireless sensor networks have attracted considerable attention recently due to its potential wide applications in various areas and the moreover, the ubiquitous computing. Many excellent researches have been conducted to study both the electronic part and the networking part of the wireless sensor networks. For networking, there are also many interesting topics such as topology control, routing, energy conservation, QoS, mobility management, and so on. In this chapter, we present an overview of the recent progress of applying computational geometry techniques to solve some questions, such as topology construction and localized routing, in wireless sensor networks. Nevertheless, there are still many excellent results are not covered in this chapter due to space limit.

For example, Meguerdichian *et al.* [119] and Li *et al.* [101] solved the coverage problems in sensor networks, combining computational geometry and graph theoretic techniques. Their algorithms rely heavily on some geometrical structures such as Delaunay triangulation, Voronoi diagram, relative neighbor graph. We believe that with more research works arising in sensor networks field computational geometry techniques can help us solve more questions and plays an important role.

There are many interested open questions for topology control and localized routing in wireless sensor networks. Firstly, we would like to know whether the YaoYao structure $YY_k(V)$ is a length spanner for general graphs. Secondly, whether we can design a localized structure that achieves all desirable features such as bounded degree, planar, low weight and spanner. Thirdly, if consider interference and fault tolerance, how to design the network topology. Fourthly, when the overhead cost c of signal transmission is not negligible, whether the structures reviewed here are still power spanners. Fifthly, how to control the network topology when different nodes have different transmission ranges such that the topology has some nice properties? Sixthly, can we design a localized routing protocol that achieves constant ratio of the length of the found path to the minimum? The answer is probably negative, see [87].

References

[1] Ashwinder Ahluwalia, Eytan Modiano, and Li Shu. On the complexity and distributed construction of energy efficient broadcast trees in static ad hoc wireless networks. In *Proceedings of 36th Annual Conference on Information Sciences and Systems (CISS)*, 2002.

[2] Khaled Alzoubi, Peng-Jun Wan, and Ophir Frieder. Message-optimal connected-dominating-set construction for routing in mobile ad hoc networks. In *3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, 2002.

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

[4] Khaled M. Alzoubi, Peng-Jun Wan, and Ophir Frieder. New distributed algorithm for connected dominating set in wireless ad hoc networks. In *HICSS, Hawaii*, 2002.

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

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

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

[8] S. Arya and M. Smid. Efficient construction of a bounded degree spanner with low weight. In *Proc. 2nd Annu. European Sympos. Algorithms (ESA), volume 855 of Lecture Notes in Computer Science*, pages 48–59, 1994.

[9] Sunil Arya, Gautam Das, David M. Mount, Jeffrey S. Salowe, and Michiel Smid. Euclidean spanners. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, 1995.

[10] Mohsen Bahramgiri, Mohammad Taghi Hajiaghayi, and Vahab S. Mirrokni. Fault-tolerant and 3-dimensional distributed topology control algorithms in wireless multi-hop networks. In *Proceedings of the 11th Annual IEEE Internation Conference on Computer Communications and Networks (ICCCN)*, pages 392–397, 2002.

[11] S. Basagni. Distributed clustering for ad hoc networks. In *Proceedings of the IEEE International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN)*, pages 310–315, 1999.

[12] S. Basagni, I. Chlamtac, and A. Farago. A generalized clustering algorithm for peer-to-peer networks. In *Workshop on Algorithmic Aspects of Communication*, 1997.

[13] S. Basagni, I. Chlamtac, V.R. Syrotiuk, and B.A. Woodward. A distance routing effect algorithm for mobility (dream). In *Proceedings of ACM/IEEE MobiCom'98*, 1998.

[14] P. Berman, M. Furer, and A. Zelikovsky. Applications of matroid parity problem to approximating steiner trees. Technical Report 980021, Computer Science, UCLA, 1998.

[15] Christian Bettstetter. On the minimum node degree and connectivity of a wireless multihop network. In *3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, June 2002.

[16] P. Bose, A. Brodnik, S Carlsson, E. D. Demaine, R. Fleischer, A. Lopez-Ortiz, P. Morin, and J. I. Munro. Online routing in convex subdivisions. In *International Symposium on Algorithms and Computation*, pages 47–59, 2000.

[17] 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.

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

[19] P. Bose and P. Morin. Competitive online routing in geometric graphs. In *Proceedings of the VIII International Colloquium on Structural Information and Communication Complexity (SIROCCO 2001)*, pages 35–44, 2001.

[20] 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.

[21] Prosenjit Bose, Joachin Gudmundsson, and Pat Morin. Ordered theta graphs. In *Proceedings of the Canadian Conference on Computational Geometry (CCCG)*, 2002.

[22] Prosenjit Bose, Joachin Gudmundsson, and Michiel Smid. Constructing plane spanners of bounded degree and low weight. In *Proceedings of the European Symposium on Algorithms (ESA)*, 2002.

[23] J. Broch, D. Johnson, and D. Maltz. The dynamic source routing protocol for mobile ad hoc networks, 1998.

[24] Mario Cagalj, Jean-Pierre Hubaux, and Christian Enz. Minimum-energy broadcast in all-wireless networks: Np-completeness and distribution issues. In *Proceedings of the ACM MOBICOM 02*, 2002.

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

[26] S. Capkun, M. Hamdi, and J.P. Hubaux. Gps-free positioning in mobile ad-hoc networks. In *Proc. Hawaii Int. Conf. on System Sciences*, 2001.

[27] J. Cartigny, D. Simplot, and I. Stojmenvic. Localized minimum-energy broadcasting in ad-hoc networks. In *Proc. of IEEE INFOCOM 2003*, 2003.

[28] Barun Chandra, Gautam Das, Giri Narasimhan, and Jos Soares. New sparseness results on graph spanners. In *Proceedings of the eighth annual symposium on Computational geometry*, 1992.

[29] G. Chen, F. Garcia, J. Solano, and I. Stojmenovic. Connectivity based k-hop clustering in wireless networks. In *CD Proc. IEEE Hawaii Int. Conf. System Science*, 2002.

[30] Xiuzhen Cheng. A polynomial time approximation algorithm for connected dominating set problem in wireless ad hoc networks, 2002.

[31] Joseph Cheriyan, Santosh Vempala, and Adrian Vetta. Approximation algorithms for minimum-cost k-vertex connected subgraphs. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 306–312. ACM Press, 2002.

[32] C. Chiang and M. Gerla. demand multicast in mobile wireless networks, 1998.

[33] C.C. Chiang. Routing in clustered multihop, mobile wireless networks with fading channel. In *Proceedings of IEEE SICON'97*, pages 197–211, Apr. 1997.

[34] I. Chlamtac and A. Farago. A new approach to design and analysis of peer to peer mobile networks. *Wireless Networks*, 5:149–156, 1999.

[35] V. Chvátal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.

[36] A. Clementi, P. Crescenzi, P. Penna, G. Rossi, and P. Vocca. On the complexity of computing minimum energy consumption broadcast subgraphs. In *18th Annual Symposium on Theoretical Aspects of Computer Science, LNCS 2010*, pages 121–131, 2001.

[37] A. Clementi, P. Penna, and R. Silvestri. Hardness results for the power range assignment problem in packet radio networks. In *II International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (RANDOM/APPROX'99), LNCS(1671):197–208,*, 1999.

[38] A. Clementi, P. Penna, and R. Silvestri. The power range assignment problem in radio networks on the plane. In *XVII Symposium on Theoretical Aspects of Computer Science (STACS'00), LNCS(1770):651–660,*, 2000.

[39] A. Clementi, P. Penna, and R. Silvestri. On the power assignment problem in radio networks. *Electronic Colloquium on Computational Complexity*, 2001. To approach. Preliminary results in APPROX'99 and STACS'2000.

[40] Andrea E.F. Clementi, Paolo Penna, and Riccardo Silvestri. On the power assignment problem in radio networks, 2000.

[41] T. J. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* MIT Press and McGraw-Hill, 1990.

[42] Arindam K. Das, Robert J. Marks, Mohamed El-Sharkawi, Payman Arabshahi, and Andrew Gray. Minimum power broadcast trees for wireless networks: An ant colony system approach. In *Proc. IEEE Int. Symp. on Circuits and Systems*, 2002.

[43] Arindam K. Das, Robert J. Marks, Mohamed El-Sharkawi, Payman Arabshahi, and Andrew Gray. Minimum power broadcast trees for wireless networks: Integer programming formulations. In *Proc. of IEEE INFOCOM 2003*, 2003.

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

[45] Gautam Das and Giri Narasimhan. A fast algorithm for constructing sparse euclidean spanners. In *Proceedings of the tenth annual symposium on Computational geometry*, 1994.

[46] Gautam Das, Giri Narasimhan, and Jeffrey Salowe. A new way to weigh malnourished euclidean graphs. In *ACM Symposium of Discrete Algorithms*, pages 215–222, 1995.

[47] S. Datta, I. Stojmenovic, and J. Wu. Internal node and shortcut based routing with guaranteed delivery in wireless networks. *Cluster Computing*, 5(2):169–178, 2002.

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

[49] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer-Verlag, 1987.

[50] Deborah Estrin, Ramesh Govindan, John S. Heidemann, and Satish Kumar. Next century challenges: Scalable coordination in sensor networks. In *Mobile Computing and Networking (Mobicom)*, pages 263–270, 1999.

[51] M. Faloutsos and M. Molle. Creating optimal distributed algorithms for minimum spanning trees. Technical Report Technical Report CSRI-327 (also submitted in WDAG '95), 1995.

[52] H. N. Gabow, J. L. Bently, and R. E. Tarjan. Scaling and related techniques for geometry problems. In *ACM Symposium on Theory of Computing*, pages 135–143, 1984.

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

[54] R. Gallager, P. Humblet, and P. Spira. A distributed algorithm for minimumweight spanning trees. *ACM Transactions on Programming Languages and Systems*, 5(1):66–77, 1983.

[55] 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.

[56] J. A. Garay, S. Kutten, and D. Peleg. A sub-linear time distributed algorithms for minimum-weight spanning trees. In *Symp. on Theory of Computing*, pages 659–668, 1993.

[57] M. R. Garey and D. S. Johnson. *Computers and Intractability.* W.H. Freeman and Co., NY, 1979.

[58] Paul-Louis George and Houman Borouchaki. *Delaunay Triangulations and Meshing.* HERMES, 1998.

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

[60] Joachim Gudmundsson, Christos Levcopoulos, and Giri Narasimhan. Improved greedy algorithms for constructing sparse geometric spanners. In *Scandinavian Workshop on Algorithm Theory*, pages 314–327, 2000.

[61] Sudipto Guha and Samir Khuller. Approximation algorithms for connected dominating sets. In *European Symposium on Algorithms*, pages 179–193, 1996.

[62] P. Gupta and P. Kumar. Capacity of wireless networks. Technical report, University of Illinois, Urbana-Champaign, 1999.

[63] P. Gupta and P. R. Kumar. Critical power for asymptotic connectivity in wireless networks. *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming, W. M. McEneaney, G. Yin, and Q. Zhang (Eds.)*, 1998.

[64] Z. Haas and B. Liang. Ad-hoc mobility management with uniform quorum systems. *IEEE/ACM Transactions on Networking*, 7(2):228–240, 1999.

[65] MohammadTaghi Hajiaghayi, Nicole Immorlica, and Vahab S. Mirrokni. Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks. In *Proceedings of the ACM MOBICOM 03*, 2003.

[66] Christopher Ho, Katia Obraczka, Gene Tsudik, and Kumar Viswanath. Flooding for reliable multicast in multi-hop ad hoc networks. In *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 64–71, Seattle, WA, 1999.

[67] Dorit S. Hochbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and vlsi. *Journal of ACM*, 32:130–136, 1985.

[68] Harry B. Hunt III, Madhav V. Marathe, Venkatesh Radhakrishnan, S. S. Ravi, Daniel J. Rosenkrantz, and RichardE. Stearns. NC-approximation schemes for NP- and PSPACE -hard problems for geometric graphs. *Journal of Algorithms*, 26(2):238–274, 1998.

[69] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Mobile Computing and Networking (Mobicom)*, pages 56–67, 2000.

[70] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen. Scalable routing strategies for ad hoc wireless networks. *JSAC99, IEEE Journal on Selected Areas in Communications*, 17(8):1369–1379, August 1999.

[71] Jorjeta G. Jetcheva, Yih-Chun Hu, David A. Maltz, and David B. Johnson. A simple protocol for multicast and broadcast in mobile ad hoc networks, 2001.

[72] Lujun Jia, Rajmohan Rajaraman, and Christain Scheideler. On local algorithms for topology control and routing in ad hoc networks. In *Proceedings of the 15th Annual ACM Symposium on Parallel Algorithms and Architectures*, 2003.

[73] Lujun Jia, Rajmohan Rajaraman, and Torsten Suel. An efficient distributed algorithm for constructing small dominating sets. In *ACM PODC*, 2000.

[74] Mario Joa-Ng and I-Tai Lu. A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks. *IEEE Journal on Selected Areas in Communication*, 17(8):1415–1425, August 1999.

[75] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: Mobile networking for "smart dust". In *International Conference on Mobile Computing and Networking (MOBICOM)*, pages 271–278, 1999.

[76] Menelaos I. Karavelas and Leonidas J. Guibas. Static and kinetic geometric spanners with applications. In *Proceeding of the Twelfth Annual Symposium on Discrete algorithms*, pages 168–176, 2001.

[77] Brad Karp. *Geographic Routing for Wireless Networks*. PhD thesis, Harvard University, 2000.

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

[79] J. Katajainen. The region approach for computing relative neighborhood graphs in the lp metric. *Computing*, 40:147–161, 1988.

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

[81] J.M. Keil. Approximating the complete euclidean graph. In *SWAT 88: 1st Scandinavian Workshop on Algorithm Theory*, 1988.

[82] L. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc. Power consumption in packet radio networks. In *Symposium on Theoretical Aspects of Computer Science (STACS) '97.*, 1997.

[83] L. M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc. Power consumption in packet radio networks. *Theoretical Computer Science*, 243:289–305, 2000.

[84] Guy Kortsarz and Zeev Nutov. Approximating node connectivity problems via set covers. In *Proc. of the third International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX 2000)*, 2000.

[85] E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *Proc. 11 th Canadian Conference on Computational Geometry*, pages 51–54, 1999.

[86] Fabian Kuhn, Roger Wattenhofer, Yan Zhang, and Aaron Zollinger. Geometric Ad-Hoc Routing: Of Theory and Practice". In *Proc. 22^{nd} ACM Int. Symposium on the Principles of Distributed Computing (PODC)*, 2003.

[87] Fabian Kuhn, Roger Wattenhofer, and Aaron Zollinger. Asymptotically optimal geometric mobile ad-hoc routing. In *Proc. of the 6th international workshop on Discrete algorithms and methods for mobile computing and communications (Dial-M)*, pages 24–33. ACM Press, 2002.

[88] Fabian Kuhn, Roger Wattenhofer, and Aaron Zollinger. Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing. In *Proc. 4^{th} ACM Int. Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc)*, 2003.

[89] Christos Levcopoulos, Giri Narasimhan, and Michiel Smid. Efficient algorithms for constructing fault-tolerant geometric spanners. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, 1998.

[90] Christos Levcopoulos, Giri Narasimhan, and Michiel Smid. Improved algorithms for constructing fault tolerant geometric spanners. *Algorithmica*, 2000.

[91] Fulu Li and Ioanis Nikolaidis. On minimum-energy broadcasting in all-wireless networks. In *Proc. IEEE 26th Annual IEEE Conference on Local Computer Networks (LCN'01)*, 2001.

[92] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris. A scalable location service for geographic ad-hoc routing. In *Proceedings of the 6th ACM International Conference on Mobile Computing and Networking (MobiCom '00)*, pages 120–130, 2000.

[93] Li Li, Joseph Y. Halpern, Paramvir Bahl, Yi-Min Wang, and Roger 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.

[94] Ning Li and Jennifer C. Hou. Blmst: a scalable, power efficient broadcast algorithm for wireless sensor networks, 2003. Submitted for publication, a version as UIUC Computer Science Department Technical Report.

[95] Ning Li, Jennifer C. Hou, and Lui Sha. Design and analysis of a mst-based topology control algorithm. In *Proc. of IEEE INFOCOM 2003*, 2003.

[96] Xiang-Yang Li. Topology control in wireless ad hoc networks. 2002.

[97] Xiang-Yang Li. Approximate mst for udg locally. In *The 9th Annual International Computing and Combinatorics Conference COCOON 2003*, 2003.

[98] Xiang-Yang Li, G. Calinescu, and Peng-Jun Wan. Distributed construction of planar spanner and routing for ad hoc wireless networks. In *21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 3, 2002.

[99] Xiang-Yang Li and Ivan Stojmenovic. Partial delaunay triangulation and degree limited localized bluetooth scatternet formation. In *AdHocNow*, 2002.

[100] Xiang-Yang Li, Ivan Stojmenovic, and Yu Wang. Partial delaunay triangulation and degree limited localized bluetooth multihop scatternet formation, 2002. Submitted for publication. The short version appeared at AdHocNow, 2002.

[101] Xiang-Yang Li, Peng-Jun Wan, and Ophir Frieder. Coverage in wireless ad-hoc sensor networks. In *IEEE International Conference on Communications (ICC)*, 2002.

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

[103] Xiang-Yang Li, Peng-Jun Wan, Yu Wang, and Ophir Frieder. Sparse power efficient topology for wireless networks. In *IEEE Hawaii Int. Conf. on System Sciences (HICSS)*, 2002.

[104] Xiang-Yang Li, Peng-Jun Wan, Yu Wang, and Ophir Frieder. Sparse power efficient topology for wireless networks. *Journal of Parallel and Distributed Computing*, 2002. Accepted for publication. Short version in IEEE ICCCN 2001.

[105] Xiang-Yang Li and Yu Wang. Efficient construction of bounded degree planar spanner. In *The 9th Annual International Computing and Combinatorics Conference (COCOON2003)*, 2003.

[106] Xiang-Yang Li and Yu Wang. Localized routing for wireless ad hoc networks. In *IEEE International Conference on Communications (ICC 2003)*, 2003.

[107] Xiang-Yang Li, Yu Wang, Peng-Jun Wan, and Ophir Frieder. Localized low weight graph and its applications in wireless ad hoc networks, 2003. Submitted for publication.

[108] Xiang-Yang Li, Yu Wang, Peng-Jun Wang, and Chih-Wei Yi. Fault tolerant deployement and topology constrol for wireless ad hoc networks. In *The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc2003)*, 2003.

[109] Weifa Liang. Constructing minimum-energy broadcast trees in wireless ad hoc networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 112–122, 2002.

[110] Chunhung Richard Lin and Mario Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal of Selected Areas in Communications*, 15(7):1265–1275, 1997.

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

[112] M. Liu, R. Talpade, and A. McAuley. Amroute: Adhoc multicast routing protocol, 1999.

[113] L. Lloyd, Rui Liu, Madhav V. Marathe, Ram Ramanathan, , and S. S. Ravi. Algorithmic aspects of topology control problems for ad hoc networks. In *IEEE MOBIHOC*, 2002.

[114] Tamas Lukovszki. *New Results on Geometric Spanners and Their Applications.* PhD thesis, University of Paderborn, 1999.

[115] Madhav V. Marathe, Heinz Breu, Harry B. Hunt III, S. S. Ravi, and Daniel J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25:59–68, 1995.

[116] R. J. Marks, A.K. Das, M. El-Sharkawi, P. Arabshahi, and A. Gray. Minimum power broadcast trees for wireless networks: optimizing using the viability lemma. In *Proc. IEEE Int. Symp. on Circuits and Systems*, pages 245–248, 2002.

[117] Martin Mauve, Jorg Widmer, and Hannes Harenstein. A survey on position-based routing in mobile ad hoc networks. *IEEE Network Magazine*, 15(6):30–39, 2001.

[118] Seapahn Megerian and Miodrag Potkonjak. Wireless sensor networks. Book Chapter in Wiley Encyclopedia of Telecommunications, Editor: John G. Proakis, 2002.

[119] Seapahn Meguerdichian, Farinaz Koushanfar, Miodrag Potkonjak, and Mani Srivastava. Coverage problems in wireless ad-hoc sensor network. In *IEEE INFOCOM'01*, pages 1380–1387, 2001.

[120] J. Monks, V. Bharghavan, and W.-M Hwu. Transmission power control for multiple access wireless packet networks. In *IEEE Conference on Local Computer Networks (LCN)*, 2000.

[121] P. Morin. *Online routing in Geometric Graphs.* PhD thesis, Carleton University School of Computer Science, 2001.

[122] Vincent D. Park and M. Scott Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of IEEE INFOCOM '97, Kobe, Japan*, April 1997.

[123] G. Pei, M. Gerla, and T.-W. Chen. Fisheye state routing: A routing scheme for ad hoc wireless networks. In *Proceedings of ICC 2000, New Orleans, LA*, 2000.

[124] G. Pei, M. Gerla, and T.-W. Chen. Fisheye state routing in mobile ad hoc networks. In *Proceedings of Workshop on Wireless Networks and Mobile Computing, Taipei, Taiwan, Apr*, 2000.

[125] Mathew Penrose. The longest edge of the random minimal spanning tree. *Annals of Applied Probability*, 7:340–361, 1997.

[126] Mathew Penrose. On k-connectivity for a geometric random graph. *Random Structures and Algorithms*, 15:145–164, 1999.

[127] C.E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *Computer Communications Review*, pages 234–244, October 1994.

[128] C.E. Perkins and E.M.Royer. Ad-hoc on demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA*, pages 90–100, February 1999.

[129] Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir Das. Ad hoc on demand distance vector (aodv) routing, 2002.

[130] G.J. Pottie and W.J. Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43(5):551–558, 2000.

[131] Franco P. Preparata and Michael Ian Shamos. *Computational Geometry: an Introduction.* Springer-Verlag, 1985.

[132] Amir Qayyum, Laurent Viennot, and Anis Laouiti. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. Technical Report Research Report RR-3898, INRIA, February 2000. Conference version in Proc. of IEEE HICSS'35, 2001.

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

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

[135] G. Robins and A. Zelikovsky. Improved steiner tree approximation in graphs. In *Proceedings of ACM/SIAM Symposium on Discrete Algorithms*, pages 770–779, 2000.

[136] E. Royer and C. Toh. A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Communications*, Apr. 1999.

[137] M. Sanchez, P. Manzoni, and Z. Haas. Determination of critical transmission range in ad-hoc networks. In *Multiaccess, Mobility and Teletraffic for Wireless Communications (MMT'99)*, 1999.

[138] C. Santivanez, B. McDonald, I. Stavrakakis, and R. Ramanathan. the scalability of ad hoc routing protocols. In *Proc. of IEEE INFOCOM 2002*, 2002.

[139] Mahtab Seddigh, J. Solano Gonzalez, and I. Stojmenovic. Rng and internal node based broadcasting algorithms for wireless one-to-one networks. *ACM Mobile Computing and Communications Review*, 5(2):37–44, 2002.

[140] P. Sinha, R. Sivakumar, and V. Bharghavan. Cedar: Core extraction distributed ad hoc routing, 1999.

[141] R. Sivakumar, B. Das, and V. Bharghavan. An improved spine-based infrastructure for routing in ad hoc networks. In *IEEE Symposium on Computers and Communications*, Athens, Greece, June 1998.

[142] I. Stojmenovic. A routing strategy and quorum based location update scheme for ad hoc wireless networks. Technical Report TR-99-09, Computer Science, SITE, University of Ottawa, 1999.

[143] I. Stojmenovic and S. Datta. Power and cost aware localized routing with guaranteed delivery in wireless networks. In *Proc. Seventh IEEE Symposium on Computers and Communications ISCC*, 2002.

[144] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 13(1):14–25, 2002.

[145] Ivan Stojmenovic and Xu Lin. Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(10), 2001.

[146] Ivan Stojmenovic, Mahtab Seddigh, and Jovisa Zunic. Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 13(1):14–25, 2002.

[147] C-K. Toh. A novel distributed routing protocol to support ad-hoc mobile networks. In *Proc 1996 IEEE 15th Annual Int'l. Phoenix Conf. Comp. and Commun.*, pages 480–86, Mar 1996.

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

[149] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. *Wireless Networks*, 8:153–167, 2002. Short version in MOBICOM 99.

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

[151] Peng-Jun Wan, G. Calinescu, Xiang-Yang Li, and Ophir Frieder. Minimum-energy broadcast routing in static ad hoc wireless networks. *ACM Wireless Networks*, 2002. Preliminary version appeared in IEEE INFOCOM 2000.

[152] WeiZhao Wang, Xiang-Yang Li, Kousha Moaveninejad, Yu Wang, and Wen-Zhan Song. The spanning ratios of *beta*-skeleton. In *Canadian Conference on Computational Geometry (CCCG)*, 2003. Accepted for publication.

[153] Yu Wang and Xiang-Yang Li. Distributed spanner with bounded degree for wireless ad hoc networks. In *International Parallel and Distributed Processing Symposium: Parallel and Distributed Computing Issues in Wireless networks and Mobile Computing*, April 2002.

[154] Yu Wang and Xiang-Yang Li. Geometric spanners for wireless ad hoc networks. In *Proc. of 22nd IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2002.

[155] Yu Wang and Xiang-Yang Li. Localized construction of bounded degree planar spanner for wireless networks. In *ACM DIALM-POMC Joint Workshop on Foundations of Mobile Computing*, 2003.

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

[157] J. Wieselthier, G. Nguyen, and A. Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *Proc. IEEE INFOCOM 2000*, pages 586–594, 2000.

[158] J.E. Wieselthier, G.D. Nguyen, and A. Ephremides. The energy efficiency of distributed algorithms for broadcasting in ad hoc networks. In *Proceedings of IEEE 5th International Symposium on Wireless Personal Multimedia Communication (WPMC)*, pages 499–503, 2002.

[159] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 194–205, 2002.

[160] C. W. Wu, Y. C. Tay, and C.-K Toh. Ad hoc multicast routing protocol utilizing increasing id-numbers (amris) functional specification, 1998.

[161] Jie Wu, F. Dai, M. Gao, and I. Stojmenovic. On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks. *IEEE/KICS Journal of Communication and Networks*, 4(1):59–70, 2002.

[162] Jie Wu and Hailan Li. Domination and its applications in ad hoc wireless networks with unidirectional links. In *Proc. of the International Conference on Parallel Processing 2000*, pages 189–197, 2000.

[163] Jie Wu and Hailan Li. A dominating-set-based routing scheme in ad hoc wireless networks. *the special issue on Wireless Networks in the Telecommunication Systems Journal*, 3:63–84, 2001.

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

[165] Yan Yu, Ramesh Govindan, and Deborah Estrin. Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks. Technical report, UCLA Computer Science Department Technical Report UCLA/CSD-TR-01-0023, 2001.

[166] Haas Z.J. and Pearlman M.R. The zone routing protocol(zrp) for ad hoc networks. In *Internet draft - Mobile Ad hoc NETworking (MANET), Working Group of the Internet Engineering Task Force (IETF)*, Novermber 1997.