Ad Hoc Wireless Networking
X. Cheng, X. Huang and D.-Z. Du (Eds.) pp. – – –

Applications of Computational Geometry in Wireless Networks

Xiang-Yang Li
Department of Computer Science
Illinois Institute of Technology, Chicago, IL 60616
E-mail: xli@cs.iit.edu

Contents

1 Introduction

Ad Hoc Wireless Networks Due to its potential applications in various situations such as battlefield, emergency relief, and so on, wireless networking has received significant attention over the last few years. There are no wired infrastructures or cellular networks in *ad hoc* wireless network. Each mobile node has a transmission range. 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 wireless network also acts as a router, forwarding data packets for other nodes. In this survey, we consider that each wireless 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. 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 [1].

Wireless ad hoc networks can be subdivided into two classes: *static* and *mobile*. In static networks, the position of a wireless node does not change or changes very slowly once the node was deployed. Typical example of such static networks includes sensor networks. In mobile networks, wireless nodes move arbitrarily. Since mobile wireless networks change their topology frequently and often without any regular pattern, topology maintenance and routing in such networks are challenging tasks. For the sake of the simplicity, we assume that the nodes are quasi-static during the short period of topology reconstruction or route finding.

We consider a wireless ad hoc network consisting of a set V of n wireless nodes distributed in a two-dimensional plane. By a proper scaling, we assume that all nodes have the maximum transmission range equal to one unit. These wireless nodes define a *unit disk graph* $\mathrm{UDG}(V)$ in which there is an edge between two nodes if and only if their Euclidean distance is at most one.

Computational Geometry Computational geometry emerged from the field of algorithms design and analysis in the late 70s. It studies various problems [2, 3, 4] 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 survey, we concentrate on how to apply some structural properties of a point set for wireless networks as we treat wireless devices as two-dimensional points.

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 be suited for supporting a good routing scheme. Wireless ad hoc network needs some special treatment as it intrinsically has its own special characteristics and some unavoidable limitations compared with traditional wired networks. Wireless 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 ad hoc networks, which is suitable for designing an efficient routing scheme to save energy and storage memory consumption, than the traditional wired networks.

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 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.

Network Structures in Wireless Networks 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 wireless 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 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 survey, we review and study how to construct a spanner (a sparse network topology) efficiently for a set of static wireless nodes.

The other imperative requirement for network topology control in wireless ad hoc networks is the fault tolerance. To guarantee a good fault tolerance, the underlying network structure must be at least bi-connected, i.e., there are at least two disjoint paths for any pair of wireless nodes. Here, without doubt, we assume that the original unit disk graph is bi-connected.

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 [5, 6, 7, 8]. These methods often construct a virtual backbone by using the connected dominating set [9, 10, 11], which is often constructed from dominating set or maximal independent set.

Routing Many routing algorithms were proposed recently for wireless ad hoc networks. The routing protocols proposed may be categorized as table-driven protocols or demand-driven protocols. A good survey may be found in [12].

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) [13] and Zone-Routing Protocol (ZRP) [14, 15] are two of the table driven protocols proposed recently. The mobility nature of the wireless 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) [16], the Dynamic Source Routing (DSR) [17], and the Temporarily Ordered Routing Algorithm (TORA) [18]. In addition, the Associativity Based Routing (ABR) [19] and Signal Stability Routing (SSR) use various criteria for selecting routes.

Introducing a hierarchical structure into routing has also been used in many protocols such as the Clusterhead Gateway Switch Routing (CGSR) [20], the Fisheye Routing [21, 22], and the Hierarchical State Routing [23]. Dominating set based methods were also adopted by several researchers [6, 7, 8]. To facilitate this, several methods [24, 9, 10, 25] 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 memory scarcity of wireless network nodes. The geometric nature of the multi-hop ad-hoc wireless 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 ad-hoc 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. [26] (see also [27]) 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.* [28] conducted an excellent survey of position-based localized routing protocols.

Organization The rest of the survey 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 ad hoc networks. Specifically, we specify how the wireless network is modeled in this survey, 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 ad hoc networks, especially the structures with bounded stretch factor, or with bounded node degree, or planar structures. 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. In Section 4, 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 5. Many routing algorithms were proposed in the literature. We concentrate on the localized routing protocols as they utilize the geometry nature of the wireless ad hoc networks. Location service protocols are also discussed. Section 6 reviews the broadcasting protocols that apply the geometry nature to guarantee the performance. In Section 7, we review the current status of applying stochastic geometry to study the connectivity, capacity, etc, in wireless networks. We conclude the survey in Section 8 by pointing out some possible future research questions.

2 Preliminaries

2.1 Power-Attenuation Model

Energy conservation is a critical issue in *ad hoc* wireless network for the node and network life, as the nodes are powered by batteries only. Each mobile 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 required to support the transmission between two nodes is dependent on their distance. Finally, when a node receives the signal, it needs consume some power to receive, store and then process that signal. 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.

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 wireless transmission environment. This power consumption is typically called *path loss*. In this survey, we assume that the path loss is the major part of power consumption to transmit signals.

Notice that, practically, there is some other overhead cost for each device to receive and then process the 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 $c = 0$.

2.2 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 networking applications. Let $G = (V, E)$ be a geometric graph defined on V.

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

The *relative neighborhood graph*, denoted by $\mathrm{RNG}(G)$, is a geometric concept proposed by Toussaint [29]. 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 $\|wv\| < \|uv\|$. Thus, an edge uv is included if the intersection of two circles centered at u and v and with radius $\|uv\|$ do not contain any vertex w from the set V such that edges uw and wv are in E. 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* [30] $\mathrm{GG}(G)$ 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 satisfying $\|uw\| < \|uv\|$ and $\|wv\| < \|uv\|$. Same to the definition of $\mathrm{RNG}(G)$, if G is a directed graph, then edges uw and wv also are directed in the above definition of $\mathrm{GG}(G)$, i.e., we use $\overrightarrow{uw}$ and $\overrightarrow{wv}$ instead. $\mathrm{GG}(G)$ is a planar graph (that is, no two edges cross each other) if G is the complete graph. It is easy to show that $\mathrm{RNG}(G)$ is a subgraph of the Gabriel graph $\mathrm{GG}(G)$. For an undirected and connected graph G, both $\mathrm{GG}(G)$ and $\mathrm{RNG}(G)$ are connected and contain the minimum spanning tree of G.

The *Yao graph* 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 1 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 [31], 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. Here the axis of a cone is the angular bisector of the cone. For more detail, please refer to [31].

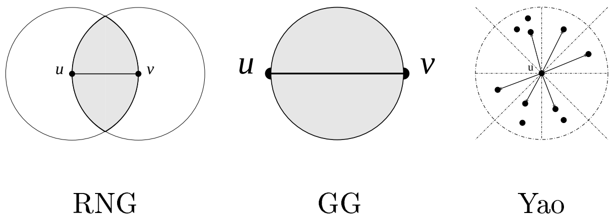


Figure 1: The definitions of RNG, GG, and Yao on point set. Left: The lune using uv is empty for RNG. Middle: The diametric circle using uv is empty for GG. Right: The shortest edge in each cone is added as a neighbor of u for Yao.

Notice all these definitions are exactly the conventional definitions [32, 33, 34, 35] when graph G is the completed Euclidean graph $K(V)$. We will use RNG(V), GG(V), and Yao(V) to denote the corresponding resulting graph if G is the complete graph $K(V)$. Gabriel graph was used as a planar subgraph in the Face routing protocol [26, 36, 37] and the GPSR routing protocol [27] 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 [38].

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.

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.

Besides these geometric structures, some graph notations will also be used in this survey. 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 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.

2.3 Spanners

Spanners have been studied intensively in recent years [39, 40, 41, 42, 43, 44, 45, 46, 34]. 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. Geometric spanners were first introduced in computational geometry community by Chew [47]. Now they have numerous applications in computer science, such as VLSI, robotics motion planning, distributed systems, and communication networks. In this survey, we focus on their applications in wireless networks.

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. The path in G connecting u, v and consuming the least energy $p_G(u,v)$ is called the *least-energy path* in G for u and v. When G is the unit disk graph UDG(V), we will omit the subscript G in $p_G(u,v)$.

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 [48] that, for a constant δ, $\rho_H(G) \le \delta$ iff for any link $v_i v_j$ in graph G but not in H, $p_H(v_i, v_j) \le \delta \|v_i v_j\|^{\beta}$. 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 $\rho_{H_1}(G) \ge \rho_{H_2}(G)$.

2.4 Localized Algorithm

Due to the limited resources of the wireless nodes, 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 EMST(V) and the Delaunay triangulation Del(V) can not be constructed by any localized algorithm. In this survey, we are interested in localized algorithms that construct sparse and power efficient network topologies.

3 Topology Control

In this section, we study the power stretch factor of several new sparse spanners for unit disk graph. 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, which is defined as the maximum ratio of the minimum power needed to support the connection of two nodes in this spanner to the least necessary in the unit disk graph.

3.1 RNG, GG, and Yao

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.* [48] showed that it is actually $n-1$.

Theorem 3.1 *[48]* $\rho_{RNG}(n) = n-1$.

First $\rho_{RNG}(n)$ is at most $n-1$. Consider the path between u and v in EMST(V). This path contains at most $n-1$ edges and each edge has length at most $\|uv\|$. Thus, its total power consumption is at most $(n-1)\|uv\|^\beta$. Notice $EMST(V) \subset RNG(V)$ if UDG(V) is connected. Thus,

$$\rho_{RNG}(n) \leq n-1.$$

Then $\rho_{RNG}(n) \geq n-1-\varepsilon$ for any small positive ε by constructing an example illustrated in Figure 2.

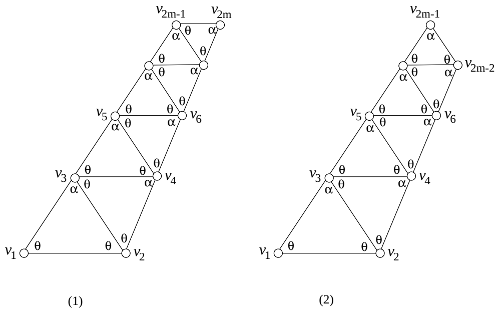


Figure 2: The Euclidean minimum spanning tree has large stretch factor.

They considered two cases. First consider even n, say $n = 2m$. The construction of the point set V is shown in Figure 2 (1), which was used in [42]. Let $\alpha = \frac{\pi}{3} + 2\delta, \theta = \frac{\pi}{3} - \delta$, where δ is a sufficiently small positive number which will be fixed later. The m points with odd subscripts v_1, v_3, v_5, $\cdots$, v_{2m-1} are collinear, so are the m points with even subscripts v_2, v_4, v_6, $\cdots$, v_{2m}. As proved in [42], $RNG\,(V)$ is a path v_1, v_3, v_5, $\cdots$, v_{2m-1}, v_{2m}, $\cdots$, v_6, v_4, v_2. As $\delta \longrightarrow 0$, the length of each edge in $RNG\,(V)$ tends to $\|v_1v_2\|$ from below, which implies $\frac{p_{RNG}(u,v)}{p(u,v)} \longrightarrow n-1$. So we can find a sufficiently small $\delta > 0$ such that $\frac{p_{RNG}(u,v)}{p(u,v)} > n - 1 - \epsilon$, which implies $\rho_{RNG}(n) > n - 1 - \epsilon$.

When n is odd, the construction is shown in Figure 2 (2) and the existence can be proved by a similar argument.

The Gabriel graph has length stretch factor between $\frac{\sqrt{n}}{2}$ and $\frac{4\pi\sqrt{2n-4}}{3}$ [42]. Then its power stretch factor is at most $\left(\frac{4\pi\sqrt{2n-4}}{3}\right)^2$.

Theorem 3.2 *[48] The power stretch factor of any Gabriel graph is one.*

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.* [48] proved a stronger result.

Theorem 3.3 *[48] The power stretch factor of the Yao graph $YG_k(V)$ is at most $\frac{1}{1-(2\sin\frac{\pi}{k})^\beta}$.*

See [48] for a detailed proof of this theorem. Li *et al.* [49] 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.

We end this subsection by commenting a result by Wattenhofer *et al.* [50]. Their two-phased approach 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 bugs in their proof of the constant power stretch factor and their result is erroneous, which was discussed in detail in [48].

Li *et al.* [51] 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 [51] 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 [51].

3.2 Bounded Degree Spanners

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 a very large in-degree. The nodes configuration given in Figure 3 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.

3.2.1 Sink Structure

Arya *et al.* [40] gave an ingenious technique to generate a bounded degree graph with constant length stretch factor. In [48], 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

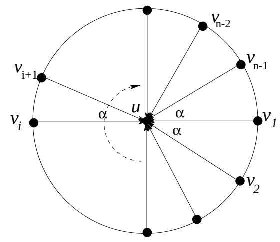


Figure 3: Node u has degree (or in-degree) $n-1$.

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.

Algorithm: 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 Tree(u,$I(u)$) to build tree $T(u)$.

Algorithm: Constructing-$T(u)$ Tree(u,$I(u)$)

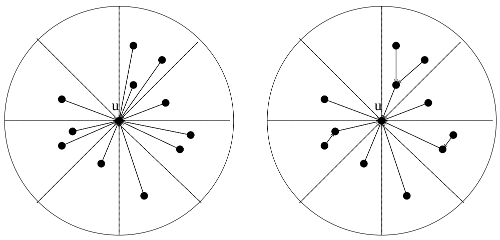
1. To partition the unit disk centered at u, choose k equal-sized cones centered at u: $C_1(u)$, $C_2(u)$, $\cdots$, $C_k(u)$.

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 $\overrightarrow{y_iu}$ 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 4 (a) illustrates a directed star centered at u and Figure 4 (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 node 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.

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 $\overrightarrow{YG}_k$,



(a) (b)

Figure 4: (a) Star formed by links toward to u. (b) Directed tree $T(u)$ sinked at u.

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 3.4 *[48] 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.

3.2.2 YaoYao Structure

In this section, we review another algorithm proposed by Li *et al.* [49] 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 3.5 *[49] Graph $\overrightarrow{YY}_k(V)$ is strongly connected if UDG(V) is connected and $k > 6$.*

It was proved in [52] 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 [53], they called the civilized unit disk graph as the λ-precision unit disk graph. Notice the wireless devices in wireless networks can not be too close or overlapped. Thus, it is reasonable to model the wireless ad hoc networks as a civilized unit disk graph.

Theorem 3.6 *[49] The power stretch factor of the directed topology $\overrightarrow{YY}_k(V)$ is bounded by a constant ρ in civilized graph.*

The experimental results by Li *et al.* [49] showed that this sparse topology has a small power stretch factor in practice. They [49] conjectured that $\overrightarrow{YY}_k(V)$ also has a constant bounded power stretch factor theoretically in any unit disk graph. The proof of this conjecture or the construction of a counter-example remain a future work.

3.2.3 Symmetric Yao Graph

In [49], 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 3.7 *[48] The graph $YS_k(V)$ is strongly connected if UDG(V) is connected and $k \geq 6$.*

This was proved by showing that RNG is a subgraph of $YS_k(V)$ if $k \geq 6$. Notice that, Theorem 3.7 immediately implies the connectivity of the Yao graph, sink structure, and the YaoYao graph as RNG is also the subgraph of all these structures.

The experiment by Li *et al.* also showed that it has a small power stretch factor in practice. However, it was shown in [54] recently that $YS_k(V)$ is not a spanner theoretically. The basic idea of the counter example is similar to the counter example for RNG proposed by Bose *et al.* [42]. For the completeness of the presentation, we still review the counter example here.

Let nodes v_1 and v_0 have distance half unit from each other. Assume the ith cone of v_1 contains v_0, and the i'th cone of v_0 contains v_1. Then draw two lines $l_1 = v_1v_3$ and $l_2 = v_0v_2$ such that both the angles $\angle v_3v_1v_0$ and $\angle v_2v_0v_1$ are $\frac{\pi}{2} - \alpha$, where α is a very small positive number. Let's first consider even n, say $n = 2m$. Figure 5 illustrates the construction of the point set V. The node v_{2j} is placed on l_2 in the ith cone of v_{2j-1} and it is very close to the upper boundary of the ith cone of v_{2j-1}. The node v_{2j+1} is placed on l_1 in the i'th cone of v_{2j} close to the upper boundary of that cone. Using this method, place all nodes from v_2 to v_{2m} in order. Then it is easy to show that the $YS_k(V)$ does not contain any edge $v_{2j}v_{2j+1}$ and $v_{2j+1}v_{2j+2}$ for $0 \leq j \leq m-1$. The nearest neighbor of v_{2j} is v_{2j+1}, but for v_{2j+1}, the nearest neighbor is v_{2j+2}. So although in $YS_k(V)$ there is a path from v_1 to v_2, its length is $\|v_1v_{2m-1}\| + \|v_{2m-1}v_{2m}\| + \|v_{2m}v_2\|$. So when α is appropriately small, the length stretch factor of $YS_k(V)$ cannot be bounded by a constant. Similarly, its power stretch factor cannot be bounded also. When n is odd, the construction is similar.

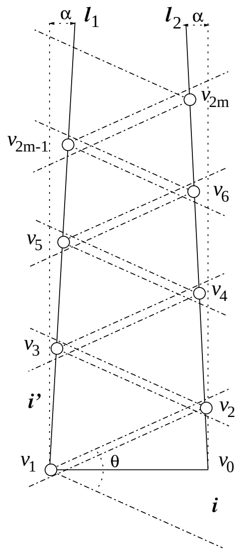


Figure 5: An example that $YS_k(V)$ has a large stretch factor.

3.3 Planar Spanner

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 $K(V)$. This was first proved by Dobkin, Friedman and Supowit with constant $t = \frac{1+\sqrt{5}}{2}\pi \approx 5.08$. Then Kevin and Gutwin improved the upper bound on t to be $\frac{2\pi}{3\cos\frac{\pi}{6}} =$

$\frac{4\sqrt{3}}{9}\pi \approx 2.42$. 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.* [35] considered the *unit Delaunay triangulation* UDel(V) for planar spanner of UDG, which is a subset of the Delaunay triangulation. It was proved in [35] that UDel(V) is a t-spanner of the unit disk graph UDG(V).

Theorem 3.8 *[35] For any two vertices u and v of V,*

$$||\Pi_{UDel(V)}(u,v)|| \leq \frac{1+\sqrt{5}}{2}\pi \cdot ||\pi_{UDG(V)}(u,v)||$$

Notice that, Kevin and Gutwin [55] 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 [55] 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 $UDG(V)$.

3.3.1 Localized Delaunay triangulation

Li *et al.* [35] 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.

Unit Gabriel graph It consists of all edges uv such that $\|uv\| \leq 1$ and the open disk using uv as diameter does not contain any vertex from V. Such edge uv is called the *Gabriel edge*. We denote the unit Gabriel graph by $GG(V)$ hereafter.

k-localized Delaunay triangle 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.

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

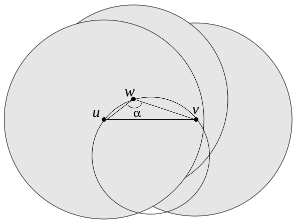


Figure 6: LDel: The circumcircle $disk(u, v, w)$ is not necessarily covered by unit disks centered at u and v. But it is empty of other vertices from $N_1(u) \cup N_1(v) \cup N_1(w)$.

When it is clear from the context, we will omit the integer k in our notation of $LDel^{(k)}(V)$. They originally conjectured that $LDel^{(1)}(V)$ is a planar graph and thus a planar t-spanner of $UDG(V)$ can be easily constructed by using localized approach. Unfortunately, as shown in [35], the graph $LDel^{(1)}(V)$ may contain some edges intersecting. On the other hand, $LDel^{(2)}(V)$ is a planar graph.

Theorem 3.9 *[35] $LDel^{(k)}(V)$ is a planar graph for any $k \geq 2$.*

Theorem 3.10 *Assume two triangles $\triangle uvw$ and $\triangle xyz$ of $LDel^{(k)}(V)$, $k \geq 1$, intersect, then either $disk(u, v, w)$ contains at least one of the nodes of $\{x, y, z\}$ or $disk(x, y, z)$ contains at least one of the nodes of $\{u, v, w\}$.*

Notice that, although $LDel^{(1)}(V)$ is not a planar graph, the following theorem proved in [35] guarantees that it is sparse.

Theorem 3.11 *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 this approach is $O(m \log n)$ bits, where m is the number of edges in $UDG(V)$ and could be as large as $O(n^2)$. This is more complicated than some other non-planar t-spanners, such as the Yao structure [34] and the θ-graph [55] (although the lattes are not planar). 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 1 *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 angle 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 3.11. Putting together the arguments above, we have:

Theorem 3.12 *[35] Algorithm 1 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 2 *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 3.11 is $O(n)$. Putting together all the arguments above and Theorem 3.8,

Theorem 3.13 *$PLDel(V)$ 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.*

3.3.2 Partial Delaunay triangulation

Stojmenovic and Li [56] 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 $UDel(V)$. Thus, the spanning ratio of the partial Delaunay triangulation could be very large.

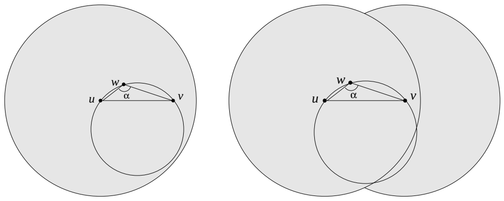


Figure 7: Left: Only one hop information is known to u. Then it requires $disk(u, v, w)$ to be covered by the transmission range of u (denoted by the shaded region) and is empty of neighbors of u. Right: Node u knows $N_1(u)$ and node v knows $N_1(v)$. The circumcircle $disk(u, v, w)$ is covered by the union of the transmission ranges of u and v and is empty of other vertices.

3.3.3 Restricted Delaunay Graph

Gao *et al.* [57] also proposed another structure, called *restricted Delaunay graph* RDG and showed that it has good spanning ratio properties and is easy to maintain 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 maintain the 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.

3.4 Bounded Degree Planar Spanner

The structures discussed so far either have bounded degree, or is planar, or is spanner, but none of the structures has all these three properties together.

We then review one recent result [60] that can construct a bounded degree planar spanner in a localized manner (total communication cost is $O(n)$ messages).

3.4.1 Centralized Construction for UDG

Our algorithms borrow some idea from the algorithm by Bose *et al.* [58] which constructs a bounded degree and planar spanner for a given points set V. For completeness of presentation, we review the basic steps of their algorithm.

First, it computes the Delaunay triangulation of V, $Del(V)$, and a degree-3 spanning subgraph $BDS(V)$ of $Del(V)$. Then, for each polygon P in $BDS(V)$, their algorithm first orders the nodes according to a geometry based breadth-first search, and processes the nodes of P in increasing order. It prunes this part of the Delaunay triangulation such that each node of P has low degree. The resulting graph is a planar spanner for the nodes of P. By combining all the spanners for each of the polygons, we get a planar spanner of bounded degree. Finally, they run a greedy algorithm in [59] on these structure to bound the total weight from a constant factor of the weight of the Euclidean minimum spanning tree.

They show that the length stretch factor of the final graph is $2\pi(\pi+1)/((3\cos\pi/6)(1+\epsilon))$ 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. In this section, we give a new method for constructing a planar spanner with bounded node degree for $UDG(V)$, and show that it can be converted to a localized method in Section 3.4.2. The basic idea of our method is to combine (localized) Delaunay triangulation and the ordered Yao structure [34].

Centralized Algorithm for UDG We first study how to construct bounded degree planar spanner for UDG in a centralized approach.

Algorithm 3 *Centralized Construction of Planar Spanner with Bounded Degree for $UDG(V)$*

1. Compute the Delaunay triangulation $Del(V)$ of V.
2. Remove edges longer than 1 in $Del(V)$. Call the remaining graph unit Delaunay triangulation $UDel(V)$. For every node u, we know its unit

Delaunay neighbors $N_{UDel}(u)$ and its node degree d_u in $UDel(V)$.

3. Then, find an order π of V as follows: Let $G_1 = UDel(V)$ and $d_{G,u}$ is the node degree of u in graph G. Remove the node u with the smallest degree $d_{G_i,u}$ (smaller ID breaks tie) from graph G_i, and call the remaining graph G_{i+1}, for $1 \leq i \leq n$. Let $\pi_u = i$. Let P_v denote the predecessors of v in π, i.e., $P_v = \{u \in V : \pi_u < \pi_v\}$. Since G_i is always a planar graph, the smallest value of $d_{G_i,u}$ is at most 5. Then, in ordering π, node u has at most 5 edges to its predecessors P_u in $UDel(V)$.

4. Let E be the edge set of $UDel(V)$, E' be the edge set of the desired spanner. Initialize E' to 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' (only consider the unit Delaunay neighbors $N_{UDel}(u)$ of u):

   (a) For node u, let $v_1, v_2, \cdots, v_k$ be the unprocessed neighbors of u in $UDel(V)$ (see Figure 8). Here $k \leq 5$. Then k *open* sectors at node u are defined by rays emanated from u to the processed nodes v_i in $UDel(V)$. For each sector centered at u, we divide it into a minimum number of *open* cones of degree at most α, where $\alpha \leq \pi/3$ is a parameter.

   (b) For each cone, let $s_1, s_2, \cdots, s_m$ be the geometrically ordered neighbors of u in $N_{UDel}(u)$. Notice, $s_1, s_2, \cdots, s_m$ are *unprocessed* nodes. First add the shortest edge us_i in this cone to E', then add to E' all the edges s_js_{j+1}, $1 \leq j < m$. Here edges s_js_{j+1} are not necessarily in $UDel(V)$.

   (c) Mark node u processed.

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

Notice that in the algorithm we use *open* sectors, which means that we do not consider adding the edges on the boundaries.For example, in Figure 8, the cones do not include any edges uv_i. This guarantee the algorithm does not add any edges to node v_i after v_i has been processed. This approach, as we will show it later, bounds the node degree.

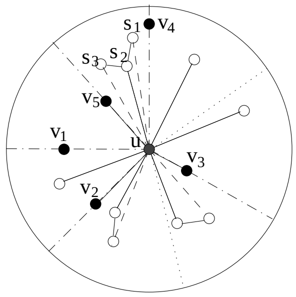


Figure 8: Constructing Planar Spanner with Bounded Degree for $UDG(V)$: Process node u.

Analysis of Centralized Algorithm for UDG

Theorem 3.14 *The maximum node degree of the graph $BPS_1(UDG(V))$ is at most $19 + \lceil \frac{2\pi}{\alpha} \rceil$.*

For example, when $\alpha = \pi/3$, then the maximum node degree is at most 20. Method in [58] does not work for UDG.

Theorem 3.15 *$BPS_1(UDG(V))$ is a planar graph.*

Finally, we prove $BPS_1(UDG(V))$ is a spanner.

Theorem 3.16 *Graph $BPS_1(UDG(V))$ is a t-spanner, where*

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

For example, when $\alpha = \pi/3$, then the spanning ratio is at most $(\frac{\pi}{2}+1)\cdot C_{del}$; when $\alpha = 2\arcsin(\frac{1}{2} - \frac{1}{\pi}) \simeq 20.9^o$, then the spanning ratio is at most $(\frac{\pi}{2}) \cdot C_{del}$. We expect to further improve the bound on the spanning ratio by using the following property: all such Delaunay neighbors s_i is inside the circumcircle of the triangle uvv'. See [60] for the detail of the proof.

Notice that we can build Delaunay triangulation in $O(n \log n)$, and do ordering in time $O(n \log n)$ (using heap for the ordering based on degrees), and Yao structure in $O(n)$ (each edge is processed at most a constant times and there are $O(n)$ edges to be processed). Consequently, the time complexity of our centralized algorithm is $O(n \log n)$, same with the method by Bose *et al.* [58]. However, our algorithm has smaller bounded node degree, and (more importantly) our algorithm has potential to become a localized version for wireless networks application as we will describe next.

3.4.2 Localized Construction

In [63], Wang *et al.* showed that an algorithm presented in [40] does construct a bounded degree spanner for UDG with $O(n)$ messages (with unit $\log n$ bits) under the broadcast communication model, Li *et al.* [35] presented the first algorithm that constructs a planar spanner using only $O(n)$ messages under the broadcast communication model. No localized method is known before for constructing a planar spanner with bounded node degree.

In this section, we reviewed the method in [60] that extended the algorithm presented in previous section to generate bounded degree planar spanner for UDG in a localized manner. The algorithm is based on a planar spanner $LDel^{(2)}(V)$ for UDG proposed by Li *et al.* [35]. They [35] 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. The approach (using $O(n)$ messages total) by Gruia [64] is based on the specific connected dominating set introduced by Alzoubi, Wan, and Frieder [65]. 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.

Finally, the following lemma was proved in [60].

Lemma 3.1 *An edge uv is in $LDel^{(2)}(V)$ iff there is a disk passing through u, and v, which does not contain node from $N_2(u) \cup N_2(v)$ inside.*

Bound the Degree Locally In the previous section, we described a localized algorithm to construct a planar spanner $LDel^{(2)}$ using $O(n)$ messages. However, some node in $LDel^{(2)}$ could have degree as large as $O(n)$. We [60] then gave an efficient method to bound the node degree.

Algorithm 4 *Localized Construction of Planar Spanner with Bounded Degree for $UDG(V)$*

1. First, compute the planar localized Delaunay triangulation $LDel^{(2)}(V)$, so that every node u knows its neighbors $N_{LDel^{(2)}}(u)$ and its node degree d_u in $LDel^{(2)}(V)$.

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: (All nodes unprocessed initially.)

   (a) If a unprocessed node u has the highest local order in its unprocessed neighbors in $LDel^{(2)}(V)$, let k be the number of processed neighbors [2] of u in $LDel^{(2)}(V)$. Node u divides its transmission range to 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 of u in $s_1, s_2, \cdots, s_m$. Node u then tells s_j to add the edges $s_{j-1}s_j$, $s_j s_{j+1}$, $1 \le j \le m$. 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 vv'.

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

Analysis of Localized Algorithm We first show that the algorithm does process all nodes. First of all, the algorithm cannot stop at stage of ordering nodes locally. This can be shown by contradiction. Assume that there are some nodes are unordered. The graph formed by these unordered are

[1] If all unordered neighbors with $d_v \le 5$ has larger ID, we call such query round a *failed round*. Node u performs a new round of queries only if it finds that the unordered neighbors have been reduced from previous failed round.

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

planar, and thus it contains some nodes with at most 5 unordered neighbors. Among these nodes, the node with the smallest ID will perform step 2 (a), and reduces the number of unordered nodes consequently.

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

Additionally, remember that our algorithm checks if $d_u \leq 5$ for computing an ordering locally. Here number 5 can be replaced by any integer larger than 5. Using larger integer may make the algorithm run faster, but on the other hand, it worsens the theoretical bound on the node degree. It is not difficult to show that the constructed final topology still has bounded node degree.

Theorem 3.17 *The maximum node degree of the graph $BPS_2(UDG(V))$ is at most $19 + \lceil \frac{2\pi}{\alpha} \rceil$.*

Notice that, the algorithms [58, 66] always add the edges in the Delaunay triangulation to construct a bounded degree planar spanner for a set of points. Thus, the planarity of the final structure is straightforward. The algorithm we proposed in Section 3.4.1 may add some edges (such as edges $s_i s_{i+1}$ added in step 4(b) of Algorithm 3) that do not belong to the $UDel(V)$. To prove the planarity of the structure $BPS_1(UDG(V))$, we show that no two added diagonal edges intersect. The property that edges, which possibly intersect $s_i s_{i+1}$ in the centralized algorithm, are all Delaunay edges is crucial in the proof of Theorem 3.15. This property does not hold anymore in the localized algorithm. We will show that $BPS_2(UDG(V))$ is a planar graph using a different approach.

Theorem 3.18 *$BPS_2(UDG(V))$ is a planar graph.*

Theorem 3.19 *Graph $BPS_2(UDG(V))$ is a t-spanner, where*

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

Theorem 3.20 *Algorithm 4 uses at most $O(n)$ messages, where each message has $O(\log n)$ bits.*

PROOF. Notice that it was shown in [64] 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 $N_2(u)$, our algorithms can be performed asynchronously. However, collecting $N_2(u)$ need synchronized communication since otherwise, a node cannot determine if it indeed already collected $N_2(u)$.

3.5 Examples of Geometry Structure

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

3.6 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

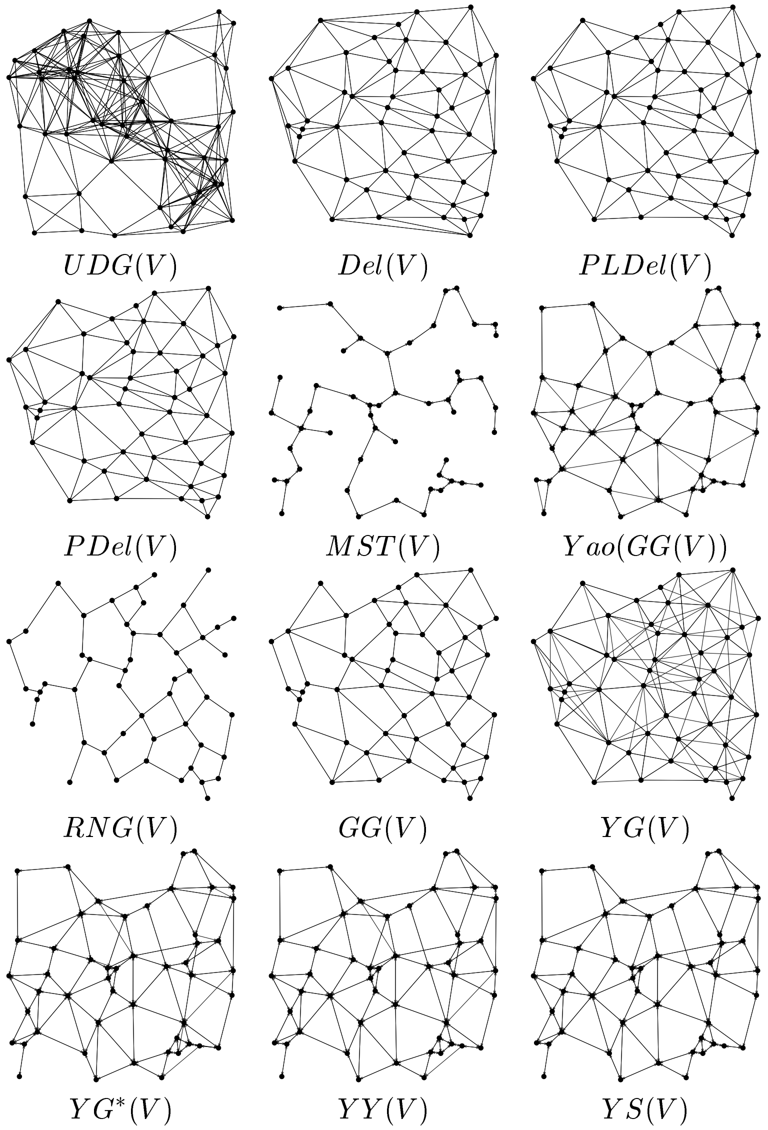


Figure 9: Different topologies from $UDG(V)$.

for a moment in this subsection. 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 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 $\overrightarrow{v_i v_j}$ if and only if $||v_i v_j||^\beta \leq f(v_i)$. 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.* [67] 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 [68, 69]. Let EMST(V) be the Euclidean minimum spanning tree over a point set V. Both [68] and [69] use the power assignment induced by EMST(V). The correctness of using minimum spanning tree is proved in [68]. 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 [68] and [69] 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 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 [68] that the longest edge of the Euclidean minimum spanning tree EMST(V) is always the critical link.

The best distributed algorithm [70, 71, 72] 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 3.21 *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.* [73] and by Clementi *et al.* [74, 75, 76]. Kiroustis *et al.* [73] 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 [73]. The algorithm guarantees the same approximation ratio in any dimensions. Then Clementi *et al.* [74, 75, 76] 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.

4 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.

4.1 Centralized Methods

Guha and Khuller [77] studied the approximation of the connected dominating set problem for general graphs. They gave two different approaches, both of them guarantee approximation ratio of $\Theta(H(\Delta))$. 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. They first pick the node (marked with *black*) with the maximum node degree and all of its neighbors as its children (marked with *gray*). They give two rules for selecting nodes (either gray node or a gray node and a white node adjacent to it) to grow the spanning tree: (1) the gray node with the maximum number of white neighbors; (2) two adjacent nodes, one is gray and one is white, with the maximum number of white neighbors. This approach has approximation ratio $2(H(\Delta) + 1)$, see [77] .

The other approach is first approximating the dominating set and then connecting the dominating set to a connected dominating set. It runs in two phases. At the start of the first phase all nodes are colored white. Each time a vertex is included into the dominating set, we color it black. Dominators are colored gray. In this first phase, the algorithm picks a node at each step and colors it black and colors all its adjacent nodes gray (as dominators). A *piece* is defined as a white node, or a black connected component. At each step, pick a node to color black that gives the maximum non-zero reduction in the number of pieces. In the second phase, recursively connect pairs of black components by choosing a chain of vertices, until there is only one black connected component. The final connected dominating set is the set of black vertices. They [77] 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 [78].

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.* [79] and Marathe *et al.* [80] 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 [81], 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 [11], we have an approximation algorithm (constructing a minimum spanning tree $VirtG$) for MCDS with approximation ratio $3+\epsilon$. Notice that, Berman *et al.* [82] gave an $\frac{4}{3}$ approximation method to connect a dominating set and Robins *et al.* [78] 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 [83]. Recently, Cheng *et al.* [84] designed a PTAS for MCDS in UDG. However, it is difficult to distributize their method efficiently.

4.2 Distributed Methods

Many distributed clustering (or dominating set) algorithms have been proposed in the literature [9, 85, 86, 87, 24, 88]. 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.

4.2.1 Clustering without Geometry Property

For general graphs, Jia *et al.* [89] 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.* [6, 90] 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 [91] 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.* [6, 90] 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.* [6, 90] finds a dominating set and then grows it to a connecting dominating set, the algorithm proposed by Wu and Li [92, 7] 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.* [9] that the approximation ratio of this algorithm could be as large as $\frac{n}{2}$.

Stojmenovic *et al.* [8] 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 [86, 85], the ordered pair of degree and ID [93], and an ordered pair of degree and location [8]. 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 [9] that the worst case approximation ratio of this method is also $\frac{n}{2}$, although it works well in practice.

In [94, 95], 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.

4.2.2 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.* [8] 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.* [10] 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.

Lemma 4.1 *In UDG, for every dominatee node v, it can be connected to at most 5 dominator nodes.*

Generally, it was shown in [10, 11] that for each node (dominator or dominatee), there are at most a constant number of dominators that are at most k units away.

Lemma 4.2 *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 4.3 *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.* [10]. Notice that, in the approach by Stojmenovic *et al.* [8], 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 [9]. Instead, Wan *et al.* suggested to find only one unique shortest path to connect any two dominators that are at most three hops away.

We first briefly review their basic idea of forming a CDS in a distributed manner. Let $\Pi_{UDG}(u, v)$ be the path connecting two nodes u and v in UDG with the smallest number of hops. Let's first consider how to connect two dominators within 3 hops. If the path $\Pi_{UDG}(u, v)$ has two hops, then u finds the dominatee with the smallest ID to connect u and v. If the path $\Pi_{UDG}(u, v)$ has three hops, then u finds the node, say w, with the smallest ID such that w and v are two hops apart. Then node w selects the node with the smallest ID to connect w and v.

Wang and Li [11] and Alzoubi *et al.* [10] discussed in detail some approaches to optimize the communication cost and the memory cost. We briefly review the approaches proposed in [65, 11]. Notice that, for example, it is not obvious how node u can find such node w efficiently. In addition that, using the smallest ID is not efficient because we may have to postpone the selecting of connectors till the node collects the IDs of all its one-hop neighbors. Instead of using the intermediate node with the smallest ID, we pick any node that comes first to the notice of the node that makes the selection of connectors. Their method uses the following primitive messages (some messages are used in forming clusters):

- $\mathsf{IamDominator}(u)$: u tells its 1-hop neighbors that u is a dominator;
- $\mathsf{IamDominatee}(u, v)$: node u tells its 1-hop neighbors that u is a dominatee of node v;
- $\mathsf{2HopsPath}(u, w, v)$: node u tells its 1-hop neighbors that u has a 2-hops path uwv and w is the unique node selected by u among all intermediate nodes that can connect u and v.
- $\mathsf{3HopsPath}(x, u, w, v)$: node x tells its 1-hop neighbors that x has a 3-hops path $xuwv$ and u and w are the uniquely selected nodes among all intermediate nodes. Node u is selected by node x and node w is selected by node u.

The message $\mathsf{IamDominator}(u)$ is only broadcasted at most once by each node; $\mathsf{IamDominatee}(u, v)$ is only broadcasted at most five times by each node u for all possible dominators v from Lemma 4.1; from Lemma 4.2, we know that $\mathsf{2HopsPath}(u, w, v)$ and $\mathsf{3HopsPath}(x, u, w, v)$ are also broadcasted at most a constant times by each node for all possible dominator v.

To save the memory cost of each wireless node, they [11] also designed the following link lists for each node u:

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

Notice that for each node, there are at most five dominators. So the size of link list $\mathsf{Dominators}$ is at most five. Then from Lemma 4.2, for each node u, there are at most ℓ_k number of dominators v that are k-hops apart from u. Therefore, the sizes of link lists $\mathsf{Connector2HopsPath}$, $\mathsf{Connector3HopsPath}$

are bounded by ℓ_2 and ℓ_3 respectively. Then we are in the position to review the distributed algorithm proposed in [65, 11] to find the connectors efficiently. Assume that a maximal independent set is already constructed by a cluster algorithm.

Algorithm 5 *Finding Connectors*

1. Every dominatee node w broadcasts to its 1-hop neighbors a message $\mathsf{IamDominatee}(w, v)$ for each dominator v stored at $\mathsf{Dominators}$.

2. Assume node u receives a message $\mathsf{IamDominatee}(w, v)$ for the first time. If $u \neq v$, v is not in $\mathsf{Dominators}$ list of u, and there is no pair $(*, v)$ in $\mathsf{Connector2HopsPath}$, then u adds (w, v) to $\mathsf{Connector2HopsPath}$. Here $*$ denotes any node ID. If u is a dominatee, then it broadcasts message a $\mathsf{2HopsPath}(u, w, v)$ to its 1-hop neighbors. If node u is a dominator, node u already knows a path uwv to connect a 2-hops apart dominator v.

   Node u will discard any message $\mathsf{IamDominatee}(*, v)$ afterward.

3. When a node w (it must be a dominatee here) receives the message $\mathsf{2HopsPath}(u, w, v)$, node w marks itself as a *connector*, if u is a dominator.

4. Assume a dominator x receives the message $\mathsf{2HopsPath}(u, w, v)$, where $x \neq w$. If there is no triple $(*, *, v)$ in $\mathsf{Connector3HopsPath}$, then x adds (u, w, v) to $\mathsf{Connector3HopsPath}$ and broadcasts the message $\mathsf{3HopsPath}(x, u, w, v)$ to its 1-hop neighbors. Then node x already knows a path $xuwv$ to connect a 3-hops apart dominator v.

5. When a node u (it must be dominatee here) receives the message $\mathsf{3HopsPath}(x, u, w, v)$, node u marks itself as a connector. Node u sends a message to node w asking w to be a connector.

Notice that it is possible that, given any two nodes u and v, the path found by node u to connect v is different from the path found by v to connect u. This increases the robustness of the backbone. When only one connecting path between any pair of dominators is needed, they suggested to add the following restrictions: a dominator node u stores a 2-hops or 3-hops path connecting it to another dominator node v if and only if node u has a smaller ID. In other words, the decision to select the connectors is always made by the node with smaller ID.

The graph constructed by the above algorithm **FindingConnectors** 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'.

Lemma 4.4 *The number of connectors found is at most ℓ_3 times of the minimum. The size of the connected dominating set found by the above algorithm is within a small constant factor of the minimum.*

Let *opt* be the size of the minimum connected dominating set. It was shown [80] that the size of the computed maximal independent set has size at most $4*opt+1$. We already showed that the size of the connected dominating set found by the above algorithm is at most $\ell_3 k+k$, where k is the size of the maximal independent set found by the clustering algorithm. It implies that the found connected dominating set has size at most $4(\ell_3+1)*opt+\ell_3+1$. Consequently, the computed connected dominating set is at most $4(\ell_3+1)$ factor of the optimum (with an additional constant ℓ_3+1).

4.2.3 The Properties of Backbone

It was shown in [65, 11] that the CDS' graph is a sparse spanner in terms of both hops and length, meanwhile CDS has a bounded node degree.

Lemma 4.5 *The node degree of CDS is bounded by* $\max(\ell_3, 5+\ell_2)$.

The above lemma immediately implies that CDS is a sparse graph, i.e., the total number of edges is $O(k)$, where k is the number of dominators. Moreover, the graph CDS' is also a sparse graph because the total number of the links from dominatees to dominators is at most $5(n-k)$. Notice that we have at most $n-k$ dominatees, each of which is connected to at most 5 dominators. The node degree in CDS is bounded, however, the degree of some dominator node in CDS' may be arbitrarily large.

After we construct the backbone CDS and the induced graph CDS', if a node u wants to send a message to another node v, it follows the following procedure. If v is within the transmission range of u, node u directly sends message to v. Otherwise, node u asks its dominator to send this message to v (or one of its dominators) through the backbone. They showed that CDS' (plus all implicit edges connecting dominatees that are no more than one unit apart) is a good spanner in terms of both hops and length.

Lemma 4.6 *The hops stretch factor of CDS' is bounded by a constant* 3 *and the length stretch factor of CDS' is bounded by a constant* 6.

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.* [11] proposed a method to make the graph CDS planar without losing the spanner property of the backbone. Their method applies the localized Delaunay triangulation [35] on top of the induced graph from CDS, denoted by ICDS. It was proved in [35] 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.

Lemma 4.7 *[11] The hops and length stretch factors of $LDel(ICDS)$ are bounded by some constants.*

5 Localized Routings

The geometric nature of the multi-hop ad-hoc wireless networks allows a promising idea: 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* [27], *online* [96, 61], or *distributed* [97].

5.1 Location Service

In order to make the localized 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 ad-hoc 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 ad hoc networks makes it very unlikely that there is at least one location server available for each node. Thus, we will concentrate on distributed location services.

For the wireless ad hoc networks, the location service provided can be classified into four categorizes: *some-for-all*, *some-for-some*, *all-for-some*, *all-for-all*. Some-for-all service means that some wireless nodes provide location services for all wireless nodes. Other categorizations are defined similarly.

An example of all-for-all services is the location services provided in the Distance Routing Effect Algorithm for Mobility (DREAM) by Basagni *et al.* [98]. Each node stores a database of the position information for all other nodes in the wireless networks. Each node will regularly flood packets containing its position to all other nodes. A frequency of the flooding and the range of the flooding is used as a control of the cost of updating and the accuracy of the database.

Using the idea of *quorum* developed in the databases and distributed systems, Hass and Liang [99], Stojmenovic [100] developed quorum based location services for wireless ad-hoc networks. Given a set of wireless nodes V, a quorum system is a set of subset (Q_1, Q_2, $\cdots$, Q_k) of nodes whose union is V. These subsets could be mutually disjoint or often have equal number of intersections. When one of the nodes requires the information of the other, it suffices to query one node (called the representative node of Q_i) from each quorum Q_i. A virtual backbone is often constructed between the representative nodes using a non-position-based methods such as [10, 9]. The updating information of a node v is sent to the representative node (or the nearest if there are many) of the quorum containing v. The difficulty of using quorum is that the mobility of the nodes requires the frequent updating of the quorums. The quorum based location service is often *some-for-some* type.

The other promising location service is based on the quadtree partition of the two-dimensional space [101]. It divides the region containing the wireless network into hierarchy of squares. The partition of the space in [101] is uniform. However, we notice that the partition could be non-uniform if the density of the wireless nodes is not uniform for some applications. Each node v will have the position information of all nodes within the same *smallest* square containing v. This position information of v is also propagated to up-layer squares by storing it in the node with the nearest identity to v in each up-layer square containing v. Using the nearest identity over the smallest identity can avoid the overload of some nodes. The query is conducted accordingly. It is easy to show that it takes about $O(\log n)$ time to update the location of v and to query another node's position information.

5.2 Localized Routing Protocols

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

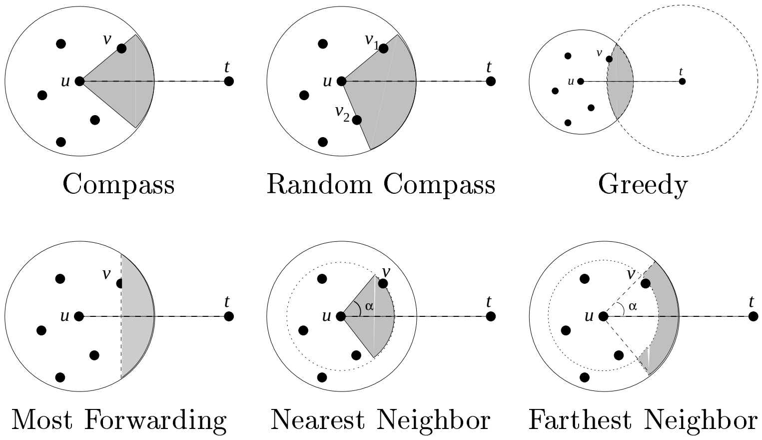


Figure 10: Various localized routing methods. Shaded area is empty and v is next node.

The following routing algorithms on the graphs were proposed recently.

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[102].

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[102].

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 [26].

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 [97].

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 counter-clockwise 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 [61, 103]

Notice that it is shown in [26, 102] 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 is often used to guarantee the packet delivery after simple localized routing heuristics fail [26, 97, 27].

Theorem 5.1 *[103] 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 [103] 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 survey. It is easy to see that there is no memoryless routing method that works in the unit disk graph.

5.3 Quality Guaranteed Protocols

With respect to localized routing, there are several ways to measure the quality of the protocol. Given the scarcity of the power resources in wireless 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.* [61, 103] 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 [62]. 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 11. 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 [62] 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

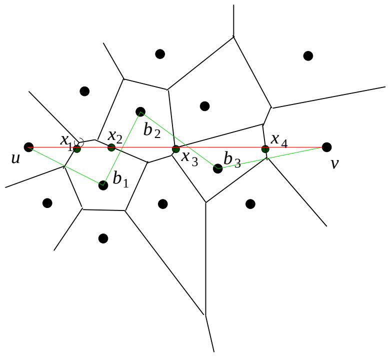


Figure 11: There is a good approximation path using the edges of $tunnel(u, v)$.

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 [62] 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 [103] 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 [104], they further extent their method to any triangulations satisfying the diamond property. Here, a triangulation satisfying the diamond property if 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 $i = 1, 2$.

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.

6 Broadcasting & Multicasting

Minimum-energy broadcast/multicast routing in a simple ad hoc networking environment has been addressed by the pioneering work in [105, 106, 107, 108]. 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 [108], 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 [108], there are sufficient bandwidth and transceiver resources. Under these assumptions, centralized (as opposed to distributed) algorithms were presented by [108] 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.

6.1 Broadcasting

Three greedy heuristics were proposed in [108] 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 [108], 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. The analytical performance is very essential and more convincing in evaluating these heuristics, because one may come up with several seemingly reasonable greedy heuristics. But it is hard to tell from simulation outputs which one is better or worse in the worst case scenario.

For a pure illustration purpose, another slight variation of BIP was discussed in detail in [109]. This greedy heuristic is similar to the Chvatal's algorithm [110] 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 [110], 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 [110], 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.* [109, 111] 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.

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 [112], 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 [113], 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 in the survey, 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.* [105] proved that the minimum-energy broadcast routing problem is a NP-hard problem and obtained a parallel but weaker result to those of [109, 111].

Wan *et al.* [109, 111] gave some lower bounds on the approximation ratios of MST and BIP by studying some special instances in [109, 111]. Their deriving of the upper bounds relies extensively on the geometric structures of Euclidean MSTs. They first observed that as long as the cost of a link is an increasing function of the Euclidean length of the link, the set of MSTs of any point set *coincides* with the set of Euclidean MSTs of the same point set. In particular, for any spanning tree T of a finite point set P, parameter $\sum_{e\in T}\|e\|^2$ achieves its minimum if and only if T is an Euclidean MST of P. For any finite point set P, let $mst\,(P)$ denote an arbitrary Euclidean MST of P. The *radius* of a point set P is defined as

$$\inf_{\mathbf{p}\in P}\sup_{\mathbf{q}\in P}\|\mathbf{pq}\|.$$

Thus, a point set of radius one can be covered by a disk of radius one. A key result in [109, 111] 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 [109, 111].

Theorem 6.1 *[109, 111]* $6 \le c \le 12$.

The proof of this theorem involves complicated geometric arguments; see [109, 111] 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 6.1 implies that for any point set P of radius one and any real number $\beta \geq 2$,

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

The next theorem proved in [109, 111] 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 6.1 *[109, 111] For any point set P in the plane, the total energy required by any broadcasting among P is at least $\frac{1}{c}\sum_{e \in mst(P)} \|e\|^{\beta}$.*

PROOF. Let T be an arborescence for a broadcasting among P with the minimum energy consumption. For any none-leaf node $\mathbf{p}$ in T, let $T_{\mathbf{p}}$ be an Euclidean MST of the point set consisting $\mathbf{p}$ and all children of $\mathbf{p}$ in T. Suppose that the longest Euclidean distance between $\mathbf{p}$ and its children is r. Then the transmission power of node $\mathbf{p}$ is r^{β}, and all children of $\mathbf{p}$ lie in the disk centered at $\mathbf{p}$ with radius r. From the definition of c, we have

$$\sum_{e \in T_{\mathbf{p}}} \left(\frac{\|e\|}{r}\right)^{\beta} \leq c,$$

which implies that

$$r^{\beta} \geq \frac{1}{c} \sum_{e \in T_{\mathbf{p}}} \|e\|^{\beta}.$$

Let T^{*} denote the spanning tree obtained by superposing of all $T_{\mathbf{p}}$'s for non-leaf nodes of T. Then the total energy required by T is at least $\frac{1}{c}\sum_{e \in T^{*}} \|e\|^{\beta}$, which is further no less than $\frac{1}{c}\sum_{e \in mst(P)} \|e\|^{\beta}$. This completes the proof. ◻

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 6.1, 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 6.1, this observation leads to the following theorem.

Theorem 6.2 *[109, 111] 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 6.2 *[109, 111] 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 $\sum_{e \in mst(P)} \|e\|^{\beta}$.*

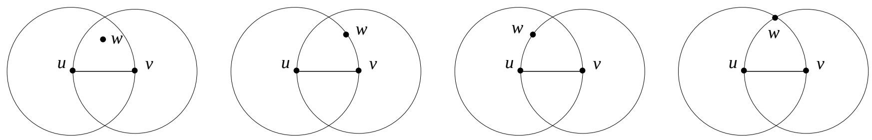
6.2 Approximate MST of UDG Locally

The best distributed algorithm [70, 71, 72] 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 that we can construct the minimum spanning tree in a distributed manner using $O(n \log n)$ messages. Unfortunately, even for wireless network modeled by a ring, the $O(n \log n)$ number of messages is still necessary for constructing the minimum spanning tree.

Given a graph G, let $\omega_b(G) = \sum_{e \in G} \|e\|^b$. We [114] recently presented the first localized method to construct a bounded degree planar connected structure whose total edge length is within a constant factor of that of the minimum spanning tree. The total communication cost of our method is $O(n)$, and every node only uses its two-hop information to construct such structure. We showed that the energy consumption using this structure is within $O(n^{\beta-1})$ of the optimum, i.e., $\omega_{\beta}(H) = O(n^{\beta-1}) \cdot \omega_{\beta}(MST)$ 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 $\omega_{\beta}(RNG) = \Theta(n^{\beta}) \cdot \omega_{\beta}(MST)$.

Our low-weight structure 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. We then naturally extend 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 12 for an illustration when an edge uv is not included in the modified relative neighborhood graph. We denote such structure by RNG' hereafter. Obviously, RNG' is a subgraph of traditional RNG. We [114] proved that RNG' has a maximum node degree 6 and still contains a minimum spanning tree as a subgraph.



(1) (2) (3) (4)

Figure 12: Which edges are not in the modified RNG.

So far RNG' is the previously best known connected structures that can be constructed locally and has a small total edge weight. As shown in [114], its total weight could still be as large as $O(n)$ times of $\omega(MST)$.

We then give the first localized algorithm that constructs a low-weighted structure using only some two hops information.

Algorithm 6 *Construct Low Weight Structure*

1. All nodes together construct the modified relative neighborhood 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. If node u received a message informing existence of edge xy from its neighbor x, for each edge uv in RNG', if uv is the longest among 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.

Let H be the final structure formed by all remaining edges in RNG', and we call it low weighted modified relative neighborhood graph. Obviously, if an edge uv is kept by node u, then it is also kept by node v.

Theorem 6.3 *[114] 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 satisfies the *isolation property* (defined in [115]). We [114] also showed that the final structure contains EMST of UDG as a subgraph. It was also shown in [114] that it is *impossible* to construct a low-weighted structure using only one hop neighbor information.

6.3 Forwarding Neighbors

The simplest broadcasting mechanism is to let every node retransmit the message to all its one-hop neighbors when receiving the first copy of the message, which is called *flooding* in the literature. Despite its simplicity, flooding is very inefficient and can result in high redundancy, contention, and collision. One approach to reducing the redundancy is to let a node only forward the message to a subset of one-hop neighbors who together can cover the two-hop neighbors. In other words, when a node retransmits a message to its neighbors, it explicitly ask a subset of its neighbors to relay the message.

Călinescu *et al.* [116] gave two practical heuristics for this problem (they called selecting forwarding neighbors). The first algorithm runs in time $O(n \log n)$ and returns a subset with size at most 6 times of the minimum. The second algorithm has an improved approximation ratio 3, but with running time $O(n^2)$. Here n is the number of total two-hop neighbors of a node. When all two-hop neighbors are in the same quadrant with respect to the source node, they gave an exact solution in time $O(n^2)$ and a solution with approximation factor 2 in time $O(n \log n)$. Their algorithms partition the region surrounding the source node into four quadrants, solve each quadrants using an algorithm with approximation factor α, and then combine these solutions. They proved that the combined solution is at most 3α times of the optimum solution.

Their approach assumes that every node u can collect its 2-hop neighbors $N_2(u)$ efficiently. Notice that, the 1-hop neighbors of every node u can be collected efficiently by asking each node to broadcast its information to its 1-hop neighbors. Thus all nodes get their 1-hop neighbors information by using total $O(n)$ messages. However, until recently, it is unknown how to collect the 2-hop neighbors information with $O(n)$ communications. The simplest broadcasting of 1-hop neighbors $N_1(u)$ to all neighbors u does let all nodes in $N_1(u)$ to collect their corresponding 2-hop neighbors. However, the total communication cost of this approach is $O(m)$, where m is the total number of links in UDG. Recently, Călinescu [64] proposed an efficient approach to collect $N_2(u)$ using the connected dominating set [10, 11] as forwarding nodes. Assume that the node position is known. He proved that the approach takes total communications $O(n)$, which is optimum within a constant factor.

7 Stochastic Geometry

In wireless ad hoc networks, one of the critical issues is that, for every pair of nodes, there is a path connecting them, i.e., the network is connected. With this in mind, Gupta and Kumar [117] studied what is the critical power at which each node has to transmit so as to guarantee the connectivity of the network asymptotically.

7.1 Background

Given an event Y, let $Pr(Y)$ be the probability of Y. Given a random variable X, we denote the expected value of X by $E[X]$, i.e., $E[X] = \sum_x x \cdot Pr(X = x)$ for discrete variables. As standard, we write log for base-2 logarithm and ln for natural logarithm. We say a function $f(n) \to a$ if $\lim_{n\to\infty} f(n) = a$.

A point set is said to a *random point process*, denoted by $\mathcal{X}_n$, if it consists of n independent points each of which is uniformly distributed over the region. The standard probabilistic model of *homogeneous Poisson process* is characterized by the property that the number of nodes in a region is a random variable depending only on the area (or volume in higher dimensions) of the region. In other words,

- The probability that there are exactly k nodes appearing in any region Ψ of area A is $\frac{(\lambda A)^k}{k!} \cdot e^{-\lambda A}$.
- For any region Ψ, the conditional distribution of nodes in Ψ given that exactly k nodes in the region is *joint uniform*.

Here after, we let $\mathcal{P}_n$ be a homogeneous Poisson process of intensity n on the unit cube $\mathcal{C} = [-0.5, 0.5] \times [-0.5, 0.5]$.

7.2 Connectivity

Given a finite set of n points V in a metric space and a positive real number r, let the r-graph, denoted by $G(V, r)$, be the graph with vertex set V

and with an edge connecting each pair of points separated by a distance of at most r. Two paths in a graph are said to be *vertex independent* if the only common vertices are the end-vertex of both paths. A graph is called k-vertex connected if, for each pair of vertices, there are k mutually vertex independent paths connecting them. Two paths are said to be *edge independent* if there is no common edge between them. Equivalently, a graph is called k-edge connected if, for each pair of vertices, there are k mutually edge independent paths connecting them. The *vertex connectivity*, denoted by $\kappa(G)$, of a graph G is the maximum k such that G is k vertex connected. The *edge connectivity*, denoted by $\xi(G)$, of a graph G is the maximum k such that G is k edge connected. The minimum degree of a graph G is denoted by $\delta(G)$ and the maximum degree of a graph G is denoted by $\Delta(G)$. Clearly, for any graph G,

$$\kappa(G) \leq \xi(G) \leq \delta(G) \leq \Delta(G).$$

A graph property is called *monotone increasing* if G has such property then all graphs on the same vertex set containing G as a subgraph have this property. Let $\mathcal{Q}$ be any monotone increasing property of graphs, for example, the connectivity, the k-edge connectivity, the k-vertex connectivity, the minimum node degree at least k, and so on. The *hitting radius* $\varrho(V, \mathcal{Q})$ is the infimum of all r such that graph $G(V, r)$ has property $\mathcal{Q}$. For example, $\varrho(V, \kappa \geq k)$ is the minimum radius r such that $G(V, r)$ is at least k vertex connected; $\varrho(V, \delta \geq k)$ is the minimum radius r at which the graph $G(V, r)$ has the minimum degree at least k. It is obvious that, for any V,

$$\varrho(V, \kappa \geq k) \geq \varrho(V, \delta \geq k).$$

Let $P_k(\mathcal{X}_n, r(n))$ be the probability that a graph in $\mathcal{G}(\mathcal{X}_n, r(n))$ is k-connected.

It was proved by Penrose [118] that, given any metric l_p with $2 \leq p \leq \infty$ and any positive integer k,

$$\lim_{n \to \infty} Pr\left(\varrho(\mathcal{X}_n, \kappa \geq k) = \varrho(\mathcal{X}_n, \delta \geq k)\right) = 1.$$

This result says that, if n is large enough, then with high probability, if we start with isolated n random points $\mathcal{X}_n$ in C, and add the edges in order of the increasing length to connect the points of $\mathcal{X}_n$, the resulting graph becomes k vertex connected at the moment when the minimum degree of the graph becomes k. This result is analogous to the well-known results in the graph theory [119] that graph becomes k vertex connected when it achieves the minimum degree k if we add the edges randomly and uniformly from $\binom{n}{2}!$ possibilities. Similarly, instead of considering $\mathcal{X}_n$, Penrose also considered a homogeneous Poisson point process with intensity n on the unit-cube $\mathcal{C}$. Penrose gave loose upper and lower bound on the hitting radius $r_n = \varrho(\mathcal{X}_n, \delta \geq k)$ as

$$\frac{\ln n}{2d+1} \leq n \cdot r_n^d \leq d! \cdot 2 \cdot \ln n$$

for homogeneous Poisson point process on a d-dimensional unit cube.

The connectivity of random graphs, especially the geometric graphs and its variations, have been considered in the random graph theory literature [119] in the stochastic geometry literature [120, 121, 118, 122], and the wireless ad hoc network literature [123, 117, 124, 125].

Let $\mathcal{B}(n, p(n))$ be the set of graphs on n nodes in which each edge of the completed graphs K_n is chosen independently with probability $p(n)$. Then it has been shown that the probability that a graph in $\mathcal{B}(n, p(n))$ is connected goes to one if $p(n) = \frac{\ln n + c(n)}{n}$ for any $c(n) \to \infty$. Notice that, although their asymptotic expressions are the same with that by Gupta and Kumar [117], but we can not apply this to the wireless model as, in wireless network, the existences of two edges are not independent, and we do not choose edges from the completed graph using Bernoulli model.

Bollobás and Thomason proved that, if $c(n) \to \infty$, $c(n) \leq \ln \ln \ln n$ and $p(n) = \frac{\ln n + (k-1) \ln \ln n - c(n)}{n}$, then almost no graph from $\mathcal{B}(n, p(n))$ contains a non-trivial $(k-1)$-separator. Notice that a graph with minimum degree k is k-connected unless it contains a non-trivial $(k-1)$-separator. Thus, this result by Bollobás and Thomason implies that if $p(n) = \frac{\ln n + (k-1) \ln \ln n - c(n)}{n}$ then graphs from $\mathcal{B}(n, p(n))$ almost surely have minimum degree $k-1$ and thus almost surely are k-connected.

Another closely related question is the *coverage* problem: disks of radius r are placed in a two-dimensional unit-area disk $\mathcal{D}$ with centers from a Poisson point process with intensity n. A result shown by Hall [126] implies that, if $\pi \cdot r^2 = \frac{\ln n + \ln \ln n + c(n)}{n}$ and $c(n) \to \infty$, then the probability that there is a vacancy area in $\mathcal{D}$ is 0 as n goes infinity; if $c(n) \to -\infty$, then the probability that there is a vacancy in $\mathcal{D}$ is at least $\frac{1}{20}$. Thus, for $c(n) \to +\infty$,

$$\pi \cdot \varrho(\mathcal{P}, \kappa)^2 \leq 4 \frac{\ln n + \ln \ln n + c(n)}{n}.$$

Given n nodes V *randomly and independently* distributed in a unit-area disk $\mathcal{D}$, Gupta and Kumar [117] showed that the graph $G(V, r(n))$ is connected almost surely if $\pi \cdot r_n^2 \geq \frac{\ln n + c(n)}{n}$ for any $c(n)$ with $c(n) \to \infty$

as n goes infinity. This bound is tight as they also proved that the graph $\mathcal{G}(\mathcal{X}_n, r(n))$ is asymptotically disconnected with positive probability if

$$\pi \cdot r(n)^2 = \frac{\ln n + c(n)}{n}$$

and $\limsup_n c(n) < +\infty$. Notice, they actually derived their results for a homogeneous Poisson process of points instead of the independent and uniform point process. They showed that the difference between them is negligible. This result is essentially the same as [120], which was developed earlier by mathematicians, but not introduced to the computer science community.

8 Conclusion

Wireless ad hoc networks has attracted considerable attentions recently due to its potential wide applications in various areas and the moreover, the ubiquitous computing. Many excellent researches have been conducted to study the electronic part of the wireless ad hoc networks, the networking part of the wireless ad hoc networks. For networking, there are also many interesting topics such as topology control, routing, energy conservation, QoS, mobility management, and so on. In this survey, 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 ad hoc networks. Nevertheless, there are still many excellent results are not covered in this survey due to space limit.

9 Acknowledgment

The author would like to thank the anonymous reviewers for the helpful suggestions to improve this survey.

References

[1] 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.

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

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

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

[5] P. Sinha, R. Sivakumar, and V. Bharghavan, "Cedar: Core extraction distributed ad hoc routing," in *Proc. of IEEE INFOCOMM '99*, 1999.

[6] 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)*, 1997, vol. 1, pp. 376–380.

[7] 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*, vol. 3, pp. 63–84, 2001.

[8] 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*, vol. 13, no. 1, pp. 14–25, 2002.

[9] 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.

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

[11] 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.

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

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

[14] 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*, vol. 17, no. 8, pp. 1415–1425, August 1999.

[15] 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.

[16] 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*, February 1999, pp. 90–100.

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

[18] Vincent D. Park and M. Scott Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *IEEE INFOCOM*, 1997.

[19] 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.*, Mar 1996, pp. 480–86.

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

[21] G. Pei, M. Gerla, and T.-W. Chen, "Fisheye state routing: A routing scheme for ad hoc wireless networks," in *Proceedings of ICC 2000*, 2000.

[22] 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.

[23] 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*, vol. 17, no. 8, pp. 1369–1379, August 1999.

[24] 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*, 1991, pp. 204–210.

[25] 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*, vol. 4, no. 1, pp. 59–70, 2002.

[26] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *ACM/Kluwer Wireless Networks*, vol. 7, no. 6, pp. 609–616, 2001, 3rd int. Workshop on Discrete Algorithms and methods for mobile computing and communications, 1999, 48-55.

[27] B. Karp and H. T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *ACM/IEEE International Conference on Mobile Computing and Networking*, 2000.

[28] Martin Mauve, Jorg Widmer, and Hannes Harenstein, "A survey on position-based routing in mobile ad hoc networks," *IEEE Network Magazine*, vol. 15, no. 6, pp. 30–39, 2001.

[29] Godfried T. Toussaint, "The relative neighborhood graph of a finite planar set," *Pattern Recognition*, vol. 12, no. 4, pp. 261–268, 1980.

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

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

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

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

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

[35] 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)*, 2002, vol. 3.

[36] S. Datta, I. Stojmenovic, and J. Wu, "Internal node and shortcut based routing with guaranteed delivery in wireless networks," *Cluster Computing*, vol. 5, no. 2, pp. 169–178, 2002.

[37] 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.

[38] 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*, vol. 5, no. 2, pp. 37–44, 2002.

[39] 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.

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

[41] 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*, 1994, pp. 48–59.

[42] 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.

[43] 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.

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

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

[46] 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.

[47] P.L. Chew, "There is a planar graph as good as the complete graph," in *Proceedings of the 2nd Symposium on Computational Geometry*, 1986, pp. 169–177.

[48] 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)*, 2001, pp. 564–567.

[49] 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.

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

[51] 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.

[52] 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.

[53] 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," *J. Algorithms*, vol. 26, no. 2, pp. 238–274, 1999.

[54] M. Grünewald, T. Lukovszki, C. Schindelhauer, and K. Volbert, "Distributed maintenance of resource efficient wireless network topologies," 2002, Submitted for publication.

[55] J. M. Keil and C. A. Gutwin, "Classes of graphs which approximate the complete euclidean graph," *Discr. Comp. Geom.*, vol. 7, pp. 13–28, 1992.

[56] Xiang-Yang Li, Ivan Stojmenovic, and Yu Wang, "Partial delaunay triangulation and degree limited localized bluetooth scatternet formation," in *AdHocNow*, 2002.

[57] 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.

[58] Prosenjit Bose, Joachim Gudmundsson, and Michiel Smid, "Constructing plane spanners of bounded degree and low weight," in *Proceedings of European Symposium of Algorithms*, 2002.

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

[60] Xiang-Yang Li and Yu Wang, "Localized construction of bounded degree planar spanner for wireless networks," 2003, Submitted for publication.

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

[62] D.P. Dobkin, S.J. Friedman, and K.J. Supowit, "Delaunay graphs are almost as good as complete graphs," *Discr. Comp. Geom.*, pp. 399–407, 1990.

[63] Yu Wang, Xiang-Yang Li, and Ophir Frieder, "Distributed spanner with bounded degree for wireless networks," *International Journal of Foundations of Computer Science*, 2002, Accepted for publication.

[64] Gruia Călinescu, "Computing 2-hop neighborhoods in ad hoc wireless networks," 2002, Submitted for publication.

[65] 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.

[66] Xiang-Yang Li and Yu Wang, "Efficient construction of low weight bounded degree planar spanner," 2002, Submitted for publication.

[67] 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.

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

[69] 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.

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

[71] R. Gallager, P. Humblet, and P. Spira, "A distributed algorithm for minimumweight spanning trees," *ACM Transactions on Programming Languages and Systems*, vol. 5, no. 1, pp. 66–77, 1983.

[72] 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*, 1993, pp. 659–668.

[73] 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.

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

[75] 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.

[76] 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.

[77] Sudipto Guha and Samir Khuller, "Approximation algorithms for connected dominating sets," in *European Symposium on Algorithms*, 1996, pp. 179–193.

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

[79] H.B. Hunt, M. V. Marathe, V. Radhakrishnan, S. Ravi, D. J. Rosenkrantz, and R. E. Stearns, "Nc-approximation schemes for np and pspace hard problems for geometric graphs," *Journal of Algorithms*, vol. 26, pp. 238–274, 1998.

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

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

[82] P. Berman, M. Furer, and A. Zelikovsky, "Applications of matroid parity problem to approximating steiner trees," Tech. Rep. 980021, Computer Science, UCLA, 1998.

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

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

[85] Chunhung Richard Lin and Mario Gerla, “Adaptive clustering for mobile wireless networks,” *IEEE Journal of Selected Areas in Communications*, vol. 15, no. 7, pp. 1265–1275, 1997.

[86] I. Chlamtac and A. Farago, “A new approach to design and analysis of peer to peer mobile networks,” *Wireless Networks*, vol. 5, pp. 149–156, 1999.

[87] 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*, 2000, vol. 1, pp. 32–41.

[88] 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.

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

[90] 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.

[91] V. Chvatal, “A greedy heuristic for the set-covering problem,” *Mathematics of Operation Research*, vol. 4, no. 3, pp. 233–235, 1979.

[92] 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*, 2000, pp. 189–197.

[93] 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.

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

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

[96] 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*, 2000, pp. 47–59.

[97] 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*, vol. 12, no. 10, 2001.

[98] 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.

[99] Z. Haas and B. Liang, "Ad-hoc mobility management with uniform quorum systems," *IEEE/ACM Transactions on Networking*, vol. 7, no. 2, pp. 228–240, 1999.

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

[101] 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)*, 2000, pp. 120–130.

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

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

[104] 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)*, 2001, pp. 35–44.

[105] 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*, 2001, pp. 121–131.

[106] 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.

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

[108] 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*, 2000, pp. 586–594.

[109] Peng-Jun Wan, G. Calinescu, Xiang-Yang Li, and Ophir Frieder, "Minimum-energy broadcast routing in static ad hoc wireless networks," in *IEEE Infocom*, 2001.

[110] V. Chvátal, "A greedy heuristic for the set-covering problem," *Mathematics of Operations Research*, vol. 4, no. 3, pp. 233–235, 1979.

[111] 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.

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

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

[114] Xiang-Yang Li, "Efficient local construction of low weight topology and its applications in broadcast for wireless networks," 2002, Submitted for publication.

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

[116] Gruia Călinescu, Ion Măndoiu, Peng-Jun Wan, and Alexander Zelikovsky, "Selecting forwarding neighbors in wireless ad hoc networks," in *ACM DialM*, 2001.

[117] 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.

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

[119] B. Bollobás, *Random Graphs*, Cambridge University Press, 2001.

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

[121] Mathew Penrose, "Extremes for the minimal spanning tree on normally distributed points," *Advances in Applied Probability*, vol. 30, pp. 628–639, 1998.

[122] Mathew Penrose, "A strong law for the longest edge of the minimal spanning tree," *Annals of Probability*, vol. 27, pp. 246–260, 1999.

[123] Matthias Grossglauser and David Tse, "Mobility increases the capacity of ad-hoc wireless networks," in *INFOCOMM*, 2001, vol. 3, pp. 1360 –1369.

[124] P. Gupta and P. Kumar, "Capacity of wireless networks," Tech. Rep., University of Illinois, Urbana-Champaign, 1999.

[125] Olivier Dousse Patrick, "Connectivity in ad-hoc and hybrid networks," in *IEEE INFOCOM*, 2002.

[126] P. Hall, *Introduction to the Theory of Coverage Processes*, J. Wiley and Sons, New York, 1988.