Algorithmic, geometric and graphs issues in wireless networks — Source link

Xiang-Yang Li

Institutions: Illinois Institute of Technology

Published on: 01 Mar 2003 - Wireless Communications and Mobile Computing (Wiley)

Topics: Wireless ad hoc network, Connected dominating set, Minimum spanning tree, Wireless network and Spanning tree

Algorithmic, geometric and graphs issues in wireless networks

Xiang-Yang Li *

September 19, 2002

Abstract

We present an overview of the recent progress of applying computational geometry techniques to solve some questions, such as topology construction and broadcasting, in wireless ad hoc networks. Treating each wireless device as a node in a two dimensional plane, we model the wireless networks by unit disk graphs in which two nodes are connected if their Euclidean distance is no more than one. We first summarize the current status of constructing sparse spanners for unit disk graphs with various combinations of the following properties: bounded stretch factor, bounded node degree, planar, and bounded total edges weight (compared with the minimum spanning tree). Instead of constructing subgraphs by removing links, we then review the algorithms for constructing a sparse backbone (connected dominating set), i.e., subgraph from the subset of nodes. We then review some efficient methods for broadcasting and multicasting with theoretic guaranteed performance.

Keywords: Computational geometry, wireless networks, network optimization, power consumption, routing, spanner, topology control.

1 Introduction

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 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

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

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.

It is common to separate the network design problem from the management and control of the network in the communication network literature. The separation is very convenient and helps to significantly simplify these two tasks, which are already very complex on its own. Nevertheless, there is a price to be paid for this modularity as the decisions made at the network design phase may strongly affect the network management and control phase. In particular, if the issue of designing efficient routing schemes is not taken into account by the network designers, then the constructed network might not suited for supporting a good routing scheme. Wireless 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.

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.

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.

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 have 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 scarcity memory 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. Mauve *et al.* [26] conducted an excellent survey of position-based localized routing protocols. Thus, we will not repeat it here.

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. 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 $c = 0$.

The rest of the survey is organized as follows. In Section 2, we review some geometry structures, define the graph spanners, and introduce the localized algorithm concept. In Section 3, we review the structures with bounded stretch factor, or with bounded node degree, or planar structures. In Section 4, we summarize 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 5, 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. Section 6 reviews the broadcasting protocols that We conclude the survey in Section 7 by pointing out some possible future research questions.

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 [27]. 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* [28] $\mathrm{GG}(G)$ 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 RNG(G) is a subgraph of the Gabriel graph GG(G). For an undirected and connected graph G, both GG(G) and RNG(G) are connected and contain the minimum spanning tree of G.

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 [29], 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 [29].

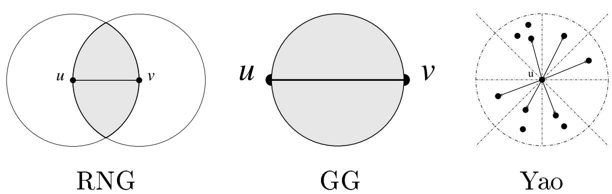

Figure 1: The definitions of RNG, GG, and Yao on point set. Left: The lune using uv is empty for RNG. Middle: The diametric circle using uv is empty for GG. Right: The shortest edge in each cone is added as a neighbor of u for Yao.

Notice all these definitions are exactly the conventional definitions [30, 31, 32, 33] 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 $K(V)$.

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.

Due to the limited resources of the wireless nodes, it is preferred that the underlying network topology can be constructed in a localized manner. Stojmenovic *et al.* first defined what is a localized algorithm in several pioneering papers [34, 35]. 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. Gabriel graph was used as a planar subgraph in the Face routing protocol [34, 36, 37] and the GPSR routing protocol [38]. Right hand rule is used to guarantee the delivery of the packet in [34]. Relative neighborhood graph RNG was used for efficient broadcasting (minimizing the number of retransmissions) in one-to-one broadcasting model in [39]. In this survey, we are interested in localized algorithms that construct sparse and power efficient network topologies.

3 Spanners

Spanners have been studied intensively in recent years [40, 41, 42, 43, 44, 45, 46, 47, 32]. 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 $u, v \in V$, $d_H(u, v) \le t \cdot d_G(u, v)$. The value of t is called the *stretch factor*.

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 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 $\rho_{H_1}(G) \ge \rho_{H_2}(G)$.

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.

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 $(n-1)^\beta$. Li *et al.* [48] showed that it is actually $n-1$. Thus, any graph contains the Euclidean minimum spanning tree has the power stretch factor at most $n-1$.

The Gabriel graph has length stretch factor between $\frac{\sqrt{n}}{2}$ and $\frac{4\pi\sqrt{2n-4}}{3}$[43]. Li *et al.* [48] proved that its power stretch factor is at most $\left(\frac{4\pi\sqrt{2n-4}}{3}\right)^\beta$.

The Yao graph 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: its power stretch factor is at most $\frac{1}{1-(2\sin\frac{\pi}{k})^\beta}$.

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.

The two-phased approach by Wattenhofer *et al.* [50] 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 $p_{u,\alpha}$. They proved 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 2 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.

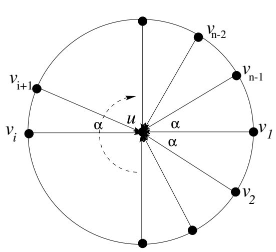

Figure 2: Node u has degree (or in-degree) $n-1$.

3.2.1 Sink Structure

Arya *et al.* [41] 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 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. Each node u computes the set of in-coming nodes $I(u) = \{v \mid \overrightarrow{vu} \in \overrightarrow{YG}_k(V)\}$.
2. Node u uses Tree(u,$I(u)$) to build tree $T(u)$.

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

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

Figure 3 (a) illustrates a directed star centered at u and Figure 3 (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.

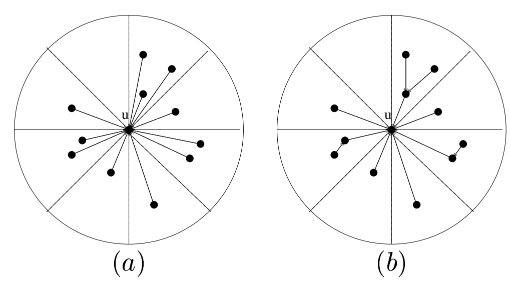

Figure 3: (a) Star formed by links toward to u. (b) Directed tree $T(u)$ sinked at u.

The algorithm uses a directed tree $T(u)$ to replace the directed star for each node u. Therefore, if nodes u and v are connected by a path in $\overrightarrow{YG}_k$, they are also connected by a path in $\overrightarrow{YG}^*_k$. It is already known that $\overrightarrow{YG}_k$ is strongly connected if UDG(V) is connected, so does $\overrightarrow{YG}^*_k$. Li *et al.* [48] showd that 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 $(k+1)^2 - 1$, and 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 $YY_k(V)$. The directed graph $\overrightarrow{YY}_k(V)$ is strongly connected if UDG(V) is connected and $k > 6$, see [49].

It was proved in [52, 49] 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.

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.

Li *et al.* [48] proved that the graph $YS_k(V)$ is strongly connected if UDG(V) is connected and $k \geq 6$ by showing that RNG is a subgraph of $YS_k(V)$ if $k \geq 6$. This 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.* [43]. 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 4 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.

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 $\frac{2\pi}{3\cos\frac{\pi}{6}} \approx 2.42$. 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.* [55] con-

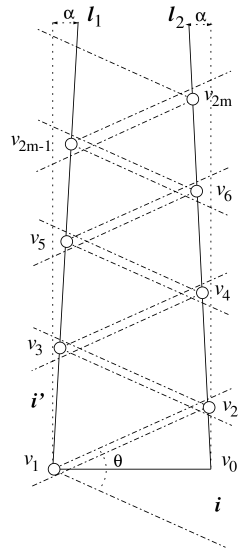

Figure 4: An example that $YS_k(V)$ has a large stretch factor.

sidered the *unit Delaunay triangulation* UDel(V) for planar spanner of UDG, which is a subset of the Delaunay triangulation. In [55], they proved that UDel(V) is a t-spanner of the unit disk graph UDG(V). Specifically, they showed that, 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 [56] showed that the Delaunay triangulation is a t-spanner for a constant $t \approx 2.42$. They proved this using 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 [56] 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.* [55] 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.

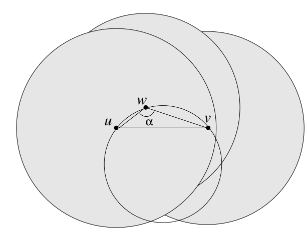

Figure 5: LDel: The $disk(u,v,w)$ is not necessarily covered by unit disks centered at u and v. But it is empty of other vertices from $N_1(u) \cup N_1(v) \cup N_1(w)$.

A sequence of localized Delaunay graphs $LDel^{(k)}(V)$, where $1 \leq k \leq n$ is defined. All graphs are t-spanner of the unit-disk graph with the following properties [55]: (1) $UDel(V) \subseteq LDel^{(k)}(V)$, for all $1 \leq k \leq n$; (2) $LDel^{(k+1)}(V) \subseteq LDel^{(k)}(V)$, for all $1 \leq k \leq n$; (3) $LDel^{(k)}(V)$ are planar graphs for all $2 \leq k \leq n$; and (4) $LDel^{(1)}(V)$ is not always planar.

Notice that, although $LDel^{(1)}(V)$ is not a planar graph, graph $LDel^{(1)}(V)$ has thickness 2; see [55].

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 [32] and the θ-graph [56] (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 description of the algorithm constructing $LDel$, the order of three nodes in a triangle is immaterial.

Algorithm 1 *Localized Unit Delaunay Triangulation*

1. Each node u broadcasts its identity and location and listens to the messages from other nodes.
2. Node u computes $Del(N_1(u))$ of its 1-neighbors $N_1(u)$ and u itself.
3. If angle $\angle wuv \geq \frac{\pi}{3}$ and $\|vw\| \leq 1$, node u broadcasts a message $\mathsf{proposal}(u,v,w)$ to form a 1-localized Delaunay triangle $\triangle uvw$ in $LDel^{(1)}(V)$.
4. When u receives a message $\mathsf{proposal}(u,v,w)$, u accepts the proposal of constructing $\triangle uvw$ if $\triangle uvw$ belongs to $Del(N_1(u))$ by broadcasting message $\mathsf{accept}(u,v,w)$; otherwise, it rejects the proposal by broadcasting message $\mathsf{reject}(u,v,w)$.
5. Node u adds the edges uv and uw to its set of incident edges if $\triangle uvw$ is in $Del(N_1(u))$, $\|vw\| \leq 1$, 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. Consequently, the local Delaunay construction method generates $LDel^{(1)}(V)$ with total communication cost $O(n \log n)$ bits [55].

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

Algorithm 2 *Planarize $LDel^{(1)}(V)$*

1. Each node u broadcasts the Gabriel edges incident on u and the triangles $\triangle uvw$ of $LDel^{(1)}(V)$.
2. Assume 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 node u broadcasts all the triangles which it has not removed in the previous step.
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 $\triangle uvw$ 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 $PLDel(V)$ contains $LDel^{(2)}(V)$. 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 is $O(n)$. In summary, $PLDel(V)$ 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 [57] 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, the partial Delaunay triangulation is a subgraph of $UDel(V)$. Thus, the spanning ratio of the partial Delaunay triangulation could be very large.

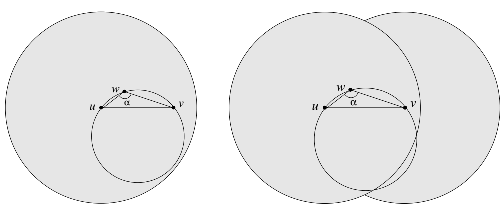

Figure 6: Left: Only one hop information is known to u. Then it requires $disk(u,v,w)$ to be covered by the transmission range of u (denoted by the shaded region) and is empty of neighbors of u. Right: Node u knows $N_1(u)$ and node v knows $N_1(v)$. The circumcircle $disk(u,v,w)$ is covered by the union of the transmission ranges of u and v and is empty of other vertices.

3.3.3 Restricted Delaunay Graph

Gao *et al.* [58] 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 $E(u)$.

They proved that 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.

Figure 7 gives some concrete examples of the geometry structures discussed before.

4 Transmission Power Control

In the previous sections, we have assumed that the transmission power of every node is equal and is normalized to one unit. We relax this assumption for a moment in this 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 $\overrightarrow{v_iv_j}$ if and only if $||v_iv_j||^\beta \leq f(v_i)$. We call a transmission power assignment f *complete* if the communication graph G_f is strongly connected. Recall that a directed graph is strongly connected if, for any given pair of ordered nodes s and t, there is a directed path from s to t.

The *maximum-cost* of a transmission power as-

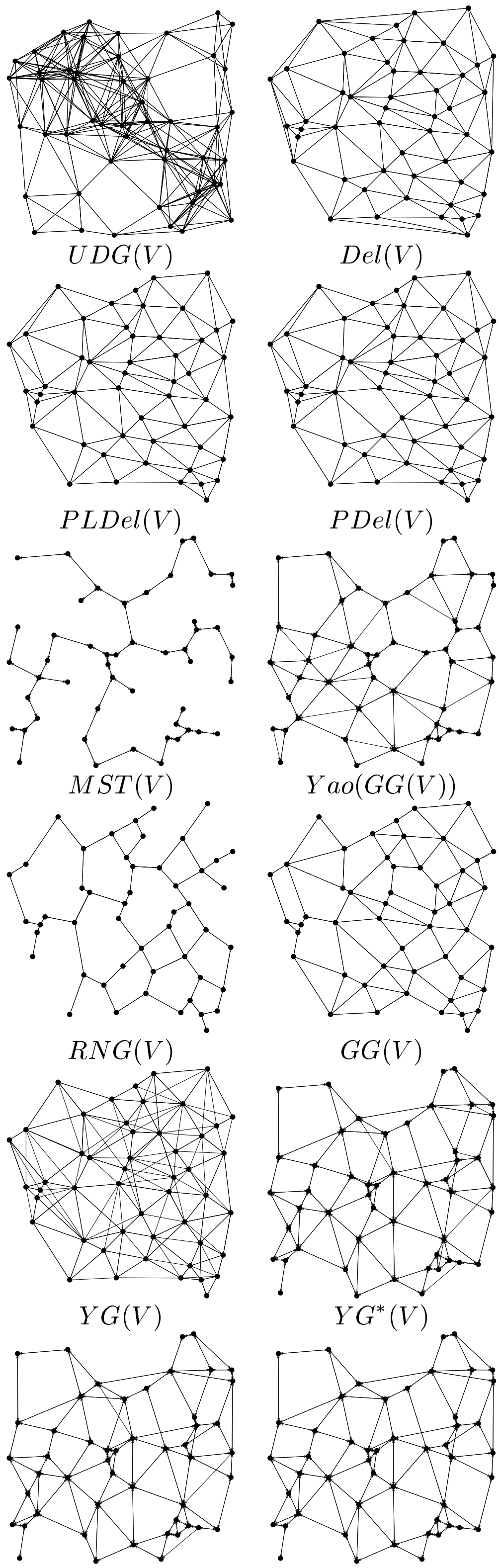

Figure 7: Different topologies from $UDG(V)$.

signment 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.* [59] 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 [60, 61]. Let EMST(V) be the Euclidean minimum spanning tree over a point set V. Both [60] and [61] use the power assignment induced by EMST(V). The correctness of using minimum spanning tree is proved in [60]. 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 [60] and [61] 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, Li [62] gave a simple $O(n \log n)$ time complexity centralized algorithm and also show how this algorithm can 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 [60] that the longest edge of the Euclidean minimum spanning tree EMST(V) is always the critical link.

The best distributed algorithm [63, 64, 65] 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 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.* [66] and by Clementi *et al.* [67, 68, 69]. Kiroustis *et al.* [66] 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 [66]. The algorithm guarantees the same approximation ratio in any dimensions. Then Clementi *et al.* [67, 68, 69] 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.* [70] gave a method that achieves better approximation ratio than the approach by the minimum spanning tree by using idea from the minimum Steiner tree.

5 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.

5.1 Centralized Methods

Guha and Khuller [71] 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. They [71] proved that this approach has approximation ratio $2(H(\Delta)+1)$.

The other approach is first approximating the dominating set and then connecting the dominating set to a connected dominating set. 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 [71] 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 [72].

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.* [73] and Marathe *et al.* [74] 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 [75], 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.* [76] gave an $\frac{4}{3}$ approximation method to connect a dominating set and Robins *et al.* [72] 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 [77]. Recently, Cheng *et al.* [78] designed a PTAS for MCDS in UDG. However, it is difficult to distributize their method efficiently.

5.2 Distributed Methods

Many distributed clustering (or dominating set) algorithms have been proposed in the literature [9, 79, 80, 81, 24, 82]. 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.

5.2.1 Clustering without Geometry Property

For general graphs, Jia *et al.* [83] 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, 84] 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 [85] 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, 84] 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, 84] finds a dominating set and then grows it to a connecting dominating set, the algorithm proposed by Wu and Li [86, 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 [80, 79], the ordered pair of degree and ID [87], 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 [88, 89, 87], several researchers 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.

5.2.2 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. 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.

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. It is also well-known that, $VirtG$ 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 by Wang and Li [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):

- IamDominator(u): node u tells its 1-hop neighbors that u is a dominator;
- IamDominatee(u, v): node u tells its 1-hop neighbors that u is a dominatee of node v;
- 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.
- 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.

Notice that the message IamDominator(u) is only broadcasted at most once by each node; the message IamDominatee(u, v) is only broadcasted at most five times by each node u for all possible dominators v; 2HopsPath(u, w, v) and 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:

- 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 Dominators.
- 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.
- 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 Dominators is at most five. Then 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 Connector2HopsPath, Connector3HopsPath are bounded by ℓ_2 and ℓ_3 respectively. Then we are in the position to review the distributed algorithm proposed by Wang and Li [11] to find the connectors efficiently. Assume that a maximal independent set is already constructed by a cluster algorithm.

Algorithm 3 *Finding Connectors*

1. Every dominatee node w broadcasts to its 1-hop neighbors a message IamDominatee(w, v) for each dominator v stored at Dominators.
2. Assume node u receives a message IamDominatee(w, v) for the first time. If $u \neq v$, v is not in Dominators list of u, and there is no pair $(*, v)$ in Connector2HopsPath, then u adds (w, v) to Connector2HopsPath. Here $*$ denotes any node ID. If u is a dominatee, then it broadcasts message a 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 IamDominatee$(*, v)$ afterward.
3. When a node w (it must be a dominatee here) receives the message 2HopsPath(u, w, v), node w marks itself as a *connector*, if u is a dominator.
4. Assume a dominator x receives the message 2HopsPath(u, w, v), where $x \neq w$. If there is no triple $(*, *, v)$ in Connector3HopsPath, then x adds (u, w, v) to Connector3HopsPath and broadcasts the message 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 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 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'.

Wang and Li [11] proved that 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.

5.2.3 The Properties of Backbone

It was shown in [11] 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. 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.* [52] proposed a method to make the graph CDS planar without losing the spanner property of the backbone. Their method applies the localized Delaunay triangulation [55] on top of the induced graph from CDS, denoted by ICDS. It was proved in [55] 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. In addition, $LDel(ICDS)$ has constant bounded hop-stretch factor and a bounded node degree [52].

6 Broadcasting & Multicasting

Minimum-energy broadcast/multicast routing in a simple ad hoc networking environment has been addressed by the pioneering work in [90, 91, 92, 93]. 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 [93], 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 [93], there are sufficient bandwidth and transceiver resources. Under these assumptions, centralized (as opposed to distributed) algorithms were presented by [93] 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 [93] 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 [93], 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 [94]. This greedy heuristic is similar to the Chvatal's algorithm [95] 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 [95], 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 [95], 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.* [94, 96] 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 [97], 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 [98], 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.* [90] proved that the minimum-energy broadcast routing problem is a NP-hard problem and obtained a parallel but weaker result to those of [94, 96].

Wan *et al.* [94, 96] gave some lower bounds on the approximation ratios of MST and BIP by studying some special instances in [94, 96]. 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 [94, 96] 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. They [94, 96] proved that c is at most 12. The proof of this theorem involves complicated geometric arguments.

Note that for any point set P of radius one, the length of each edge in $mst(P)$ is at most one. Therefore, 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 [94, 96] 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 1 *[94, 96] 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 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. This observation implies that the approximation ratio of the link-based MST heuristic is at most c, and therefore is at most 12 [94, 96]. In addition, they derived an upper bound on the approximation ratio of the BIP heuristic: 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 $\sum_{e\in mst(P)}\|e\|^{\beta}$. Once again, the Euclidean MST plays an important role.

6.2 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.* [99] 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 [100] proposed an efficient approach to collect $N_2(u)$ using the connected dominating set [10, 52] 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 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 broadcasting, in wireless ad hoc networks. Nevertheless, there are still many excellent results, such as localized routing methods, connectivity and capacity results of wireless networks, and location sevices, are not covered in this survey due to space limit.

8 Acknowledgment

The author would like to thank Ivan Stojmenovic for some helpful suggestions in preparing 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, To appear.

[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 *Proceedings of 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, New Orleans, LA*, 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] 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.

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

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

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

[30] 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.

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

[32] 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.

[33] Xiang-Yang Li, "Localized delaunay triangulation is as good as unit disk graph," *Journal of Computer Networks*, 2002, Submitted for publication.

[34] 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, 2001.

[35] I. Stojmenovic and X. Lin, "Power-aware localized routing in wireless networks," in *IEEE Int. Parallel and Distributed Processing Symp.*, 2000.

[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] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

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

[46] 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.

[47] 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.

[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] 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.

[56] 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.

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

[58] 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.

[59] 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.

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

[61] 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.

[62] Xiang-Yang Li, "Efficient power control in wireless ad hoc networks," 2002, Manuscript.

[63] 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.

[64] 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.

[65] J. A. Garay, S. Kutten, and D. Peleg, "A sublinear time distributed algorithms for minimum-weight spanning trees," in *Symp. on Theory of Computing*, 1993, pp. 659–668.

[66] 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.

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

[68] 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.

[69] 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.

[70] Gruia Călinescu, I. Mandoiu, and A. Zelikovsky, "Symmetric connectivity with minimum power consumption in radio networks," in *IFIP-TCS*, 2002, To appear.

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

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

[73] 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.

[74] 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.

[75] 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.

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

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

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

[79] 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.

[80] 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.

[81] 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.

[82] 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.

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

[84] 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.

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

[86] 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.

[87] 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.

[88] 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.

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

[90] 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.

[91] 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.

[92] 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.

[93] 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.

[94] 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.

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

[96] 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, To appear. Preliminary version appeared in IEEE INFOCOM 2000.

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

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

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

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