Partial Delaunay Triangulation and Degree Limited Localized Bluetooth Scatternet Formation

Xiang-Yang Li, *Member*, *IEEE*, Ivan Stojmenovic, and Yu Wang, *Member*, *IEEE*

Abstract—This paper addresses the problem of localized scatternet formation for multihop Bluetooth-based personal area ad hoc networks. Nodes are assumed to know their positions and are able to establish connections with any of their neighboring nodes, located within their transmission radius, in the neighbor discovery phase. The next phase of the proposed formation algorithm is optional and can be applied to construct a sparse geometric structure in a localized manner. We propose here a new sparse planar structure, namely, partial Delaunay triangulation (PDT), which can be constructed locally and is denser than other known localized planar structures. In the next mandatory phase, the degree of each node is limited to seven by applying the Yao structure, and the master-slave relations in piconets are formed in created subgraphs. This phase consists of several iterations. In each iteration, undecided nodes with higher keys than any of their undecided neighbors apply the Yao structure to bound the degrees, decide master-slave relations on the remaining edges, and inform all neighbors about either deleting edges or master-slave decisions. To the best of our knowledge, our schemes are the first schemes that construct degree limited (a node has at most seven slaves) and connected piconets in multihop networks, without parking any node. The creation and maintenance require small overhead in addition to maintaining accurate location information for one-hop neighbors. The experiments confirm good functionality of created Bluetooth networks in addition to their fast creation and straightforward maintenance.

Index Terms—Bluetooth, scatternet formation, Delaunay triangulation.

1 Introduction

The rapid adoption of the Internet and mobile wireless technologies is paving the way for high bandwidth to the mobile terminal. Local and personal area networks are also increasingly becoming wireless, incorporated into seamless all IP wireless and mobile networks. Ad hoc enabled consumer products will begin to form small-scale ad hoc networks between a small group of people/devices. Each device (called node hereafter) in an ad hoc network has a transmission radius (assumed normally to be the same for each device) for communicating with neighbors. In single-hop ad hoc networks, each node is within transmission range of any other node. In more general multihop ad hoc networks, some pairs of nodes cannot directly communicate with each other and routes between them are passing by intermediate nodes. In this article, we assume multihop networks. Ad hoc networking in such small networks should offer user friendly and secure network establishment that enable various services. One important service is, of course, to provide Internet access by interworking ad hoc networks with already existing infrastructures. Bluetooth [1], [18] is well-suited medium access protocol that provides ad hoc networking for the consumer market. It has, however, some technical challenges, such as scheduling, network creation, and routing. User mobility poses additional challenges for connection rerouting and QoS services. This paper deals with the problem of creating ad hoc networks using Bluetooth technology.

Bluetooth is an open specification for short-range wireless communication and networking, mainly intended to be a cable replacement between portable and/or fixed electronic devices. According to the standard [1], when two Bluetooth devices come into each other's communication range, one of them assumes the role of *master* of the communication and the other becomes the *slave*. This simple one hop network is called a *piconet*, and may include more slaves. The network topology resulting from the connection of piconets is called a *scatternet*. There is no limit on the maximum number of slaves connected to one master, although the number of *active* slaves at one time cannot exceed seven. If a master node has more than seven slaves, some slaves must be parked. To communicate with a parked slave, a master has to *unpark* it, thus possibly parking another active slave instead. The standard also allows multiple roles for the same device. A node can be master in one piconet and a slave in one or more other piconets. However, one node can be active only in one piconet. To operate as a member of another piconet, a node has to switch to the hopping frequency sequence of the other piconet. Since each switch causes a delay (e.g., scheduling and synchronization time), an efficient scatternet formation protocol can be one that minimizes the roles assigned to the nodes, without losing network connectivity.

The problem of scatternet formation is one of the key challenges introduced recently. Several criteria could be set as the objectives in forming scatternets. The resulting network should be connected, the number of piconets (i.e., the number of nodes with master role) should be minimized

- *X.-Y. Li and Y. Wang are with the Department of Computer Science, Illinois Institute of Technology, 10 W. 31st Street, Chicago, IL 60616. E-mail: xli@cs.iit.edu and wangyu1@iit.edu.*
- *I. Stojmenovic is with SITE, University of Ottowa, Ottowa, Ontario K1N 6N5, Canada. E-mail: ivan@site.uottawa.ca.*

Manuscript received 6 Jan. 2003; revised 17 June 2003; accepted 5 Sept. 2003. For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number 118111.

to provide faster routing, and the formation and maintenance of a scatternet should have small communication overhead. The communication overhead is reduced by applying a localized approach, where each node makes decisions based on local knowledge (preferably only about its direct neighbors). Finally, the protocol should create degree limited scatternets to avoid parking any node.

Our solutions are based on the assumption that each node knows the absolute or relative positions of itself and each of its neighbors. This assumption currently poses challenging technological tasks for short range Bluetooth devices, aimed primarily at home and office environments. However, a broad variety of location dependent services will become feasible in the near future. Although the commercial Global Position System (GPS) has an accuracy of around 10 meters, modern systems have accuracy up to three meters [19]. Indoor location systems are based on the proximity of fixed objects with known coordinates (e.g., sensors), measuring the angle of arrival and time delays of signals. The active badge system, for example, has accuracy within 9 cm of their true position [19], with work in progress to improve accuracy. If no indoor or outdoor location service is available, the distance between neighboring nodes can be estimated on the basis of incoming signal strengths or time delays. Relative coordinates of neighboring nodes can be obtained by exchanging such information between neighbors [12]. In this paper, we make use of location information of every node in the network, which is communicated and updated with neighboring nodes only. In our solutions, each node learns its own coordinates and communicates them to its neighbors. In such an approach, location imprecision has no impact on the performance as long as the errors are small with respect to the transmission radius.

For simplicity, we assume a free-space radio propagation model, which ensures that nodes within a certain distance will always be within radio range. Wireless networks are then often modeled by unit disk graphs, where two nodes are connected if and only if the distance between them is at most the transmission radius, which is equal for all nodes. In accordance to almost all research articles on wireless networks, we adopt the unit disk graph model here.

Bluetooth is a promising new wireless technology, which enables portable devices to form short-range wireless ad hoc networks based on a frequency hopping physical layer. Devices are not able to communicate unless they have previously discovered each other by synchronizing their frequency hopping patterns. Thus, even if all nodes are within direct communication range of each other, only those nodes that are synchronized with the transmitter can hear the transmission. Synchronizing the frequency hopping patterns is apparently a time consuming and pseudorandom process [28]. In this paper, we assume that the problem of discovering all neighbors within transmission radius of all neighbors is resolved by a separate Bluetooth protocol. One such protocol for discovering all one hop networks is described in [8], [28], while a protocol that provides two-hop information to every node is described in [25]. These protocols are applicable as the first phase of our scheme.

Our proposed Bluetooth formation algorithms include three phases. The first phase is the neighbor discovery phase. The second phase is optional, and can be applied to construct a sparse localized geometric structure, such as a Gabriel graph (GG), relative neighborhood graph (RNG), or Yao graph. We also propose here a new sparse planar structure, namely, partial Delaunay triangulation (PDT), which can be constructed locally and is denser than other known localized planar structures. In the third phase, we apply a Yao structure on unit disk graph or the generated structure, and assign the master-slave relations in piconets. We prove that applying a Yao structure limits the degree of each node to seven and leaves the graph connected (and planar if the selected structure was planar). We describe two methods for assigning roles to nodes: setting the higher degree node of an edge as master, and a dominating set scheme based on clustering and adding two-element gateway piconets. The third phase consists of several iterations, and applying Yao structure and assigning roles are combined together in each iteration. We therefore obtain the first Bluetooth scatternet formation algorithm for multihop network which limits the degree of each node to seven, keeps the connectivity of all the piconets, and does not park any node.

The rest of the paper is organized as follows: In Section 2, we give preliminaries needed to describe our new algorithms, and briefly review the literature on scatternet formation and related network topology design issues. We then describe the new planar localized sparse graph, called partial Delaunay triangulation (PDT), in Section 3. Section 4 presents our new Bluetooth formation algorithms in detail. Section 5 describes the experimental data on some performance measures of our scatternet formation schemes. We conclude our paper in Section 6 by pointing out some possible future research directions.

A preliminary conference version of this article appeared in [21]. This version contains an improved algorithm and experimental data contributed by the added third author, and better overall presentation.

2 Preliminaries

In this section, we first give some geometry definitions and notations that will be used in our presentation later. We then briefly review some related results in constructing network topologies for wireless ad hoc networks.

2.1 Definitions and Notations

We assume that all wireless nodes are given as a set S of n vertices in a two-dimensional space. Each node has some computational power. These nodes induce a *unit disk graph* $UDG(S)$ in which there is an edge between two nodes if and only if their distance is at most one. Hereafter, we always assume that $UDG(S)$ is a connected graph. Let k-*local nodes* of u be all nodes within a constant k hops of a node u in the unit disk graph $UDG(S)$. We denote by $N_k(u)$ the set of all such nodes. Usually, the constant k is 1 or 2, which will be omitted if it is clear from the context. Various proximity subgraphs of the unit disk graph can be defined [20] as outlined in the following paragraphs.

Let $disk(u, v)$ be the disk with edge uv as a diameter, and let $|uv|$ be the Euclidian physical distance between nodes u and v. Then, the *Gabriel graph* [14] $GG(S)$ contains edge uv if and only if $disk(u, v)$ contains no other points of S. See Fig. 1b. $GG(S)$ is a planar graph (that is, no two edges cross each other). It was proven in [10] that the intersection of a

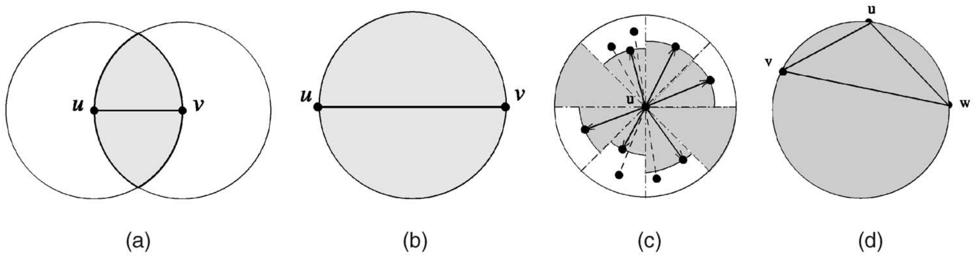


Fig. 1. The definitions of (a) *RNG*, (b) *GG*, (c) *Yao*, and (d) *Del*. The shaded area is empty of nodes inside.

connected unit disk graph $UDG(S)$ and Gabriel graph $GG(S)$ is a connected planar graph. In this paper, we only consider the intersection $UDG(S) \cap GG(S)$, which is called the *constrained Gabriel graph*. We also denote it by $GG(S)$ hereafter, since we only discuss unit disk graphs in this article. $GG(S)$ can be constructed in a localized manner. In other words, a node u can compute its incident edges in $GG(S)$ by only using information $N_1(u)$ of 1-hop neighbors. One simple method is as follows: Each node u can test whether an edge uv belongs to $GG(S)$ in $O(d)$ computation time, by verifying whether $|wp| > |uv|/2$ is satisfied for each of its neighbors w. Here, d is the cardinality of $N_1(u)$, and p is the midpoint of uv. Thus, node u can compute all its incident Gabriel edges in time $O(d^2)$.

The *relative neighborhood graph*, denoted by $RNG(S)$, is a geometric and graph theoretic concept proposed by Toussaint [31]. It consists of all edges uv such that the intersection of two circles centered at u and v and with radius $|uv|$ do not contain any vertex w from the set S. See Fig. 1a. It is easy to show that $RNG(S)$ is a subgraph of $GG(S)$. Both $GG(S)$ and $RNG(S)$ are connected and contain the Euclidean minimum spanning tree of S. The intersection of $RNG(S)$ with the unit disk graph $UDG(S)$ is a connected graph if $UDG(S)$ is connected. We call the intersection $UDG(S) \cap RNG(S)$ the *constrained relative neighborhood graph*, which is also denoted by $RNG(S)$ hereafter. Note that RNG and GG (and PDT introduced later in this paper) require no message exchange between nodes, in addition to maintaining unit disk graph.

The Yao graph [35] is proposed by Yao to construct a *minimum spanning tree* (MST) of a set of points in high dimensions efficiently. At given node u, any k equally separated rays originated at u define k cones. In each cone, choose the closest node v within the transmission range of u, if there is any, and add a directed link $\overrightarrow{uv}$. Ties are broken arbitrarily. The remaining edges are deleted from the graph. See Fig. 1c. There are several variants on how this construction can be carried at each node in the graph. One choice is to carry it simultaneously on each node, with two options about keeping an edge uv: keep only if they mutually selected each other, or keep directional edges as well (one node selected other but not vice versa). Another option considered in this paper is to carry this process first at node u, and then at node v. In this case, if u did not select v, then edge uv is considered deleted by v and is ignored when v makes its decision afterward.

We continue with the definition of the Delaunay triangulation. We assume that there are no four vertices of S that are cocircular. A triangulation of S is a *Delaunay triangulation*, denoted by $Del(S)$, if the circumcircle of each of its triangles does not contain any other vertices of S in its interior. A triangle is called the *Delaunay triangle* if its circumcircle is empty of vertices of S. See Fig. 1d. A well-known criterion that will be used in this paper is that an edge uv belongs to $Del(S)$ if and only if there exists a circle, containing u and v on its boundary, which does not contain any other point from S in its interior. Gabriel graph $GG(S)$ and the relative neighborhood graph $RNG(S)$ are subgraphs of the Delaunay triangulation $Del(S)$. Other geometric structures in wireless networks, and appropriate references, are surveyed in [20].

A subset of vertices in a graph G is a *dominating set* if all the vertices in G are either in this subset or neighbors of vertices in this subset. An example of a dominating set, which will be used in this paper, is the set of clusterhead nodes obtained in clustering scheme [22]. Nodes which are neighbors to two clusterheads are called gateway nodes. To preserve connectivity of clusters, any two clusterheads at distance three identify a pair of neighboring nodes from each cluster that are connected. A construction of a minimal number of such pairs of gateway nodes is described in [33].

2.2 Literature Review on Bluetooth Scatternet Formation

Although describing methods for device discovery and for the participation of a node in multiple piconets, the Bluetooth specification does not indicate any method for scatternet formation. The solutions proposed in literature can be divided into single-hop (see [31] for a review, which is not given here due to space constraints), and multihop ([35], [5], [32], [15], [16], [3], [7], [24]) solutions. In a single-hop topology, all devices are in the radio vicinity of each other, which is not always the case in realistic scenarios. Our schemes are designed for multihop scenarios, but are clearly applicable to single-hop networks as well.

Zaruba et al. [35] proposed two protocols for forming connected scatternet. In both cases, the resulting topology is termed a *bluetree*. The number of roles each node can assume is limited to two or three. The first protocol is initiated by a single node, called the *blueroot*, which will be the root of the bluetree. A rooted spanning tree is built as follows: The root will be assigned the role of master. Every one hop neighbor of the root will be its slave. The children of the root will now be assigned an additional master role and all their neighbors that are not assigned any roles yet will become slaves of these newly created masters. This procedure is repeated recursively till all nodes are assigned. Each node is slave for only one master, the one that *paged* it first. Each internal node of the tree is a master on one piconet, and slave of another master (its parent in the initial tree). In order to limit the number of slaves, they [35] observed that if a node in the unit disk graph has more than five neighbors, then at least two of them must be connected. This observation is used to reconfigure the tree so that each master node has no more than five slaves. If a master node has more than five slaves, it selects its two slaves s_1 and s_2 that are connected and instructs s_2 to be master of s_1, and then disconnects s_2 from itself. Such branch reorganization is carried throughout the network. In the second protocol [35], several roots are initially selected. Each of them then creates its own scatternet as in the first protocol. In the second phase, subtree scatternets are connected into one scatternet spanning the entire network.

A greedy centralized multihop algorithm where a hypothetical central entity knows the complete topology has been proposed in [5]. Distributed algorithms have also been proposed in [5], which assume 2-hop neighborhood information. This is achievable in Bluetooth since the identities of the neighboring nodes are known at the end of the device discovery procedure. The nodes are made to exchange this neighborhood information with each of its neighbors so that they have 2-hop information and a partial view of the underlying topology. The algorithm [5] applies a variant of clustering algorithm where the number of nodes in each cluster is limited to seven, in accordance to Bluetooth restriction. A node with highest degree among all its undecided neighbors will become a master node and will choose up to seven slaves among neighboring nodes, with priority given to lower degree nodes. However, there are examples where the scatternet is disconnected, which may occur when two clusterheads were originally connected, but formed clusters and "erased" their link without leaving alternate connection between their piconets. For example, assume that the graph contains two connected nodes u and v, each with its own seven more neighbors. Thus, u and v have degrees eight, and will become masters of two piconets, containing their own seven neighbors as slaves. However, the graph will then be disconnected since the link between u and v will disappear.

Basagni and Petrioli [7], [25] described multihop scatternet formation scheme based on clustering scheme [22], taking into account several Bluetooth issues which do not pertain to clustering. Clusterhead (master role) decisions are based on node weights (instead of node IDs, as used in [22]), that express their suitability to become masters, following a variant of the clustering method described in [6]. All clusterhead nodes are declared master nodes in a piconet, with all nodes belonging to their clusters as their slaves. Some of the slaves become masters of additional piconets (following, e.g., [34]), to assure connectivity. However, piconets may have more than seven slaves. This may result in performance degradation, as slaves need to be parked and unparked in order for them to communicate with their master. The topology discovery phase is performed before clustering in order to provide each node with information about all its neighbors. It is performed by each node randomly entering inquiry or inquiry scan mode (with equal probabilities), and randomly selecting the time length for being in the mode repeatedly until a timeout that should be carefully selected to enable one hop information with high probability, but within reasonable time. A performance evaluation of the clustering-based scatternet formation scheme [7], [25] is given in [4].

References [32] and [15] essentially propose variants of clustering-based scatternet formation schemes, where the clustering process does not use any ID to decide clusterheads, that is, master nodes. Instead, decisions are made at random. Already existing master nodes have priority in attracting more slaves, up to the limit. Initial connections are made by nodes entering scan or inquiry scan phases at random. After each node is assigned master or slave role, or is unable to join any piconet or attract any neighbor as its slave to create its own piconet, bridge piconets are added to connect the scatternet. However, the process does not always lead to a connected structure. The counterexample is the same that applies to [5]. On a positive side, [32] proposes two excellent measures for the performance of scatternets: average shortest-path length and maximum traffic flow.

Guerin et al. [16] proposed DFS, BFS, and MST-based scatternet formation schemes for unit graphs in two and three dimensions. They construct a tree where all nodes at one level are either masters or slaves (i.e., they construct bipartite graphs). Their construction does not guarantee maximum degree bound unless the structure itself provides the bound. For example, MST in two dimensions has a maximum degree of five, but in three dimensions, some nodes can have degrees up to 13. The schemes are not localized.

Recently, Ajmone-Marsan et al. [3] described a centralized solution for finding Bluetooth topology that provides full network connectivity, fulfills the traffic requirements and the constraints posed by the system specification, and minimizes the traffic load of the most congested node in the network.

Our schemes, originally proposed and made available in June 2001 (e.g., cited in [24]), are the first schemes that construct degree limited and connected piconets in multihop networks, without parking any node. Recently, another such scheme has been described by Petrioli and Basagni [24]. Their neat scheme does not require position information, but instead the local information is extended to two-hop information, with a two round device discovery phase for obtaining necessary information. It is a modified clustering process, where selection of slaves is performed in such a way that, if a master has more than seven neighbors, it chooses up to seve slaves among them so that it can reach all the others via them. Such a coverage is always possible with up to five slaves [35]. Scatternet formation proceeds in iterations. Nodes that participate in a given iteration perform modified clustering process. Initially all nodes are undecided. In each iteration, init-nodes (nodes having bigger weight among its immediate undecided neighbors) create piconets, choosing at most seven neighbors as slaves, and deleting remaining edges. The iteration stops when all nodes are decided, and all created masters, together with slaves that are not selected for links with slaves from other piconets, withdraw from the next iteration. Simulations by the authors show that created scatternets have a low average number of roles per node (about two), with average path length increase between 20 percent and 80 percent. The method may show weaknesses on some other metrics, especially about the worst-case number of slave roles a node can assume. For instance, in case of dense networks (e.g., complete graph), the second *biggest* node in a neighborhood may end up serving as slave to all the masters in the same neighborhood. The methods presented in this paper provide the limit on the number of slave roles for each node, and planarity which is important for routing performance. However, this is achieved by using a stronger assumption, position information. Without position information, the method [24] is currently the best available method.

Recently, Basagni et al. [9] described the results of an ns2-based comparative performance evaluation among three major solutions for forming multihop scatternet: [21], [25], [35]. They found that device discovery is the most time-consuming operation, independently of the particular protocol to which it is applied. The comparative performance evaluation showed that due to the simplicity of its operations BlueStars [26] is by far the fastest protocol for scatternet formation. However, BlueStars produces scatternets with an unbounded, possibly large number of slaves per piconet, which imposes the use of potentially inefficient Bluetooth operations. They proposed a combined solution by applying a Yao structure as described here on each piconet, to limit the degree of each master node to seven. This is a variant of the clustering-based scheme presented in this article, with degree limitation applied at the end instead of during the scatternet creation process.

3 Partial Delaunay Triangulation

We shall now propose a planar geometric structure which can be used in the first step of our algorithm. The motivation for the introduction of a new planar locally defined sparse graph is to improve the graph connectivity of planar graphs, which in turn will improve the performance of routing algorithms, and to define scatternets with improved density that preserve planarity (Yao graph construction does not preserve planarity). Notice that the Delaunay triangulation is a planar graph and it contains the Gabriel graph, the relative neighborhood graph, and the Euclidean minimum spanning tree as subgraphs. However, the Delaunay triangulation $Del(S)$ of a set of wireless nodes S cannot be constructed in a localized manner. In this section, we will propose a geometric structure, namely the partial Delaunay triangulation (PDT), that can be constructed in a localized manner. Partial Delaunay triangulation contains the Gabriel graph as its subgraph, and itself is a subgraph of the Delaunay triangulation. The algorithm for the construction of PDT can be described as follows.

Let u and v be two neighboring nodes in the network. Edge uv belongs to $Del(S)$ 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 S. 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 segment 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 $Del(S)$.

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 $Del(S)$. Edge uv belongs to $Del(S)$ if $\angle uwv + \angle uxv < \pi$ (here, we assume that no four points of S are cocircular). 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 lemma.

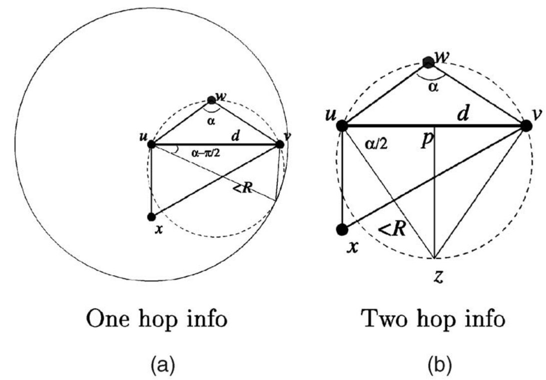


Fig. 2. Scenarios for deciding a Delaunay edge.

Lemma 1. *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 and $\alpha = \angle uwv$ (here, $\alpha \geq \frac{\pi}{2}$).*

Proof. Consider any edge uv. See Fig. 2a for an illustration. It is added to the PDT if the circumcircle passing through u, v, and w is contained inside the transmission region of u and this circumcircle does not contain any nodes of $N_1(u)$ inside. Then, this circumcircle passing through u, v, and w is guaranteed to be empty of nodes from S. Thus, edge uv is a Delaunay edge. Let r be radius of the circumcircle. Then, $\sin\alpha = cos(\alpha - \pi/2) = \frac{d}{2r}$ and the lemma follows from $2r = \frac{d}{\sin\alpha} < R$. □

Lemma 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 $\alpha = \angle uwv$.*

Proof. Consider any edge uv. See Fig. 2b for an illustration. Here, pz is the perpendicular bisector of edge uv. Edge uv is added to the PDT if the circumcircle passing through u, v, and w is contained inside the union of the transmission regions of u and v and this circumcircle does not contain any nodes of $N_1(u) \cup N_1(v)$ inside. This is equivalent to $uz < R$, where $uz = \frac{d}{2\cos(\alpha/2)}$. Then, this circumcircle passing through u, v, and w is guaranteed to be empty of nodes from S. Thus, edge uv is a Delaunay edge. □

Note that an edge uv might belong to $Del(S)$ here, but it cannot be determined from the local knowledge. If two hop neighborhood information is available, or u and v communicate their best choices, then the decision procedure is the same as by Lemma 2.

TABLE 1
Scatternet Formation Algorithms

1. Neighbor discovery and information exchange (collecting the node degree information).
2. Planar subgraph construction (constructing RNG, GG, or PDT), if desirable.
3. Bounding Degree and Assigning Roles (consisting of several iterations).
   3.1 Initially all nodes are undecided.
   3.2 In each iteration, if a undecided node u has the highest degree among its all undecided neighbors, it runs the following steps:
      a) Bound its node degree by 7 (applying Yao structure).
      b) Assign role to itself (based on the information on each link or using cluster based method).
      c) Mark itself decided, and notice the deleted edges and its status to its undecided neighbors.
   3.3 Repeat the iterations, until all nodes are decided.

4 New Scatternet Formation Algorithms

We now proceed to describe our localized scatternet formation algorithms based on sparse geometrical structures. The algorithms have several common phases which are shown in Table 1.

4.1 Neighbor Discovery and Information Exchange

Initially, in the neighbor discovery phase, each node learns about its one-hop or two-hop neighbors. This procedure is also called an *inquiry* procedure in Bluetooth specifications. Master-slave relations are decided based on a key. Several different keys can be considered. If a node's Bluetooth ID is used as a key, one-hop information suffices in our protocols. Such neighbor discovery (inquiry procedure) can be performed by a scheme described in [7], [8], [28]. It is performed by each node randomly entering *inquiry* or *inquiry scan* mode (with equal probabilities, or alternating between the two modes), and randomly selecting the length of each *inquiry/inquiry scan* cycle repeatedly until a timeout. Two neighboring nodes must be in complementary states in order to communicate and exchange information. The only modification needed for our application is that nodes exchange their position in addition to their IDs, which is a trivial addition to the packet content. If the key is selected as the record (degree, ID), where node degree is primary key, and ID is secondary key, one-hop neighbor discovery is not sufficient to exchange correct information about the number of neighbors of each neighbor. The procedure needed to collect degree information from neighbors is basically the same procedure needed to collect two-hop information (neighbors for each neighbor), the only difference again being the packet content. One such Bluetooth compatible procedure for collecting two-hop information has been described in [25] and is applicable here. It has some lengthy but straightforward details that we will not describe here.

In order to resolve the ties conveniently for proper application of Yao structures, all edges need to have mutually different lengths. To achieve this, we modify the definition of length. Each edge uv receives new label $\|uv\| = (|uv|, key(u), key(v))$, where $|uv|$ is the distance between u and v. Thus, two edges are compared by their length first. If their lengths are the same, then they are compared by their key values. Since key pairs of edges are unique, no two edges have equal labels (thus, there are no ties when the Yao structure is defined).

4.2 Planar Subgraph Construction

This phase is optional. The remaining phases can be applied on the unit disk graph directly, but will result in nonplanar scatternets. Planarity may be a desirable property in some applications, such as routing with guaranteed delivery (GG, PDT), or broadcasting (RNG). In this phase, each node computes which of its incident edges belongs to the chosen planar sparse structure, RNG, GG, or PDT. Note that each node can make local decisions about each of its edges without any message being exchanged with any of its neighbors (after completing neighbor discovery phase). Thus, this construction has basically no cost involved, since communication cost is always significantly higher than the computation cost. In fact, the construction of planar structure at this stage actually reduces the cost of subsequent phases since they are applied on remaining edges only, and the amount of information exchange is therefore reduced.

4.3 Bounding Degree and Assigning Roles

In this (mandatory) phase, the degree of each node is limited to seven by applying the Yao structure, and the master-slave relations are formed in created subgraphs. Each node creates a key which will be used for comparison with neighboring nodes when assigning roles. We consider two possibilities for the key selection. A node's Bluetooth ID can be used as the key, and such a choice requires one-hop neighbor discovery in the first phase. The other choice for the key is the record $(degree, ID)$, where $degree$ is the node's degree after the first neighbor discovery phase. To collect degree information from neighbors, two-hop neighbor discovery phase needs to be done in the first phase. However, the number of piconets will be reduced, thus scatternet is expected to function better and this choice is considered in our experiments. We will therefore only refer to this choice in the sequel.

In this phase, each node applies the Yao structure on all of its neighbors, where $k = 7$. This will guarantee that the number of slaves assigned to any node is no more than seven. To simplify the explanation, we assume that Yao construction is applied to all nodes (each at appropriate iteration), even if the number of its neighbors is no more than seven (although it is not necessary in such case). An edge remains in the structure if and only if both endpoints selected it in their respective applications of the Yao construct, otherwise it is deleted from the structure. There are two approaches to apply the Yao structure here. The *iterative* approach, described here in detail, is to divide the process into iterations, and apply the Yao structure to several nodes in each iteration. In the *simultaneous* approach, described in detail in subsequent work [29], the Yao structure is applied to all nodes with excess degree simultaneously. The iterative approach is performed to create an *undirected* graph such that the maximum node degree is at most seven. It works as follows.

Initially, all nodes are undecided. In each iteration, undecided nodes with higher keys than any of their undecided neighbors (we shall refer to such nodes as *active nodes* in the sequel) apply the Yao structure to limit the degree, decide master-slave relations on the remaining edges, and inform all neighbors about either deleting edge or master-slave decision. The next section will describe how to assign master-slave relations. The active node then switches to a decided state. Assume that an active node u is a node that applies the Yao construction. Then, node u divides the region surrounding it into seven equal angles centered at u, and chooses the closest node from each region, if there is any. All remaining connections at u are simply deleted from the graph. Note that the elimination of any such edge uv by u immediately reduces the degree of v, i.e., node v has to remove link uv also. However, in order to avoid excessive information exchange between neighbors, the originally decided keys (that is, original degrees) are used in all comparisons.

At the end of each iteration, an information exchange step is needed so that active nodes inform their neighbors in the applied structure about the decisions made following the application of the Yao structure. For eliminated edges, the other endpoint node is informed about the decision and that node then deletes that edge from its own list. For the selected edge, the active node makes the master-slave decision for the edge (as explained in the next section) and informs the other node on each edge about the decision. This information exchange step is very similar to the one-hop neighbor discovery phase, and can actually be performed by an almost identical protocol. The difference is that the active node, being in inquiry mode (acting as master node), needs only to contact each of its neighbors along remaining edges, instead of each of its original neighbors in the unit graph. The information being exchanged is, of course, different. Since communication can be restricted to edges remaining in the graph, the information exchange step is faster than neighbor discovery phase.

We shall prove that the graph remains connected after this phase.

Theorem 1. *The iterative application of the Yao structure preserves graph connectivity.*

Proof. It suffices to show that the resulting subgraph contains a minimum spanning tree. Let uv be a deleted edge. Assume that it is deleted by node u. Let w be the node selected by u from the same region as v. Thus, $\angle wuv < \frac{\pi}{3}$ because $k = 7$. Since $||uw|| < ||uv||$ (note that we modified the definition of length so that edge lengths from the common endpoint became unique), it follows that $||vw|| < ||uv||$ because edge vw cannot be the longest edge in triangle uvw. Suppose that the phase is completed, with some edges like uv being deleted. Consider now, a minimal spanning tree (MST) of the basic structure, PDT, GG, or RNG, before applying this step. The minimum spanning tree is constructed in the following way: Every edge uv is assigned its weight $||uv|| = (|uv|, key(u), key(v))$. Sort all edges uv of the unit graph, PDT, GG, or RNG according to these weights in the increasing order. The algorithm that constructs MST adds edges in the sorted order. Following the well-known Kruskal's scheme, an edge is added to the MST if its addition does not create a cycle together with previously added edges. Thus, shorter edges receive the chance to be added to MST before longer ones. Assume that an edge uv is deleted by u because of the existence of node w. We will show that there is path connecting u and v at the end. The fact that $||uw|| < ||uv||$ and $||vw|| < ||uv||$ implies that vw and uw are considered for adding to MST before uv. After edge vw is considered, nodes v and w are connected in MST, with or without adding vw. Similarly, after edge uw is considered, nodes u and w are connected in MST, with or without adding uw. Thus, before considering uv, all nodes u, v, and w will be connected in MST. Therefore, edge uv is not in MST. It implies that none of the eliminated edges is in the MST constructed as above. The constructed minimum spanning tree MST connects all the nodes from the originally connected PDT, RNG, or GG. Thus, the performed phase preserves connectivity. □

We have extracted a connected sparse subgraph such that each node has degree at most seven in a series of iterations. In addition, the constructed topology may be a planar graph, if we decide so, which makes it possible to implement some position-based routing algorithms [10]. At the end of each iteration, active nodes decide master-slave roles at each undeleted edge, and communicate the decision to the other node at each edge. We shall now describe two different ways to decide the roles: the node with the initially higher key is master, and cluster-based. Both methods keep all links "saved" by the Yao structure in the final Bluetooth topology, but converts them to directed edges so that one node on each edge is the master node and the other is the slave node.

The first method assigns roles based on the information on each link. Each node creates a key, either ID or $(degree, ID)$, where degree is the number of its neighbors in the topology constructed in the neighbor discovery phase. Two neighboring nodes u and v compare their keys, and the one with the higher key becomes the master node, and the other node is the slave node. The purpose of such

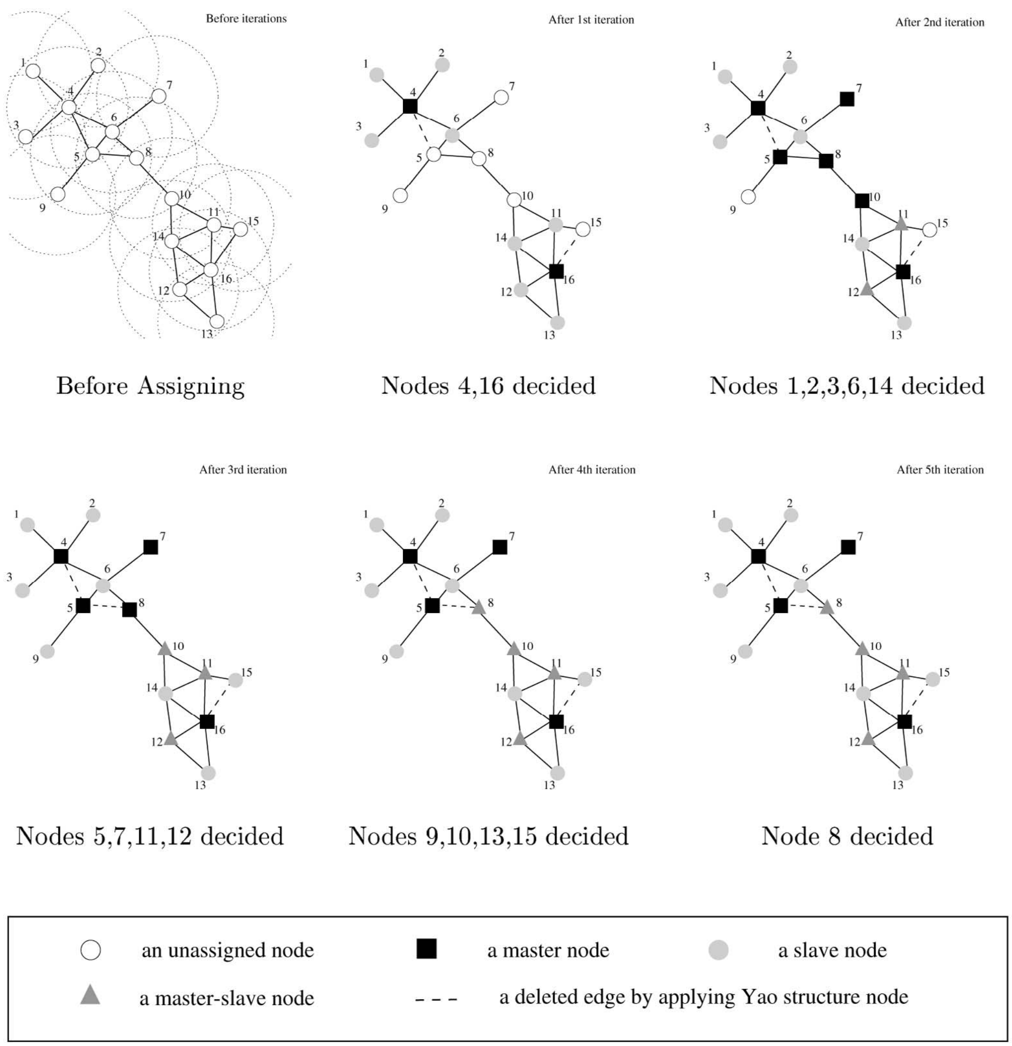


Fig. 3. An example of scenarios for assigning roles: five iterations.

role assignment is to avoid slave roles at high connectivity nodes. Let us refer to the algorithms that create scatternets using highest degree keys as $d*$, where $*$ is replaced by the name of the sparse topology from the second phase.

In the cluster-based approach, a dominating set of masters in the degree limited subgraph is constructed, and a piconet is added for each remaining edge between two nodes not selected in the dominating set, to preserve connectivity. Note that such gateway piconets may have more than two nodes in it. Notice also that the method presented in [2] constructs a connected dominating set using two rounds of construction: first a dominating set is constructed (as we construct piconets), then some gateway nodes are selected to connect the dominators (as we construct some gateway piconets) to preserve connectivity. However, the method proposed here will not distinguish these two rounds: nodes will perform both operations in a single round consisting of several iterations. In a given iteration, an active node could have received previously a master or slave or both roles from other nodes on edges that are preserved after applying the Yao structure at the node (see previous section). There are three cases for assigning role: 1) An active node decides to serve as the master node if it has only the master role or was unassigned. It notifies its undecided neighbors to add a slave role. Such a decision indicates that the node is creating a piconet. Notice that here, an active node could get the master role previously from one of its slave neighbors (described in the next case). 2) If an active node has previously received only slave roles, it decides to serve as a slave on all its remaining links. Thus, it notifies all remaining undecided neighboring nodes to add a master role. In other words, this active node decides to become a bridge to other piconets. 3) If an active node has previously been given both master and slave roles, it keeps these master-slave roles and notifies all its remaining undecided neighboring nodes to add a slave role on the link to that active node. It also indicates that the node is creating a piconet. Observe that we can also let it notify all its undecided neighboring nodes to add a master role instead. Such a resolution would correspond to the outcome of the clustering algorithm [22]. However, our experimental results show that it will select somewhat more nodes with master role. Notice that each active node marks itself decided after the above operation. Also each node, when receiving a notice of adding role, will change its role correspondingly. For example, if a slave node receives a notice of adding a master role, it will change its role to a master-slave node. Fig. 3 illustrates the detailed iterations of assigning roles for an example network. Let us refer to the algorithms that create scatternets with the cluster-based

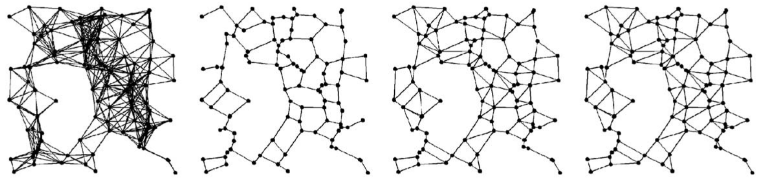


Fig. 4. Unit disk graphs and its planar subgraphs.

approach as $g*$, where $*$ is replaced by the name of the sparse topology from the second phase.

We then show that the scatternet formed by the above method is indeed connected.

Theorem 2. *The scatternet formed by the above method is connected.*

Proof. Recall that we have shown that the structure by bounding the node degree is a connected graph (see Theorem 1). Consider any two piconets centered at nodes u and v. Since the underlying structure is connected, there exists a path to connect u and v. For any link xy in the path, since every node will become active at some point, assume that x becomes active before y. From our role-assignment method, if at that time x is unassigned, master or master-slave, then it will form a scatternet to connect y; if at that time x is a slave, then y will be assigned a master role, so the link xy also exists in the scatternet. This concludes the proof. □

One problem in both proposed designs is that many master nodes in one piconet may serve as slave nodes in another. This slows down the operation of the piconet since the master node needs to switch occasionally to another piconet on a time division basis. Notice that not all links of the degree limited topology need to be kept in the final Bluetooth topology by our method. We can also apply the method from [2], [33] to minimize the number of added two node piconets and create a connected scatternet. Using geometric properties, it can be shown that the number of added two nodes piconets is at most a small constant factor of the piconets created based on the clusterhead method [33]. Note that clustering does not have localized maintenance property since a single node movement may trigger chain effect and global change in the structure. However, if a slightly different cluster update scheme is applied, the localized maintenance property can be maintained, at the cost of increasing the number of piconets.

5 Experiments

In this section, we present our experimental results that compare designed algorithms in terms of various characteristics. Each node can be 1) slave only, denoted by S, possibly to few piconets, this can be further divided as S_p, where p is the number of piconets where this slave node serves; 2) master only, denoted by M; 3) master of one piconet and slave in other piconets, denoted by MS or in general by MS_p.

Since the presented algorithms are the first algorithms that generate degree bounded (bounding both master and slave roles) and connected scatternet structures, and in addition provide planarity, there is no scheme that matches these qualitative characteristics. We therefore did not compare our schemes with other schemes on the selected quantitative metrics. In the experimental results presented here, we choose a total of $n = 100$ wireless nodes which are distributed randomly in a square area with side length $a = 50$ meters. Each node are specified by random X and Y coordinate values. The transmission radius of each wireless node is set to $r = 10$ meters. Notice that, to make UDG connected, the transmission radius should be above a minimum value. In [17], Gupta and Kumar showed that UDG is connected with very high probability if $r \geq \sqrt{\frac{\ln n}{n\pi}}$ when n goes to infinity. The graph density (average number of neighbors) of UDG can be calculated approximately by an area argument: $(n-1) \times \pi \times \frac{r^2}{a^2} \simeq 99 \times 3.14 \times \frac{10^2}{50^2} \simeq 12.4$. We randomly generate 20 node sets V, construct the $UDG(V)$ (only connected graphs are considered), then perform our localized scatternet formation algorithms to form the Bluetooth structures. All results are the averages on a total of 20 connected wireless node sets.

Fig. 5 illustrates the different Bluetooth structures using UDG, RNG, GG, or PDT as the topology (shown in Fig. 4), bounding degree by applying the Yao structure, and assigning node roles by comparing end-nodes degree of each link (denoted by d*) or using cluster-based method (denoted by g*). The master and master-slave nodes are denoted by squares and triangles, respectively, while the slaver nodes are denoted by disks.

Table 2 lists the number of slave nodes that serve as slaves of p piconets under different Bluetooth topologies. Table 3 lists the number of master-slave nodes that serve as slaves of p piconets under different Bluetooth topologies. Recall that S_p/MS_p is the number of slave/master-slave nodes that serve as slaves of p piconets. We conducted extensive simulations using different transmission ranges (from $10m$ to $100m$) and different number of nodes (from 20 to 500). We found that the results are stable, e.g., the portion of the master-slave nodes. In addition, as we expected, the cluster-based method generates smaller number of nodes with master-slave roles than the method comparing degrees of two end-points of a link.

Table 4 presents the average number of slave nodes assigned to a node with master role, that is, a master node or a master-slave node. The fifth column represents the average number of piconets assigned to a node with slave roles only. The sixth column represents the average number of piconets assigned to a node with both master and slave roles. We found that assigning node roles based on the cluster-based approach always assigns less number of slaves to a node with master role. Moreover, it also generates less number of nodes with master-slave role than the other method.

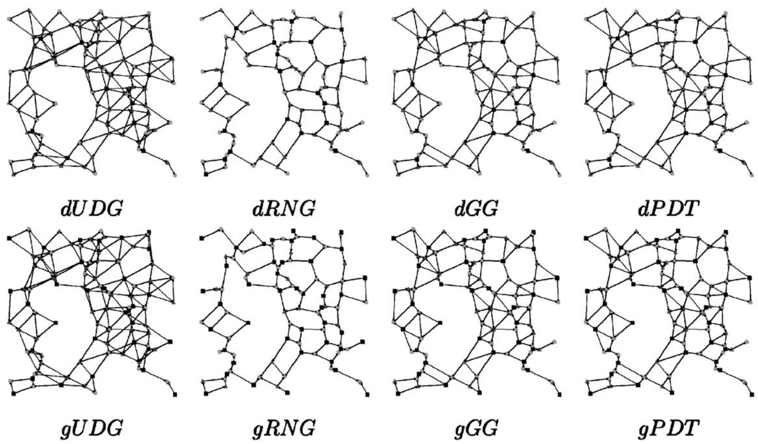


Fig. 5. Geometric structures, bounding node degrees, and assigning roles.

We found that the UDG consistently performs the worst among all underlying structures: it has less pure master nodes and has many slave nodes belonging to many piconets. The other structures (GG, RNG, PDT) perform at the same level in terms of the number of piconets generated and the number of piconets a slave node belongs to. We prefer to use PDT since it has more edges than the other two structures and, thus, can sustain more link failures and has somewhat shorter path on average. We found that scatternets generated based on GG and PDT are similar when node density is high, due to the fact that PDT has only very few more edges than GG.

Notice that our simulations consider only the performance of the geometrical structure of the resulting scatternets. Some other performance metrics for Bluetooth, such as the scatternet formation time and network capacity, are also very important and should be considered. Recently, Basagni et al. [9] conducted an ns2-based comparative performance evaluation of our method and other two major solutions [25], [35]. They showed the results for scatternet formation durations and found that device discovery is the most time-consuming operation. Unfortunately, all scatternet formation protocols need this phase. The comparative performance evaluation also showed that due to the simplicity of its operations BlueStars [25] is by far the fastest protocol for scatternet formation. However, BlueStars produces scatternets with an unbounded, possibly large number of slaves per piconet, while our method can bound the number of slaves in each piconet.

6 Conclusion and Future Work

In this paper, we have described the first scheme that creates connected degree limited scatternets without parking any node. Nodes are assumed to know their positions and are able to establish connections with any of the neighboring nodes in the neighbor discovery phase. Then, in the next optional phase, we construct a sparse geometric structure as the underlying topology for scatternet. We proposed a new sparse planar structure, partial Delaunay triangulation (PDT), which can be constructed locally and is denser than other known localized structures such as GG and RNG. In the last phase, the degree of each node is limited to seven by applying the Yao structure, and the master-slave relations in piconets are formed in the created structure. The experiments confirm good functionality of created Bluetooth networks in addition to their fast creation and straightforward maintenance.

A number of interesting issues remain for future study. One of the main problems left for future research is to design a fast scheme for the discovery of all neighbors within a transmission radius, since [9] found that device discovery is the most time-consuming operation in practice. Any scheme is applicable to methods described in this paper. For example, the neighbor discovery phase can be made significantly faster if two nodes that just discovered each other also exchange information about other neighbors already discovered. Another variant is to create connected components in the process, and propagate some network information such as component id. In order to accommodate dynamic network

TABLE 2
Number of Slave Nodes with p Masters

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	$S_{>7}$
dUDG	1.60	7.55	11.05	5.55	0.45	0.00	0.00	0.00
gUDG	0.30	3.90	6.95	8.95	5.65	2.30	0.45	0.00
dRNG	9.30	28.10	1.90	0.00	0.00	0.00	0.00	0.00
gRNG	2.60	19.40	16.15	1.05	0.00	0.00	0.00	0.00
dGG	3.15	14.70	11.35	0.55	0.00	0.00	0.00	0.00
gGG	0.95	8.55	14.15	9.30	1.70	0.05	0.00	0.00
dPDL	3.15	14.70	11.40	0.55	0.00	0.00	0.00	0.00
gPDL	0.95	8.55	14.10	9.25	1.75	0.05	0.00	0.00

TABLE 3
Number of MS Nodes with p Masters

graph	M	MS_1	MS_2	MS_3	MS_4	MS_5	MS_6	MS_7	$MS_{>7}$
dUDG	9.85	19.10	27.05	15.55	2.20	0.05	0.00	0.00	0.00
gUDG	25.50	17.70	15.35	8.20	3.60	1.05	0.10	0.00	0.00
dRNG	21.45	31.40	7.85	0.00	0.00	0.00	0.00	0.00	0.00
gRNG	41.90	13.95	4.55	0.40	0.00	0.00	0.00	0.00	0.00
dGG	13.45	27.90	24.70	4.20	0.00	0.00	0.00	0.00	0.00
gGG	32.90	17.30	10.95	3.60	0.55	0.00	0.00	0.00	0.00
dPDL	13.45	27.95	24.55	4.25	0.00	0.00	0.00	0.00	0.00
gPDL	32.90	17.30	10.95	3.65	0.55	0.00	0.00	0.00	0.00

TABLE 4
The Number of Piconets, Master-Slave (Bridge) Nodes, and Size of Piconets

graph	master	slave	master-slave	avg M of S node	avg M of MS node	avg S of (M+MS)
dUDG	9.85	26.20	63.95	2.83	2.02	2.76
gUDG	25.50	28.50	46.00	3.87	2.03	2.84
dRNG	21.45	39.30	39.25	1.81	1.20	1.95
gRNG	41.90	39.20	18.90	2.40	1.28	1.95
dGG	13.45	29.75	56.80	2.31	1.58	2.26
gGG	32.90	34.70	32.40	3.07	1.61	2.43
dPDL	13.45	29.80	56.75	2.31	1.58	2.26
gPDL	32.90	34.65	32.45	3.07	1.61	2.43

scenarios, the discovery phase may be run periodically between actual message traffic rounds.

The proposed schemes can be extended for the single-hop scenarios without position information. As observed in [31], each node can choose its virtual coordinates, and use them to create Bluetooth scatternets as described in this paper.

One of major desirable properties of the proposed dominating set-based method (using a clustering scheme) is that the number of masters that serve as slaves in other piconets is minimized, in fact, limited to gateway piconets. However, this property is not without a cost. The problem with this clustering approach is that the maintenance of the clustered graph structure is expensive since a local change due to mobility may trigger global change in updating the scatternet, thus cluster maintenance overhead has been seen as a serious disadvantage for these protocols [13]. The cluster update scheme can be modified to achieve a localized maintenance property, but at a significant cost of increasing the number of clusters. To address this problem, and still reduce the number of piconets, which is the main problem with the first proposed method here (where higher degree node on any remaining link is the master node), we intend to study alternative way of determining master-slave relations [29].

Notice that, practically, it is hard to have a free-space propagation model, thus the unit disk graph model. The problem arises if a planar structure is to be established, as missing edges may introduce different information at endpoints. The full description of a protocol outlined in this paper needs to address such details and offer resolutions. Along the same lines, details of protocol in the face of node mobility need to be specified.

Routing in Bluetooth has received little attention so far. Bhagwat and Segall [11] proposed a routing method in Bluetooth based on the concept of route vectors. They described protocols for route discovery and packet forwarding. Prabhu and Chockalingam [27] proposed battery power level-based master-slave switch, distance-based power control, and selecting a route path with maximum cumulative battery power (after initial route discovery phase). An important problem is to choose the structure that also provides efficient routing on the designed scatternet, in terms of hop count, power consumption, and delay in message delivery. Most designed structures are planar and, therefore, suitable for routing with guaranteed delivery [10], which is an additional benefit of the proposed structures. The PDT structure is expected to improve the performance of GFG routing algorithm proposed in [10] in both full subgraph variant, and in Bluetooth variant which restricts each node to at most seven neighbors. The performance for broadcasting task can also be considered. Recently, Stojmenovic [29] studied routing in Bluetooth, and proposed to apply RNG, GG, and PDT with some additional long edges selected carefully to improve the routing performance.

There are other technologies where the ideas presented here are applicable for the design of connected degree limited structures, possibly with master-slave relations. For instance, rooftop networks, with antennas placed on the top of buildings, are commercially developed for fast Internet access. Position information is easily and accurately available to nodes, with degree limitation being desirable to avoid congestion at any node.

Finally, the presented algorithms are applicable only when nodes are located in a plane. Scatternet formation for nodes in three-dimensional space (such as a building) remains an interesting challenging problem.

ACKNOWLEDGMENTS

The authors are grateful to the referees for their careful reading which has improved the readability of the manuscript, and to the editor for timely and professional handling and final proofreading of the manuscript. This research is partially supported by NSERC and CONACyT 37017-A.

REFERENCES

[1] Bluetooth SIG, Specification of the Bluetooth System, http://www.bluetooth.com, 2003.

[2] K.M. Alzoubi, P.-J. Wan, and O. Frieder, "Message-Optimal Connected-Dominating-Set Construction for Routing in Mobile Ad Hoc Networks," *Proc. Third ACM Int'l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc'02),* 2002.

[3] M. Ajmone-Marsan, C.F. Chiasserini, A. Nucci, G. Carello, and L. de Giovanni, "Optimizing the Topology of Bluetooth Wireless Personal Area Networks," *Proc. INFOCOM,* 2002.

[4] S. Basagni, R. Bruno, and C. Petrioli, "Performance Evaluation of a New Scatternet Formation Protocol for Multi-Hop Bluetooth Networks," *Proc. IEEE Wireless Personal Multimedia Comm. WPMC,* Oct. 2002.

[5] K. Balaji, S. Kapoor, A.A. Nanavati, and L. Ramachandran, "Scatternet Formation Algorithms in the Bluetooth Network." 2001.

[6] S. Basagni, "Distributed Clustering for Ad Hoc Networks," *Proc. Int'l Symp. Parallel Algorithms, Architectures and Networks ISPAN,* pp. 310-315, June 1999.

[7] S. Basagni and C. Petrioli, "A Scatternet Formation Protocol for Ad Hoc Networks of Bluetooth Devices," *Proc. IEEE Vehicular Technology Conf.,* May 2002.

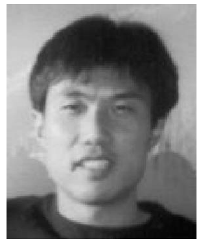
[8] S. Basagni, R. Bruno, and C. Petrioli, "Device Discovery in Bluetooth Networks: A Scatternet Perspective," *Proc. IFIP-TC6 Networking Conf.,* May 2002.

[9] S. Basagni, R. Bruno, and C. Petrioli, "A Performance Comparison of Scatternet Formation Protocols for Networks of Bluetooth Devices," *Proc. IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom),* Mar. 2003.

[10] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," *Proc. Third Int'l Workshop Discrete Algorithms and Methods for Mobile Computing and Comm.,* pp. 48-55, Aug. 1999.

[11] P. Bhagwat and A. Segall, "A Routing Vector (RVM) for Routing in Bluetooth Scatternets," *Proc. IEEE Int'l Workshop Mobile Multimedia Comm. MoMuC,* Nov. 1999.

[12] S. Capkun, M. Hamdi, and J.P. Hubaux, "GPS-Free Positioning in Mobile Ad-Hoc Networks," *Cluster Computing,* vol. 5, no. 2, pp. 157-168, 2002.
[13] L.M. Feeney and M. Nilsson, "Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment," *Proc. IEEE INFOCOM,* 2001.
[14] K.R. Gabriel and R.R. Sokal, "A New Statistical Approach to Geographic Variation Analysis," *Systematic Zoology,* vol. 18, pp. 259-278, 1969.
[15] R. Guerin, E. Kim, and S. Sarkar, "Bluetooth Technology Key Challenges and Initial Research," *Proc. SCS Comm. Networks and Distributed Systems Modeling and Simulation CNDS,* pp. 157-163, 2002.
[16] R. Guerin, S. Sarkar, and E. Vergetis, "Forming Connected Topologies in Bluetooth Adhoc Networks." 2002.
[17] P. Gupta and P.R. Kumar, "Critical Power for Asymptotic Connectivity in Wireless Networks," *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming, W.M. McEneaney, G. Yin, and Q. Zhang (eds.),* 1998.
[18] J.C. Haartsen, "The Bluetooth Radio System," *IEEE Personal Comm.,* vol. 7, pp. 28-36, Feb. 2000.
[19] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," *Computer,* pp. 57-66, Aug. 2001.
[20] X.Y. Li, "Algorithmic, Geometric and Graphs Issues in Wireless Networks," *Wireless Comm. and Mobile Computing,* vol. 3, no. 2, pp. 119-140, 2003.
[21] X.Y. Li and I. Stojmenovic, "Partial Delaunay Triangulation and Degree Limited Localized Bluetooth Scatternet Formation," *Proc. Ad Hoc Networks and Wireless (ADHOC-NOW),* pp. 17-32, Sept. 2002.
[22] C.R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *IEEE J. Selected Areas in Comm.,* vol. 15, no. 7, pp. 1265-1275, 1997.
[23] V.B. Misic and J. Misic, "Bluetoth Scatternet with a Master-Slave Bridge: A Queuing Theoretical Analysis," *Proc. IEEE GLOBECOM,* Nov. 2002.
[24] C. Petrioli and S. Basagni, "Degree-Constrained Multihop Scatternet Formation for Bluetooth Networks," *Proc. IEEE GLOBECOM,* Nov. 2002.
[25] C. Petrioli, S. Basagni, and I. Chlamtac, "Configuring BlueStars: Multihop Scatternet Formation for Bluetooth Networks," *IEEE Trans. Computers,* to appear.
[26] B.J. Prabhu and A. Chockalingam, "A Routing Protocol and Energy Efficient Techniques in Bluetooth Scatternets," *Proc. IEEE ICC,* 2002.
[27] T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaire, "Distributed Topology Construction of Bluetooth Personal Area Networks," *Proc. IEEE INFOCOM,* 2001.
[28] I. Stojmenovic, "Dominating Set Based Bluetooth Scatternet Formation with Localized Maintenance," *Proc. IEEE Int'l Parallel and Distributed Processing Symp. and Workshops,* Apr. 2002.
[29] I. Stojmenovic, "Routing in Bluetooth with Geometric Structures Enhanced with Long Links," pending publication.
[30] G. Toussaint, "The Relative Neighborhood Graph of a Finite Planar Set," *Pattern Recognition,* vol. 12, no. 4, pp. 261-268, 1980.
[31] Y. Wang, I. Stojmenovic, X.-Y. Li, "Bluetooth Scatternet Formation for Single-Hop Ad Hoc Networks Based on Virtual Positions," pending publication.
[32] Z. Wang, R.J. Thomas, and Z. Haas, "Bluenet, a New Scatternet Formation Scheme," *Proc. Hawaii Int'l Conf. System Sciences,* 2002.
[33] J. Wu and W. Lou, "Forward-Node-Set-Based Broadcast in Clustered Mobile Ad Hoc Networks," *Proc. GLOBECOM,* 2002.
[34] A.C.C. Yao, "On Constructing Minimum Spanning Trees in k-Dimensional Spaces and Related Problems," *SIAM J. Computing,* vol. 11, pp. 721-736, 1982.
[35] G.V. Zaruba, S. Basagni, and I. Chlamtac, "Bluetrees—Scatternet Formation to Enable Bluetooth Based Ad Hoc Networks," *Proc. IEEE ICC,* 2001.



Xiang-Yang Li received the MS (2000) and PhD (2001) degrees in computer science from the University of Illinois at Urbana-Champaign. He received the Bachelors degree in computer science and business management from Tsinghua University, People's Republic of China, in 1995. He has been an assistant professor of computer science at the Illinois Institute of Technology since 2000. His research interests span the computational geometry, wireless ad hoc networks, optical networks, and cryptography. He is a member of the ACM and the IEEE.

Ivan Stojmenovic received the PhD degree in mathematics in 1985. He held regular or visiting positions in Serbia, Japan, US, Canada, France, and Mexico. He published more than 160 different papers in journals and conferences, edited "Handbook of Wireless Networks and Mobile Computing" (Wiley, 2002), and coedited "Mobile Ad Hoc Networking" (IEEE Press, 2003). His current research interests include wireless ad hoc, sensor, and cellular networks. He is currently an editor of several journals including *Journal of Multiple-Valued Logic and Soft Computing*, *IEEE Transactions on Parallel and Distributed Systems*, *Parallel Processing Letters*, and *Parallel Algorithms and Applications*. He recently guest edited special issues in several journals including *IEEE Computer Magazine* and *Wireless Communications and Mobile Computing*.

Yu Wang received the BEng degree and the MEng degree from Tsinghua University, Beijing, in 1998 and 2000, both in computer science. He joined the Department of Computer Science, Illinois Institue of Technology, in 2000, as a PhD student. His current research interests include wireless networks, algorithm design, mobile computing, and artificial intelligence. He is a member of the ACM, IEEE, IEEE Computer Society, and IEEE Communication Society.