Partial Delaunay Triangulation and Degree Limited Localized Bluetooth Multihop Scatternet Formation

Xiang-Yang Li*  Ivan Stojmenovic†  Yu Wang*

* Department of Computer Science, Illinois Institute of Technology, 10 W. 31st Street, Chicago, IL 60616, USA. Email xli@cs.iit.edu, wangyu1@iit.edu.
† SITE, University of Ottawa, Ottawa, Ontario K1N 6N5, Canada. Email: ivan@site.uottawa.ca
Abstract

This paper addresses the problem of scatternet formation for multi-hop Bluetooth based personal area and ad hoc networks, with minimal communication overhead. Nodes are assumed to know their positions and are able to establish connections with any of the neighboring nodes, located within the 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 localized manner. We propose here a new sparse subgraph, namely, partial Delaunay triangulation (PDT), which can be constructed efficiently in a localized manner. It is still a planar graph and is denser than other known planar localized subgraphs. In the next mandatory phase, the degree of each node is limited to 7 by applying Yao structure, and the master-slave relations are formed in created subgraphs. This phase follows clustering based approach, and consists of several iterations. In each iteration, undecided nodes with higher keys than any of their undecided neighbors apply Yao structure to bound the degree, decide master-slave relations on the remaining edges, and inform all neighbors about either deleting edge or master-slave decision. We consider two ways to decide master-slave relations: node with initially higher key is master, and cluster based (deciding node becomes master iff it has no previously assigned slave role). In cluster based approach, a dominating set of masters in the degree limited subgraph is implicitly constructed, and some gateway piconets are added to preserve connectivity. Our schemes are the first schemes that construct degree limited and connected piconets in multi-hop networks, without parking any node. They also limit to 7 the number of slave roles a node can accept. 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.

I. 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. Communication between the devices (called nodes hereafter) in the ad-hoc network can be single hops or multiple hops. 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 [24] is well suited to provide ad-hoc networking for the consumer market. Bluetooth ad-hoc
networking presents some technical challenges, such as scheduling, network forming and routing. User mobility poses additional challenges for connection rerouting and QoS services. It has been widely predicted that Bluetooth will be the major technology for short range wireless networks and wireless personal area networks. Due to the expected low cost of Bluetooth chips, Bluetooth based networks are expected to be a preferred solution to build inexpensive but large ad hoc networks, that is, multi-hop radio networks whose nodes may all be mobile. This paper deals with the problem of building 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, 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 resulted by 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 7. If a master node has more than 7 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 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.

While several solutions and commercial products have been introduced for one-hop Bluetooth communication, the problem of scatternet formation has not been dealt with until very recently. Several criteria could be set as the objectives in forming scatternet. First of all, the resulting network should be connected. Secondly, the number of piconets (i.e., the number of nodes assigned master role) should be minimized to provide faster routing. Thirdly, the formation and maintenance of scatternet should have small communication
overhead. This is achieved by applying localized approach, where each node makes decision based on the local knowledge (preferably only its direct neighbors). Fourthly, the protocol should create degree limited scatternets, to avoid parking any node.

Our solutions are based on the assumption that each node knows 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 accuracy around ten meters, the modern systems have accuracy up to three meters [25]. Indoor location systems are based on the proximity of fixed objects with known coordinates (e.g. sensors), measuring angle of arrival and time delays of signals. Active Badge system, for example, has accuracy within 9 cm of their true position [25], with the 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 co-ordinates of neighboring nodes can be obtained by exchanging such information between neighbors [16]. 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 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 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. Previous literature on scatternet formation assumed that 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, which are synchronized with the transmitter, can hear the transmission. Synchronizing the frequency hopping patterns is apparently a time consuming and pseudo-random process [42]. In this paper we assume that the problem of discovering all neighbors within transmission radius of all neighbors is resolved by separate Bluetooth protocol. One such protocol for discovering all one hop networks is described in [42], [13], while a protocol that provides two-hop information to every node is described in [35]. These protocols are applicable as the first phase of our scheme.

The rest of the paper is organized as follows. In Section II, 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 a new planar localized sparse graph, called partial Delaunay triangulation (PDT), in Section III. Section IV presents our new Bluetooth formation algorithms. We apply Yao structure on unit disk graph or a localized geometric structure, such as Gabriel graph (GG), relative neighborhood graph (RNG), PDT or Yao graph, and prove that it limits the degree of each node to 7 and leaves the graph connected (and planar if the selected structure was planar). The last step is to assign roles to nodes, and we describe two such methods: setting the higher degree node of an edge as master, and dominating set scheme based on clustering and adding two-element gateway piconets. We therefore obtain the first Bluetooth scatternet formation algorithm for multi-hop network which limits the degree of each node to 7, keeps the connectivity of all the piconets, and does not park any node. Section V describes the experimental result of our algorithm. We conclude our paper in Section VI by pointing out some possible future research directions.

Preliminary conference version of this article appeared in [31]. This version contains an improved algorithm and experimental data contributed by the added third author, expanded literature review, and better overall presentation.

II. 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, especially the Bluetooth.
A. Geometry 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. We call all nodes within a constant $k$ hops of a node $u$ in the unit disk graph $UDG(S)$ as the *$k$-local nodes* of $u$, denoted by $N_k(u)$. Usually, here 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 [32].

Let $disk(u, v)$ be the disk with diameter $(u, v)$. Then, the *Gabriel graph* [21] $GG(S)$ contains edge $(u, v)$ if and only if $disk(u, v)$ contains no other points of $S$. $GG(S)$ is a planar graph (that is, no two edges cross each other). It was proved in [14] that the intersection of a 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*. If it is clear from the context, we also denote it by $GG(S)$ hereafter. Obviously, $GG(S)$ can be constructed in a localized manner. In other words, a node $u$ can compute its incident edges in $GG(S)$ by using only 1-hop neighbors $N_1(u)$. Each node $u$ can test whether an edge $uv$ belongs to $GG(S)$ in $O(d)$ computation time, where $d$ is the cardinality of $N_1(u)$, by verifying whether $\|wp\| > \|uv\|/2$ is satisfied for each of its neighbors $w$, where $p$ is the midpoint of $uv$. Thus, node $u$ can compute all its incident Gabriel edges in time $O(d^2)$. Alternatively, $u$ can detect which edges belong to $GG(S)$ in $O(d \log d)$ time by constructing the Delaunay triangulation of $N_1(u)$ and its dual Voronoi diagram and preserving these edges of $Del(N_1(u))$ which intersect their corresponding Voronoi edge in Voronoi diagram.

The *relative neighborhood graph*, denoted by $RNG(S)$, is a geometric and graph theoretic concept proposed by Toussaint [48]. 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 $S$. 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 \textit{constrained relative neighborhood graph}, which is also denoted by $RNG(S)$ hereafter if it is clear from the context. Note that $RNG$ and $GG$ (and $PDT$ introduced later in this paper) require no message exchange between nodes, in addition to maintaining unit graph.

The Yao graph [53] is proposed by Yao to construct MST of a set of points in high dimensions efficiently. At given node $u$, any $k$ equal-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 $u \rightarrow v$. Ties are broken arbitrarily. The remaining edges are deleted from the graph. There are several variants on how this construction can be carried out 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). The other choice (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 definition of the Delaunay triangulation. We assume that there are no four vertices of $S$ that are co-circular. A triangulation of $S$ is a \textit{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 \textit{Delaunay triangle} if its circumcircle is empty of vertices of $S$. 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. Then obviously, the Gabriel graph $GG(S)$ and the relative neighborhood graph $RNG(S)$ are subgraphs of the Delaunay triangulation $Del(S)$.

A subset of vertices in a graph $G$ is a \textit{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 [33]. 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 minimal number
of such pairs of gateway nodes is described in [17]. An improved scheme is proposed in [52].

B. Literature Review on Bluetooth Scatternet Formation

Although describing methods for device discovery and for the participation of a node to multiple piconets, the Bluetooth specification does not indicate any method for scatternet formation. The solutions proposed in literature can be divided into single-hop ([49], [42], [28], [43], [18], [7], [9], [36], [47]) and multi-hop ([54], [10], [50], [22], [23], [3], [12], [35]) 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 multi-hop scenarios, but are clearly applicable to single-hop networks as well.

Zaruba, Basagni and Chlamtac [54] 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 be now 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 [54] observed that if a node in unit disk graph has more than five neighbors, then at least two of them must be connected. This observation is used to re-configure the tree so that each master node has no more than 5 slaves. If a master node has more than 5 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. However, whether this approach will terminate is not proved in [54]. We notice that, since every node has degree at most 6 in the Euclidean minimum spanning tree (EMST), we can avoid this branch reorganization by using EMST initially. Tan et al. [49] proposed a similar method, but are restricted to single-hop scenarios. In the second protocol [54], several roots are
Initially selected. Each of them then creates its own scatternet as in the first protocol. In the second phase, sub-tree scatternets are connected into one scatternet spanning the entire network.

Law, Mehta and Siu [28] described an algorithm that creates connected degree bounded scatternet in single-hop networks. The final structure is a tree-like scatternet, which limits efficiency and robustness. A single-hop Bluetooth scatternet formation scheme based on 1-factors is described in [7]. However, piconets are not degree limited in that scheme.

Salonidis et al. [42] proposed another topology construction algorithm recently. It first collects neighborhood information using an inquiry procedure, where senders search for receivers on randomly chosen frequencies, and the detected receivers reply after random backoff delay. Leader is elected in the process, one for each connected component. Leader then collects the information about the whole network, decides the roles for each node, and distributes back the roles. In other words, basically, it is a centralized approach. Thus, the solution is not scalable, and not localized. Moreover, how to assign the roles is not elaborated in [42]. They also assume up to 36 nodes in the network. Another centralized solution for single-hop networks, where the traffic between any pair of nodes is known a priori, is described in [36].

Sun, Chang and Lai [43] described a self-routing topology for single-hop Bluetooth networks. Nodes are organized and maintained in a search tree structure, with Bluetooth ID’s as keys (these keys are also used for routing). It relies on a sophisticated scatternet merge procedure with significant communication overhead for creation and maintenance. Blueerings as scatternets are proposed in [18]. Ring structure for Bluetooth has simplicity and easy creation as advantage, but is applicable only for single-hop networks.

Barriere, Fraigniaud, Narajanjan, and Opatrny [9] described a connected degree limited and distributed scatternet formation solution based on projective geometry. They also described procedures for adding and deleting nodes from the networks.

A greedy centralized multi-hop algorithm, where a hypothetical central entity knows the complete topology has been proposed in [10]. Distributed algorithms have also been proposed in [10], 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 [10] applies a variant of clustering algorithm with limiting number of nodes in each cluster 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 [12], [37] described multi-hop scatternet formation scheme based on clustering scheme [33], 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 [33]), that express their suitability to become masters, following a variant of clustering method described in [11]. All clusterhead nodes are declared master nodes in a piconets, with all nodes belonging to their clusters as their slaves. Some of the slaves become masters of additional piconets, following [17], 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 its all 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 [12], [37] is given in [4].

References [50], [22] essentially propose variants of clustering based scatternet formation
scheme, where 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 connected structure. The counterexample is the same that applies to [10]. On a positive side, [50] proposes two excellent measures for the performance of scatternets: average shortest-path length and maximum traffic flow.

Guerin, Sarkar and Vergetis [23] 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 maximum degree 5 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. The scheduling was discussed by Baatz et al. [6].

Our schemes, originally proposed and made available in June 2001 (e.g., cited in [35]), are the first schemes that construct degree limited and connected piconets in multi-hop networks, without parking any node. Recently, another such scheme has been described by Petrioli and Basagni [35]. Their neat scheme does not require position information, but instead the local information is extended to two-hop information, with two rounds 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 7 neighbors, it chooses up to 7 slaves among them so that it can reach all the others via them. Such a coverage is always possible with up to 5 slaves [54]. 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. The simulations by authors show that created scatternets have low average number of roles per node (about two), with average path length increase between 20% and 80%. 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 stronger assumption, position information. Without position information, the method [35] is currently the best available method.

C. Literature Review on Sparse Geometric Structures

Sparse geometric structures that can be defined locally have been applied in wireless networks for localized routing and broadcasting algorithms. Gabriel graph was used in [8], [14] in order to define planar subgraph used for recovery routing to guarantee delivery, when simple heuristics fail. Gabriel graph was replaced in [20] by newly proposed restricted Delaunay graph, consisting of all the Delaunay edges with length up to transmission radius, possibly with some additional edges. However, the construction process requires additional nontrivial communication between nodes when they move or change activity status (in addition to position exchange), which is avoided with the partial Delaunay triangulation structure proposed in this paper. Relative neighborhood graph was used in [44] to provide efficient localized broadcasting for one-to-one models of wireless communications. Li et al. [29] proposed to use Gabriel graphs, RNGs, and Yao graphs to construct sparse power efficient networks. They also defined various graphs by combining the Gabriel graph structure and the Yao graph structure in order to bound the node degrees in network topology, while the energy consumption of connecting any two nodes is still within a constant factor of the minimum. Other references, applying geometric
structures in wireless networks, are surveyed in [30].

III. PARTIAL DELAUNAY TRIANGULATION

We shall now propose a planar geometry structure which can be used in the first step of our algorithm. The motivation for the introduction of new planar locally defined sparse graph is to improve the graph connectivity of planar graph, 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 $\text{Del}(S)$ of a set of wireless nodes $S$ cannot be constructed in a localized manner. In this section, we will propose 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. 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 $\text{Del}(S)$ 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 $\text{disk}(u, v)$ 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 $\text{disk}(u, v)$ then $uv$ does not belong to $\text{Del}(S)$.

Suppose now that only one side of line $uv$ contains nodes inside the circle $\text{disk}(u, v)$, and let $w$ be one such point that maximizes the angle $\angle uvw$. Let $\alpha = \angle uvw$. Consider now the largest angle $\angle uxv$ on the other side of the mentioned circle $\text{disk}(u, v)$, where $x$ is a node from the set $S$. If $\angle uvw + \angle uxv > \pi$, then edge $uv$ is definitely not in the Delaunay triangulation $\text{Del}(S)$. Edge $uv$ belongs to $\text{Del}(S)$ if $\angle uvw + \angle uxv < \pi$ (here we assume that no four points of $S$ are co-circular). 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.

**Lemma 1:** Assume only $N_1(u)$ is known to $u$, and there is one node $w$ from $N_1(u)$ that is inside $\text{disk}(u, v)$ with the largest angle $\angle u w v$. Edge $w v$ 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 $w v$ 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 $\alpha = \angle u w v$ (here $\alpha \geq \frac{\pi}{2}$).

**Proof:** Consider any edge $w v$. See left figure of Figure 1 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 $w v$ 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 $2r = \frac{d}{\sin \alpha} < R$.

![Figure 1. Scenarios for deciding a Delaunay edge.](image)

**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 $\text{disk}(u, v)$ with the largest angle $\angle u w v$. Edge $w v$ 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 $w v$ 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 $\alpha = \angle u w v$.

**Proof:** Consider any edge $w v$. See right figure of Figure 1 for an illustration. Here $p z$ is the perpendicular bisector of edge $w v$. Edge $w v$ 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 $u_z < R$, where $u_z = \frac{d}{2 \cos(\alpha/2)}$. 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 $\text{Del}(S)$ 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.

IV. NEW SCATTERNET FORMATION ALGORITHMS

We now proceed to describe several localized scatternet formation algorithms, based on sparse geometrical structures. The algorithms have several phases which are shown in following algorithm.

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).

Initially all nodes are undecided. 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 degree (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.

Repeat the iterations, until all nodes are decided.

**Algorithm 1**: Scatternet Formation Algorithms

A. **Neighbor discovery and information exchange**

Firstly, in the neighbor discovery phase, each node learns about its one-hop or two-hop neighbors, in full accordance with Bluetooth specifications. Master-slave relations are decided based on a key. Several different keys can be considered. If node’s Bluetooth ID is used as a key, one-hop information suffices in our protocols. Such neighbor discovery can be performed by a scheme described in [42], [13], already discussed in the literature review section. 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 [42], [12]. To exchange information, two neighboring nodes must be in complementary states. 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 \( (\text{degree, ID}) \), where node degree is primary key, and ID is secondary key, one-hop neighbor discovery is not sufficient to exchange correct information about 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 [35] and is applicable here. It has some lengthy but straightforward details and we will not describe it here.

In order to limit the number of slave roles a node can accept, the proof of Theorem 3 below needs edges to make all edge length from a given node to have mutually different lengths. To break the ties conveniently, each edge \( uv \) is receiving label \( ||uv|| = (\text{length}(uv), \text{key}(u), \text{key}(v)) \), where \( \text{length}(uv) \) is the distance between \( u \) and \( v \). Two edges are compared by their length first. If the length is the same then they are compared by their keys values. Since keys of neighbors (involving their IDs) are unique, no two edges out of the same nodes have equal label (thus there are no ties when \( \text{RNG, GG} \) or \( \text{PDT} \) are defined).

\[B. \text{ Planar subgraph construction} \]

This phase is optional. The remaining phases can be applied on the unit graph directly, but will result in non-planar graphs. Planarity may be a desirable property in some cases, e.g., routing with guaranteed delivery. In this phase, each node computes which of its incident edges belongs to chosen planar sparse structure, RNG (suitable for broadcasting applications), 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 exchanges is therefore reduced.

C. Bounding Degree and Assigning Roles

In the next (mandatory) phase, the degree of each node is limited to 7 by applying Yao structure, and the master-slave relations are formed in created subgraphs. Each node applies 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 7. 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 7 (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 Yao structure to several nodes in each iteration. In the simultaneous approach, described in detail in subsequent work [45], 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 7. Each node creates a key for comparison with neighboring nodes. We consider two possibilities for the key selection. 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 refer to such choice only in the sequel.

This phase follows clustering based approach, and consists of several iterations. 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 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 subsection 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 Yao construction. Then node $u$ divides the region surrounding it into 7 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 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, active node makes 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 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 one. 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 3:** The iterative application of 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, $\angle wuw < \frac{\pi}{3}$ because $k = 7$. Since $||uw|| < ||uv||$ (note that we modified the definition of length so that edge lengths from 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 \( ||uv|| = (\text{length}(AB), \text{key}(A), \text{key}(B)) \). 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 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 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 \( ||uv|| < ||uv|| \) 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 7 in a series of iterations. In addition, the constructed topology may be a planar graph, if we decide so, which makes possible to implement some geometry-position based routing algorithms [14]. 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: node with initially higher key is master, and cluster based. Both methods keep all links “saved” by Yao structure in the final Bluetooth topology but converts them to directed edges, so that one node on each edge is master node, and the other is slave node.

The first method assigns roles based on the information on each link. Each node creates a key, either \( ID \) or \( (\text{degree}, ID) \), 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 higher key becomes the master node, and the other node is the slave node. The purpose of such 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 $d_\ast$, where $\ast$ denotes 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 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 Yao structure at the node (see previous subsection).
There are three cases for assigning role: (1) An active node decides to serve as the master
node if it has only master role or was unassigned. It notices its undecided neighbors to add
a slave role. Such decision indicates that the node is creating a piconet. Notice that here
an active node could get 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 notices 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 master-slave roles and notices 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 notice all its undecided
neighboring nodes to add a master role instead. Such resolution would correspond to the
outcome of clustering algorithm [33]. 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 masterslave node. Figure 2 illustrates the
detailed iterations of assigning roles for an example network. Let us refer to the algorithms that create scatternets with the cluster based approach as $g^*$, where $*$ denotes the name of the sparse topology from the second phase.

![Diagram showing scatternet iterations](image)

**Before Assigning**  
Nodes 4,16 decided  
Nodes 1,2,3,6,14 decided

**After 3rd iteration**  
Nodes 5,7,11,12 decided  
Nodes 9,10,13,15 decided  
Node 8 decided

**After 4th iteration**  
**After 5th iteration**

Fig. 2. An example of scenarios for assigning roles: five iterations. Here an unassigned node is represented by a white disk; a master node is represented by a black square; a slave node is represented by a green disk; a masterslave node is represented by a red triangle; a deleted edge by applying Yao structure is represented by a dashed line.

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

**Theorem 4:** The scatternet formed by the above method is connected.

**Proof.** Remember that we have shown that the structure by bounding the node degree is a connected graph (see Theorem 3). Consider any two piconets centered at node $u$ and $v$. Since the underlying structure is connected, we know that there must have a path to connect $u$ and $v$. For any link $xy$ in the path, since every node will become active once, assume that $x$ becomes active before $y$. From our role-assignment method, if that time
If \( x \) is unassigned, master or masterslave, then it will form a scatternet to connect \( y \); if that time \( x \) is slave, then \( y \) will be assigned a master role, so the link \( xy \) also exists in the scatternet. This finishes 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 piconet since master node needs to switch occasionally to another piconet on 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 [17], [1], [2], [52] to minimize the number of added two node piconets and create a connected scatternet. Using geometry property, 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 clusterhead method [17], [1], [52]. 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 number of piconets.

V. EXPERIMENTS

In this section, we present our experimental results that compare designed algorithms in terms of various characteristics. Here the role of a node could 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 \( MS_p \), here \( p \) is the number of piconets in which this node serves as slave.

![Unit disk graph and its planar subgraphs.](image)

Fig. 3. Unit disk graph and its planar subgraphs.
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 total \( n = 100 \) 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 as \( r = 10 \) meters. Then the graph density (average number of neighbors) of \( UDG \) can be calculated approximately by an area argument: 

\[
(n - 1) \times \pi \times \frac{r^2}{a^2} \simeq 99 \times 3.14 \times \frac{10^2}{50^2} \simeq 12.4.
\]

All results are the averages on total 20 wireless nodes sets.

![Fig. 4. Geometry structures, bounding node degree, and assign roles.](image)

Figure 4 illustrates the different Bluetooth structures using UDG, RNG, GG, or PDT as topology (shown in Figure 3), bounding degree by applying 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 black squares and red triangles respectively, while the slaver nodes are denoted by green disks.

Table I lists the number of slave nodes that serve as slaves of \( p \) piconets under different
TABLE I
Number of slave nodes with \( p \) masters.

<table>
<thead>
<tr>
<th></th>
<th>( S_1 )</th>
<th>( S_2 )</th>
<th>( S_3 )</th>
<th>( S_4 )</th>
<th>( S_5 )</th>
<th>( S_6 )</th>
<th>( S_7 )</th>
<th>( S_{&gt;7} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>dUDG</td>
<td>1.60</td>
<td>7.55</td>
<td>11.05</td>
<td>5.55</td>
<td>0.45</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>gUDG</td>
<td>0.30</td>
<td>3.90</td>
<td>6.95</td>
<td>8.95</td>
<td>5.65</td>
<td>2.30</td>
<td>0.45</td>
<td>0.00</td>
</tr>
<tr>
<td>dRNG</td>
<td>9.30</td>
<td>28.10</td>
<td>1.90</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>gRNG</td>
<td>2.60</td>
<td>19.40</td>
<td>16.15</td>
<td>1.05</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>dGG</td>
<td>3.15</td>
<td>14.70</td>
<td>11.35</td>
<td>0.55</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>gGG</td>
<td>0.95</td>
<td>8.55</td>
<td>14.15</td>
<td>9.30</td>
<td>1.70</td>
<td>0.05</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>dPDL</td>
<td>3.15</td>
<td>14.70</td>
<td>11.40</td>
<td>0.55</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>gPDL</td>
<td>0.95</td>
<td>8.55</td>
<td>14.10</td>
<td>9.25</td>
<td>1.75</td>
<td>0.05</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Bluetooth topologies. Table II lists the number of master-slave nodes that serve as slaves of \( p \) piconets under different Bluetooth topologies. We conducted extensive simulations using different transmission range (from 10m to 100m) and different number of nodes (from 20 to 500). We find that the results are stable, i.e., the portion of the bridge nodes. In addition, as we expected, the cluster based method generates smaller number of nodes with masterslave roles than the method comparing degrees of two end-points of a link.

Table III presents the average number of slave nodes assigned to a node with master role, i.e., 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 assign 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.

We found that the UDG consistently performs the worst among all underlying structures: it has less pure master node, 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 belonging to. We prefer to use PDT since it has more edges than other two structures, thus, can sustain more link failures, and
TABLE II

Number of MS nodes with $p$ masters.

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

have shorter path for some pair of nodes. We found that scatternets generated based on GG and PDT are similar when node density are high, due to the fact that PDT has only very few more edges than GG.

VI. Future work

We have described the first scheme that creates connected degree limited scatternets. A number of issues remain for future study.

A. Neighbor Discovery

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. Any scheme is applicable to this paper. The issue is clearly not in technical feasibility (one straightforward solution is presented in [42], [12]), but the time bounds for achieving this goal. For example, the neighbor discovery phase can be sped up significantly if two nodes that just discovered each other also exchange the information about other neighbors already discovered [46]. This provides two-hop information, or even additional one-hop neighbors to the other node if position information is available. 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 scenarios, the discovery phase may be run periodically.
TABLE III
THE NUMBER OF PICONETS, BRIDGE NODES, AND SIZE OF PICONETS.

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

between actual message traffic rounds [46].

B. Scatternet Formation

The proposed schemes can be extended for the single-hop scenarios without position information. As observed in [47], 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 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 clustering approach is that the maintenance of 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 [19]. Cluster update scheme can be modified to achieve 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. One candidate is a simple and localized connected dominating set structure proposed by Wu and Li [51]. It
has been used to define scatternet formation with localized maintenance in [45].

Notice that, practically, it is hard to have a free-space propagation model, thus unit disk graph model. However, our approach works well even if some edges are not discovered, as long as sufficient number of edges are discovered to establish network connectivity. 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.

C. Routing in Bluetooth

Routing in Bluetooth received little attention so far. Bhagwat and Segall [15] proposed a routing method in Bluetooth based on a concept of route vector. They described protocols for route discovery and packet forwarding. Prabhu and Chockalingam [38] proposed battery power level based master-slave switch, distance based power control, and selecting 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 (the delay depends on the amount of multiple roles performed by various nodes). Most designed structures are planar and therefore suitable for routing with guaranteed delivery [14], which is an additional benefit of proposed structures. PDT structure is expected to improve the performance of GFG routing algorithm proposed in [14] 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.

The routing problem in Bluetooth, however, is the last link in a chain that starts with Bluetooth scatternet formation. Based on expected traffic load, each link should be assigned an appropriate maximum capacity (which was recently investigated in [55]) so that the sum of delays on each link is minimized. Capacity assignment is the link between scatternet formation and scatternet scheduling algorithms (investigated in [5], [26], [40]). Routing is then designed as the last link in this chain. The application of existing routing algorithms that guarantee delivery [8], [14], [20] in UDG for scatternet networks shall be investigated. Thus, it is interesting to see how PDT performs compared to other geometry
structures in terms of routing efficiency, the quality of the selected routes and so on.

D. Connected degree limited structures for other technologies

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. Another example is the ultra-wideband transmission, which involves transmitting very short pulses on a wide range of frequencies simultaneously at low power. The technology received massive boost recently, when it received limited approval for transmissions up to ten meters. It is capable of data rates of over 100 megabits per second on such short distances, far more than Bluetooth. Work is well advanced on the standard to enable UWB devices to locate and communicate each other. Ad hoc networking is expected to receive further boost after adopting UWB transmission. Some are even referring to infrastructureless, ad hoc UWB networks as 5G. Most important feature of UWB, as related to this paper, is that UWB enables position aware information. The solution proposed in this paper might therefore be easily adaptable and directly applicable to UWB based networks.

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.

REFERENCES


