# Partial Delaunay Triangulation and Localized Bluetooth Scatternet Formation

XIANG-YANG LI
*Illinois Institute of Technology, USA.*

IVAN STOJMENOVIC
*University of Ottawa, Canada.*

**Abstract**

This paper addresses the problem of scatternet formation algorithm for multi-hop Bluetooth based personal area and ad hoc network, with minimal communication overhead. Nodes are assumed to know their position and are able to establish connections with any of the neighboring nodes. We first propose 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. The proposed formation algorithm constructs first a sparse geometric structure in localized manner and then limits the degree of each node to 7 by applying Yao structure. The master-slave relations are formed in created subgraphs by using higher degree priority. The creation and maintenance require minimal overhead in addition to maintaining accurate location information for one-hop neighbors. We also propose an alternative master-slave decision scheme that reduces significantly the number of created piconets. A dominating set of masters in the degree limited subgraph is constructed by applying clustering scheme, and then a two-element piconet is added for each two masters at distance three, to preserve connectivity. Our schemes are the first schemes that construct degree limited and connected piconets in multi-hop networks, without parking any node. The experiments confirm good functionality of created Bluetooth networks in addition to their fast creation and straightforward maintenance.

## 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. 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. Bluetooth [14] 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. This paper deals with the problem of building ad hoc networks using Bluetooth technology.

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. Multiple roles for the same device is possible. 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 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 should be minimized to provide faster routing. Thirdly, the formation and maintenance of scatternet should have small communication overhead. Fourthly, a localized approach is preferred, in order to minimize the overhead and avoid global scatternet updates caused by node mobility.

A broad variety of location dependent services will become feasible in the near future due to the use of the Global Position System (GPS). If GPS is not available, the distance between neighboring nodes can be estimated based on the incoming signal strengths or time. Relative co-ordinates of neighboring nodes can be obtained by exchanging such information between neighbors [8]. Here we make use of location information of every node in the network, which is communicated and updated with neighboring nodes only. 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 iff 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.

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 [26]. In this paper we assume that the problem of discovering all neighbors within transmission radius of all neighbors is resolved by separate Bluetooth protocol. We make such an assumption, and leave the neighbor discovery problem as a separate issue.

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. Section 3 presents our new scatternet formation algorithms. We first describe a new planar localized sparse graph, called partial Delaunay triangulation (PDT). We then apply Yao structure on some localized geometric structure, such as PDT, GG, RNG 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: higher degree node is master node in any edge, 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 4 describes the experimental result of our algorithm. We conclude our paper in Section 5 by pointing out some possible future research directions.

# 2 Preliminaries

## 2.1 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 $UDG(S)$ 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 [20].

Let $disk(u,v)$ be the disk with diameter $(u,v)$. Then, the *Gabriel graph* [12] $GG(S)$ 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 [6] that the intersection of a connected $UDG(S)$ and $GG(S)$ is a connected planar graph. In this paper, we only consider $UDG(S) \cap GG(S)$, which is called the *constrained Gabriel graph*. If it is clear from the context, we also denote it by $GG(S)$ hereafter. It is obvious that the graph $GG(S)$ 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 edge $uv$ belongs to $GG(S)$ in $O(d)$ 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 $uv$. 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 $RNG(S)$ [29] consists of all edges $uv$ such that there is no point $w \in S$ such that $\|uw\| < \|uv\|$ and $\|wv\| < \|uv\|$. Obviously, $RNG(S)$ 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 $UDG(S) \cap RNG(S)$ the *constrained relative neighborhood graph*, which is also denoted by $RNG(S)$ hereafter if it is clear from the context.

The *constrained Yao graph* [34] with an integer parameter $k \geq 6$, denoted by $\overrightarrow{YG}_k(S)$, is defined as follows. At each 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 $\overrightarrow{uv}$. Ties are broken arbitrarily. Let $YG_k(S)$ be the undirected graph by ignoring the direction of each link in $\overrightarrow{YG}_k(S)$. If we add the directed link $\overleftarrow{uv}$ instead of $\overrightarrow{uv}$, we call the resulted graph the *reverse of Yao graph*, denoted by $\overleftarrow{YG}_k(S)$.

A triangulation of $S$ is a *Delaunay triangulation*, denoted by $Del(S)$, if the circumcircle of each of its triangles is *empty*, i.e., does not contain any other vertices of $S$ in its interior. We assume that no four vertices of $S$ are co-circular. A triangle is called the *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 an *empty* circle, containing $u$ and $v$ on its boundary. Then obviously, $GG(S)$ and$RNG(S)$ are subgraphs of $Del(S)$.

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. 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 [9].

## 2.2 Literature Review on Bluetooth Scatternet Formation

Zaruba, Basagni and Chlamtac [35] proposed two protocols, called *bluetree*, for forming connected scatternet. 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 *blue-*

*root*, which will be the root of the bluetree. Then assign the role alternatively to node based on its level in the spanning tree. Tan [30] proposed a similar method. A special treatment [35] is enforced to bound the slaves of each master node. Whether this treatment will terminate is not proved. In the second protocol, 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 by running the first protocol and treating each sub-tree as a virtual vertex. There is an extensive communication between nodes until the roles are established, and the movement of each node could trigger global update in the subtree rooted by that moving node and the subtree the node would move into. Thus both formation and maintenance of the scatternet could require considerable communication overhead, which is not measured in [35].

Salonidis *et al.* [26] proposed another topology construction algorithm. 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 [26].

Law, Mehta and Siu [18] described an algorithm that creates connected degree bounded scatternet when the devices are in-range of one another, i.e., single-hop network. In this paper we consider a more general multi-hop model of ad hoc networks. The results described here can be also applied for single-hop networks.

The feasibility problem of scatternet formation (requiring Bridges to be Slave-Slave) for multi-hop networks, has been proved to be NP-Complete [5]. A greedy centralized algorithm, was proposed. Distributed algorithms were also proposed in [5], which assume 2-hop neighborhood information. The algorithm [5] 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 7 slaves among neighbors, 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.

References [31, 13] essentially proposed 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 page 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 [5]. On positive side, [31] proposes two excellent measures for the performance of scatternets: average shortest-path length and maximum traffic flow.

Marsan *et al.* [22] described a centralized method to find optimal Bluetooth topology that provides network connectivity, fulfills the traffic requirements and the constraints posed by the system specification, and minimizes the traffic load of the most congested node. The scheduling was discussed by Baatz *et al.* [3].

### 2.3 Literature Review on Sparse Geometric Structures

Sparse geometric structures defined locally have been applied in wireless networks for localized routing and broadcasting algorithms. Gabriel graph was used in [4, 6] in order to define planar subgraph to guarantee packets delivery, when simple heuristics fail. Gabriel graph was replaced in [11] by newly proposed restricted Delaunay graph, containing all the Delaunay edges with length up to transmission radius. However, the construction process requires extensive communication, which is avoided with the partial Delaunay triangulation structure proposed in this paper. RNG was used in [27] to provide efficient localized broadcasting for one-to-one models of wireless communications. Li *et al.* [19] proposed to use GG, RNG, and Yao graphs to construct sparse power efficient networks and defined various graphs in order to bound the node degrees, while the energy consumption of connecting any two nodes is still within a constant factor of the minimum. Wattenhofer *at al.* [33] proposed a two-phased network construction: a variation of the Yao structure followed by a variation of the Gabriel graph.

There were also attempts to define some sparse structures in distributed manner, but they were not localized. For example, Hu [15] described a distributed, Delaunay triangulation based algorithm for choosing logical links and as a consequence carrying out topology control. However, the construction of Delaunay triangulation is only partially localized, since construction of some edges may require the knowledge of distant node positions.

## 3 New Scatternet Formation Algorithms

We begin this section by proposing a new planar localized geometric structure, partial Delaunay triangulation, and proceed by describing several localized scatternet formation algorithms, based on sparse geometrical structures. The algorithms have two phases: constructing a bounded-degree sparse topology and assigning roles to each node.

### 3.1 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 $Del(S)$ of a set of wireless nodes $S$ can not be constructed in a localized manner efficiently. Our newly proposed Partial Delaunay triangulation (PDT) 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. 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 that only one side of line $uv$ contains nodes inside $disk(u,v)$, 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 $uv$ is definitely not in $Del(S)$. Edge $uv$ belongs to $Del(S)$ if $\angle uwv + \angle uxv < \pi$. 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. Let $R$ be the transmission radius of each wireless node. Let $disk(u,v,w)$ be the circumcircle passing through $u$, $v$, and $w$. Then whether $uv$ is added to $PDT$ is based on the following lemma. The proof is omitted due to space limit.

**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 (1) there is no node from $N_1(u)$ that lies on the different side of uv with w and inside $disk(u,v,w)$ and (2) $\sin\alpha > \frac{d}{R}$, where $\alpha = \angle uwv$ (here $\alpha \leq \frac{\pi}{2}$).*

*Assume only 1-hop neighbors are known to u and v, and there is one node w from $N_1(u) \cup N_1(v)$ that is inside $disk(u,v)$ with the largest angle $\angle uwv$. Edge uv is added to PDT if (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 $disk(u,v,w)$, and (2) $\cos\frac{\alpha}{2} > \frac{d}{2R}$, where $\alpha = \angle uwv$.*

### 3.2 Bounded Degree

Remember that our scatternet formation algorithm first constructs a bounded degree sparse topology. We propose the following two-step approach: (1) The first step is to create the $PDT$ or other sparse subgraph. (2) The second step is applied only at nodes that have more than 7 neighbors in the constructed subgraph by the first step. These nodes apply Yao structure $YG_k$ on all of its neighbors in order to eliminate some of its neighbors, where $k = 7$. This will guarantee that the number of slaves assigned to any node is no more than 7.

There are two approaches to apply the Yao structure here. The first approach is performed to create an *undirected* graph such that the maximum node degree is at most 7. Each node creates a key $(degree, ID)$ for comparison with neighboring nodes, where *degree* is the node's degree after the first step. A node $u$ performs the second step if it has the highest key value among all of its neighbors that did not yet perform the same step. 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, with ties broken arbitrarily. 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. The simultaneous performing of the same procedure by all such nodes is also possible. However, these simultaneous operations may result in some directional edges. The proof of the following lemma is omitted due to space limit.

**Lemma 2** *The subgraph remains connected if the subgraph constructed in the first step is connected.*

The graph generated by the above approach is an undirected graph and each node has degree of no more than 7. However, this approach need to remove a link $uv$ from both $u$ and $v$. The second approach is to create a *directed* graph such that each node has out-degree at most 7. A node $u$ performs the Yao structure if it has degree more than 7. 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, with ties broken arbitrarily. Node $u$ removes all other connections at $u$. Notice that if $u$ removes a connection $uv$, $v$ is not forced to remove connection $vu$. When a node $u$ is assigned as master, only its out-neighbors are considered as its slave candidates. Thus, it has at most 7 slaves. Similarly, we can prove that the graph remains connected after such step.

## 3.3 Assigning Roles

It remains to determine master and slave roles of all nodes, which is the second phase of our scatternet formation algorithm. We propose two methods for this.

The first method assigning roles based on the information of each link. Each node creates a key $(degree, ID)$, where degree is the number of its neighbors in the sparse topology constructed in the first 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 as $d*$, where $*$ denotes the name of the sparse topology from the first phase. Notice that this method keeps all links of the sparse topology in the final Bluetooth topology by converting it to directed edges.

One problem in above design 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. The second method is designed to minimize the number of nodes with such double role. The basic idea is similar to the approaches to create a dominating set by clustering process [21, 9]. Clusterheads then become master nodes, and no two master nodes are directly connected in this phase. Each node creates a key $(degree, ID)$ for comparison with neighboring nodes. A node is assigned as master if it has the highest key value among all of its neighbors whose roles are not assigned. Then all of its neighbors are assigned as its slaves. The simultaneous performing of the same procedure by all such nodes is also possible. Nodes start to perform the following steps to guarantee the connectivity after all its neighbors' roles have been assigned. Consider two neighboring nodes $u$ and $v$ that are assigned as slaves. Nodes $u$ and $v$ check if they share a master already. If not, the node with the higher value $(degree, ID)$ is assigned as the master, and the other node is assigned as its slave. Let us refer to the algorithms that create scatternets as $g*$, where $*$ denotes the name of the sparse topology from the first phase. Notice that not all links of the sparse topology are kept in the final bluetooth topology by this method. We can also apply the method from [9, 1] to minimize the number of added two node piconets and create a connected scatternet, which is at most a small constant factor of the piconets created.

Notice that if we apply Yao structure in the first phase of our algorithm to create a *directed* topology, then it is possible that some node acts as slaves in many piconets, which causes large switching delay. To remedy this, we then consider *YaoYao* graph [34][19] for scatternet formation, for $k = 7$. In other words, in the first step of our algorithm, each node divides the region into seven equal angular regions centered at the node, and chooses the closest node in each region, if any. We then apply the reverse Yao structure as follows. Each node verifies the number of possible slave roles assigned to it by checking the number of directed edges pointed to it. It then divides the plane into 7 equal regions, and allows slave role to at most one master in each region. Typically, it chooses the one that is closest. The graph remains connected, as proved in [19]. Note that here *YaoYao* is a directed graph due to the graph construction process.

Finally, we can consider another undirected structure, called symmetric Yao graph *YaoS*, which guarantees that the node degree is at most 7. Each node $u$ divides the region into seven equal angular regions centered at the node, and chooses the closest node in each region, if any. An edge $uv$ is selected to graph *YaoS* if and only if both directed edges $uv$ and $vu$ are in the *Yao* graph. Then it is obvious that the maximum node degree is 7. The connectivity of the graph can be proved similarly to the proof of the strong connectivity of the *Yao* graph [20].

# 4 Experiments

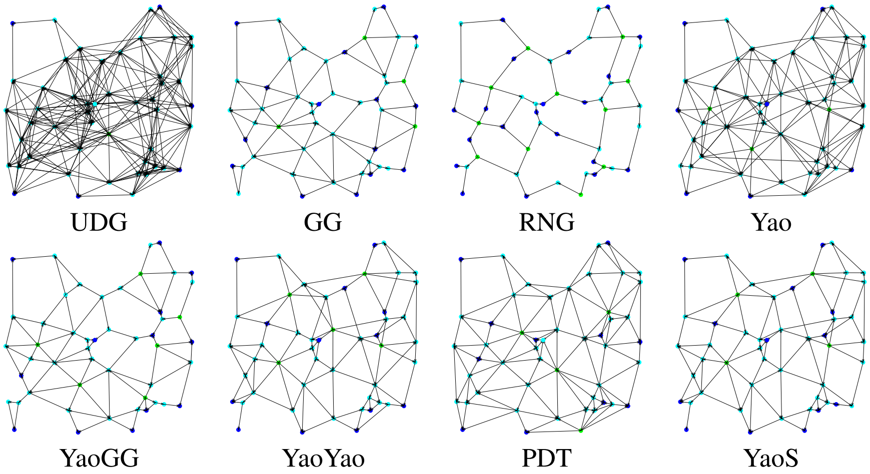


Figure 1: Assign roles based on highest degree on each link.

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.

Since the presented algorithms are the first algorithms that generate degree bounded and connected scatternet structures, there is no scheme that needs to be compared with schemes proposed here. In the experimental results presented here, we choose total $n = 50$ wireless nodes which are distributed randomly in a square area with side length 100 meters. Each node are specified by random $X, Y$ coordinates. The transmission radius of each wireless node is set as 10 meters.

Figure 1 and Figure 2 illustrate the different bluetooth structures using various topologies and assigning node roles by comparing end-nodes degree of each link and comparing the nodes degree with all its unassigned neighbors respectively.

Table 1 and 2 list the number of slave nodes, and the number of master-slave nodes, which serve as slaves of $p$ piconets under different bluetooth topologies respectively. We found that the *YaoYao* graph and the *YaoS* graph not only theoretically guarantee that the number of piconets in which each slave node serves is at most 7, but also is practically good in terms all other measurements. We also found that the results are stable when using different density and the number of nodes. In addition, as we expected, the clustering with added gateways based assigning method generates small number of nodes with master roles than the

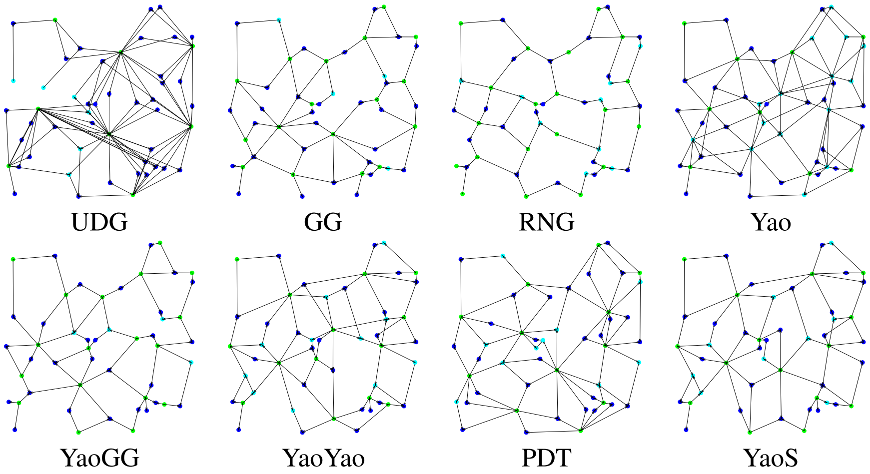


Figure 2: Assign roles based on connected clusters.

method comparing degrees of two end-points of a link.

Table 3 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 second column represents the average number of nodes with master roles only. The third column represents the average number of piconets assigned to a node with slave roles only. The fourth 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 clustering with added gateways 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.

# 5 Future work

One of the main problems left for future research is the discovery of all neighbors within a transmission radius. This paper assumes that this problem is successfully solved with reasonable time delay. Here the issue is clearly not in technical feasibility, but the time bounds for achieving this goal.

Routing in Bluetooth received little attention so far. Bhagwat and Segall [7] proposed a routing method in Bluetooth based on a concept of route vector. They described protocols for route discovery and packet forwarding.

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

Table 1: The average number of nodes with either master or slave role only.

| graph | M | $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ | $S_6$ | $S_7$ | $>7$ |
|---|---|---|---|---|---|---|---|---|---|
| dRNG | 11 | 17 | 21 | 1 | 0 | 0 | 0 | 0 | 0 |
| gRNG | 20 | 5 | 19 | 6 | 0 | 0 | 0 | 0 | 0 |
| dYao | 5 | 4 | 10 | 9 | 11 | 9 | 4 | 1 | 0 |
| gYao | 9 | 6 | 19 | 12 | 4 | 0 | 0 | 0 | 0 |
| dYaoGG | 6 | 11 | 22 | 10 | 1 | 0 | 0 | 0 | 0 |
| gYaoGG | 18 | 5 | 21 | 6 | 0 | 0 | 0 | 0 | 0 |
| dYaoYao | 5 | 9 | 17 | 14 | 5 | 0 | 0 | 0 | 0 |
| gYaoYao | 14 | 6 | 20 | 10 | 0 | 0 | 0 | 0 | 0 |
| dPDT | 4 | 7 | 11 | 14 | 13 | 1 | 0 | 0 | 0 |
| gPDT | 12 | 4 | 26 | 7 | 1 | 0 | 0 | 0 | 0 |
| dYaoS | 4 | 11 | 18 | 13 | 4 | 0 | 0 | 0 | 0 |
| gYaoS | 15 | 5 | 23 | 6 | 1 | 0 | 0 | 0 | 0 |

Table 2: The average number of nodes with a master and few slave roles.

| graph | $MS_0$ | $MS_1$ | $MS_2$ | $MS_3$ | $MS_4$ | $MS_5$ | $MS_6$ | $MS_7$ | >7 |
|---|---|---|---|---|---|---|---|---|---|
| dRNG | 11 | 11 | 8 | 0 | 0 | 0 | 0 | 0 | 0 |
| gRNG | 20 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| dYao | 5 | 8 | 8 | 5 | 5 | 8 | 1 | 3 | 1 |
| gYao | 9 | 9 | 0 | 0 | 0 | 0 | 2 | 4 | 0 |
| dYaoGG | 6 | 14 | 9 | 4 | 3 | 1 | 0 | 0 | 0 |
| gYaoGG | 18 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| dYaoYao | 5 | 11 | 9 | 7 | 2 | 3 | 0 | 0 | 0 |
| gYaoYao | 14 | 10 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| dPDT | 4 | 12 | 8 | 4 | 3 | 5 | 1 | 2 | 0 |
| gPDT | 12 | 9 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| dYaoS | 4 | 12 | 8 | 7 | 2 | 4 | 0 | 0 | 0 |
| gYaoS | 15 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 3: The average number of slave nodes assigned to a node with master role.

| graph | nodes with master role | piconets of slave node | piconets masterslave node | average slaves per node of master role |
|---|---|---|---|---|
| dRNG | 30 | 62 | 27 | 3.0 |
| gRNG | 29 | 61 | 9 | 2.2 |
| dYao | 44 | 171 | 134 | 6.9 |
| gYao | 24 | 96 | 49 | 6.0 |
| dYaoGG | 37 | 89 | 61 | 4.1 |
| gYaoGG | 22 | 65 | 4 | 3.1 |
| dYaoYao | 37 | 105 | 73 | 4.8 |
| gYaoYao | 25 | 74 | 12 | 3.4 |
| dPDT | 39 | 128 | 97 | 5.8 |
| gPDT | 22 | 81 | 11 | 4.2 |
| dYaoS | 37 | 102 | 77 | 4.8 |
| gYaoS | 24 | 73 | 9 | 3.4 |

suitable for routing with guaranteed delivery [6], which is an additional benefit of proposed structures. *PDT* structure is expected to improve the performance of *GFG* routing algorithm proposed in [6] 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 (this problem was recently investigated in [36]) 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 [2, 16, 24]). Routing is then designed as the last link in this chain. The application of existing routing algorithms that guarantee delivery [4, 6, 11] in unit graphs 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.

Many other interesting problems arise from the scatternet formation. The protocols referred to in this paper should be enhanced to include the cases when a node enters, moves within or leaves the network, such that the reorganization incurred is minimum. Although localized approach minimizes communication overhead, exact details of Bluetooth protocol need to be described. Another variation could be to take the services assigned by each node into account and form a scatternet such that inter-piconet communication is minimized.

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


[1] K. M. ALZOUBI, P.-J. WAN, O. FRIEDER, New Distributed Algorithm for Connected Dominating Set in Wireless Ad Hoc Networks, *IEEE HICSS35*, 2002.

[2] R. BRUNO, M. CONTI AND E. GREGORI, Wireless access to Internet via Bluetooth: Performance evaluation of the EDC Scheduling algorithm, *Proc. ACM Workshop on Wireless Mobile Internet*, Rome, Italy, July 2001, 43-49.

[3] S. BAATZ, M. FRANK, P. MARTINI, C. SCHOLZ, Bluetooth scatternets: an enhanced adaptive scheduling scheme, INFOCOM, 2002.

[4] L. BARRIERE, P FRAIGNIAUD, L. NARAJANAN, AND J. OPATRNY, Robust position based routing in wireless ad hoc networks with unstable transmission ranges, *DialM 01*, July 2001, 19-27.

[5] K. BALAJI, S. KAPOOR, A.A. NANAVATI AND L. RAMACHANDRAN, Scatternet Formation Algorithms in the Bluetooth Network, manuscript, 2001.

[6] P. BOSE, P. MORIN, I. STOJMENOVIC AND J. URRUTIA, Routing with guaranteed delivery in ad hoc wireless networks, *DialM 99*, August, 1999, 48-55; *ACM/Kluwer Wireless Networks*, 7, 6, November 2001, 609-616.

[7] P. BHAGWAT AND A. SEGALL, A routing vector (RVM) for routing in Bluetooth scatternets, *IEEE Int. Workshop on Mobile Multimedia Communications*, Nov. 1999.

[8] S. CAPKUN, M. HAMDI, J.P. HUBAUX, GPS-free positioning in mobile ad-hoc networks, *Proc. Hawaii Int. Conf. on System Sciences*, January 2001.

[9] CHLAMTAC, A. FARAGO, A new approach to design and analysis of peer to peer mobile networks, *Wireless Networks*, 5, 149-156, May 1999.

[10] L.M. FEENEY AND M. NILSSON, Investigating the energy consumption of a wireless network interface in an ad hoc networking environment, *IEEE INFOCOM*, 2001.

[11] J. GAO, L.J. GUIBAS, J. HERSHBERGER, L. ZHANG, A. ZHU, Geometric spanner for routing in mobile networks, *ACM MobiHoc*, October 2001, 45-55.

[12] K.R. GABRIEL AND R.R. SOKAL, A new statistical approach to geographic variation analysis, *Systematic Zoology*, 18, 259-278, 1969.

[13] R. GUERIN, E. KIM AND S. SARKAR, Bluetooth technology key challenges and initial research, Proc. SCS Communication Networks and Distributed Systems Modeling and Simulation CNDS, 2002, 157-163.

[14] J.C. HAARTSEN, The Bluetooth radio system, *IEEE Personal Communications*, 7, Feb. 2000, 28-36.

[15] L. HU, Topology control for multi-hop packet radio networks, *IEEE Trans. Communications*, 41, 10, 1993.

[16] N. JOHANSSON, F. ALRIKSSON, AND U. JOHNSSON, JUMP mode - A dynamic window-based scheduling framework for Bluetooth scatternets, *ACM Symposium on Mobile Ad Hoc Networking and Computing MobiHoc*, October 2001, 204-211.

[17] A. KUMAR, L. RAMACHANDRAN AND R. SHOREY, Performance of Network Formation and Scheduling Algorithms in the Bluetooth Wireless Ad-Hoc Network, to appear in the *Journal of High Speed Networks*, IOS Press, 2001.

[18] C. LAW, A.K. MEHTA, AND K.Y. SIU, Performance of a new Bluetooth scatternet formation protocol, *ACM MobiHoc*, October 2001, 183-192.

[19] X.Y. LI, P.J. WAN, Y. WANG AND O. FRIEDER, Sparse power efficient topology for wireless networks, *Proc. HICSS'35*, 2002. Telecommunication Systems, to appear.

[20] X.Y. LI, P.J. WAN, Y. WANG, Power efficient and sparse spanner for wireless ad hoc networks, *Proc. IEEE ICCCN*, 2001. Journal of Paral. & Distri. Sys., to appear.

[21] C. R. LIN, M. GERLA, Adaptive Clustering for Mobile Wireless Networks, *IEEE Journal of Selected Areas in Communications*, Vol 15, 7, pp 1265-1275, 1997.

[22] M.A. MARSAN, C.F. CHIASSERINI, A. NUCCI, G. CARELLO, L. DE GIOVANNI, Optimizing the topology of Bluetooth wireless personal area networks, Proc. INFOCOM, 2002.

[23] L. RAMACHANDRAN, M. KAPOOR, A. SARKAR AND A. AGGARWAL, Clustering Algorithms for Wireless Ad Hoc Networks, *Dial M*, 2000.

[24] A. RACZ, G. MIKLOS, F. KUBINSZKY AND A. VALKO, A pseudo random coordinated scheduling algorithm for Bluetooth scatternets, *Proc. ACM Symposium on Mobile Ad Hoc Networking and Computing MobiHoc*, October 2001, 193-203.

[25] R. RAMANATHAN AND R. ROSALES-HAIN, Topology control of multihop wireless networks using transmit power adjustment, *Proc. IEEE INFOCOM*, March 2000.

[26] T. SALONIDIS, P. BHAGWAT, L. TASSIULAS, R. LAMAIRE, Distributed topology construction of Bluetooth personal area networks, *Proc. IEEE INFOCOM, 2001.*

[27] M. SEDDIGH, J. SOLANO AND I. STOJMENOVIC, RNG and internal node based broadcasting algorithms for wireless one-to-one networks, *ACM Mobile Computing and Communications Review*, Vol. 5, No. 2, 2001.

[28] I. STOJMENOVIC, Dominating set based Bluetooth scatternet formation with localized maintenance, CD Proc. IEEE Int. Parallel and Distributed Processing Symposium and Workshops, Fort Lauderdale, April 2002.

[29] G. TOUSSAINT, The relative neighborhood graph of a finite planar set, *Pattern Recognition*, 12, 4, 1980, 261-268.

[30] G. TAN, A. MIU, J. GUTTAG, H. BALAKRISHNAN, Forming scatternets from Bluetooth personal area networks, *MIT-LCS-TR-826*, October 2001.

[31] Z. WANG, R.J. THOMAS, Z. HAAS, Blueneet, a new scatternet formation scheme, Proc. HICSS 2002.

[32] J. WU AND H. LI, On calculating connected dominating set for efficient routing in ad hoc wireless networks, Proc. DIAL M, Seattle, Aug. 1999, 7-14.

[33] R. WATTENHOFER, L. LI, P. BAHL AND Y.M. WANG, Distributed topology control for power efficient operation in multihop wireless ad hoc networks, Proc. IEEE INFOCOM 2001.

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

[35] G.V. ZARUBA, S. BASAGNI AND I. CHLAMTAC, Bluetrees - Scatternet formation to enable Bluetooth based ad hoc networks, *Proc. IEEE ICC*, 2001.

[36] G. ZUSSMAN AND A. SEGALL, Capacity assignment in Bluetooth scatternets, analysis and algorithms, *CCIT Report 355, Technion, Dept. of Electr. Eng.*, Oct. 2001.