Bluetooth Scatternet Formation for Single-hop Ad Hoc Networks Based on Virtual Positions

Yu Wang* Ivan Stojmenovic† Xiang-Yang Li*

***Abstract*— This paper addresses the problem of scatternet formation for single-hop Bluetooth based personal area and ad hoc networks, with minimal communication overhead. In a single-hop ad hoc network, all wireless devices are in the radio vicinity of each other, e.g., electronic devices in a laboratory, or laptops in a conference room. Recent scatternet formation schemes by Li, Stojmenovic and Wang [1] are position based and were applied for multi-hop networks. These schemes are localized and can construct degree limited and connected piconets, without parking any node. They also limit to 7 the number of slave roles in one piconet. The creation and maintenance require small overhead in addition to maintaining location information for one-hop neighbors. In this article we apply this method to single-hop networks, by showing that position information is then not needed. Each node can simply select a virtual position, and communicate it to all neighbors in the neighbor discovery phase. Nodes then act according to the scheme by Li, Stojmenovic and Wang using such virtual positions instead of real ones. In addition, in this paper we use Delaunay triangulation instead of partial Delaunay triangulation proposed in [1], since each node has all the information needed. Likewise, we can also apply Minimum Spanning Tree (MST) as the planar topology in our new schemes. Finally, we design experiments to study both the properties of formatted scatternets (such as number of piconets) and the performances of different localized routing methods on them. The experiments confirm good functionality of created Bluetooth networks in addition to their fast creation and straightforward maintenance.**

***Index Terms*— System design, graph theory, Bluetooth networks, scatternet formation, single-hop.**

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. Bluetooth [2] 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. This paper deals with the problem of building ad hoc networks using Bluetooth

* Department of Computer Science, Illinois Institute of Technology, 10 W. 31st Street, Chicago, IL 60616, USA. Email: wangyu1@iit.edu, xli@cs.iit.edu. Tel: 1-(312)-567-5207. Fax: 1-(312)-567-5067.

† SITE, University of Ottawa, Ottawa, Ontario K1N 6N5, Canada. Email: ivan@site.uottawa.ca. Fax: 1-(613)-822-0057.

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 should be minimized to provide faster routing. Thirdly, the formation and maintenance of scatternet should have small communication overhead. Fourthly, the protocol should create degree limited scatternets, to avoid parking any node.

In this paper, we focus on scatternet formation for single-hop ad hoc networks. In a single-hop ad hoc network, all wireless devices are in the radio vicinity of each other, e.g., electronic devices in a laboratory, or laptops in a conference room. A single-hop network can be modeled by a complete graph. Our scatternet formation solutions build or apply some geometric structures on the complete graph. We apply the same scheme recently proposed by Li, Stojmenovic and Wang [1] for multi-hop networks. In case of multi-hop networks, these schemes require *exact position* information. Obtaining the precise positions currently poses challenging technological tasks [3] for short range Bluetooth devices, aimed primarily at home and office environments. However, we observe that, when the same scheme is applied to single-hop network, *virtual positions* (random position selected by each node independently and without any hardware requirements) are sufficient. The problem with virtual positions being applied in multi-hop networks is that two nodes which select virtual positions that are close to each other may physically be outside of each other's transmission range. On the other hand, in single-hop ad hoc networks, every node can communicate with each other directly, and the problem in multi-hop networks does not occur. Another advantage of using virtual positions for single-hop network is that our scatternet formation can be used for wireless nodes in three-dimensional space (such as a building) by just generating 2-dimensional virtual positions in a virtual plane.

In the solution proposed by Li, Stojmenovic, and Wang [1], nodes know their positions and are able to establish connections with other nodes within their transmission radius in the neighbor discovery phase. The next phase of the proposed formation algorithm is

optional, and can be applied to construct a sparse planar geometric structure. 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.

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 [4]. 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 [4], [5], while a protocol that provides two-hop information to every node is described in [6]. 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. Section III presents Bluetooth formation algorithms from [1], while Section [**?**] describes our new algorithms for single-hop ad hoc networks. Yao structure is applied on the complete graph (CG) or a sparse geometric structure, such as minimum spanning tree (MST), Gabriel graph (GG), relative neighborhood graph (RNG), Delaunay triangulation (DT) 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 clustering based scheme which includes adding two-element gateway piconets. We therefore obtain the Bluetooth scatternet formation algorithm for single-hop networks 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.

A preliminary conference version of this article appeared in [39].

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. For single-hop ad hoc networks, we assume that every nodes are in the transmission ranges of each other. We model a single-hop network as a complete graph $CG(S)$.

Let $disk(u, v)$ be the disk with diameter uv. Then, the *Gabriel graph* [7] $GG(S)$ contains edge uv 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). 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.

The *relative neighborhood graph* [8], denoted by $RNG(S)$, 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 Yao graph [9] 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 $\overrightarrow{uv}$. Ties are broken arbitrarily. The remaining edges are deleted from the graph. There are several variants on how this construction can be carried at each node in the graph. One choice is to carry it simultaneously on each node, with two options about keeping an edge uv: keep only if they mutually selected each other, or keep directional edges as well (one node selected other but not vice versa). The other choice (considered in this paper), is to carry this process sequentially, 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 *Delaunay triangulation*, denoted by $DT(S)$, if the circumcircle of each of its triangles does not contain any other vertices of S in its interior. A triangle is called the *Delaunay triangle* if its circumcircle is empty of vertices of S. Obviously, the Gabriel graph and the relative neighborhood graph are subgraphs of the Delaunay triangulation.

A subset of vertices in a graph G is a *dominating set* if all the vertices in G are either in this subset or neighbors of vertices in this subset. An example of a dominating set, which will be used in this paper, is the set of clusterhead nodes obtained in clustering scheme [10]. 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 [11]. An improved scheme is proposed in [12].

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 [13], [14] in order to define planar subgraph used for recovery routing to guarantee delivery, when simple heuristics fail. Gabriel graph was replaced in [16] 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). Relative neighborhood graph was used in [17] to provide efficient localized broadcasting for one-to-one models of wireless communications. Li *et al.* [18] 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. To improve the graph connectivity of planar graph, Li *et al.* [19] and [1] then proposed another two planar structures, localized Delaunay triangulation (LDT) and partial Delaunay triangulation (PDT), which can be constructed locally and efficiently. Both LDT and PDT contain Gabriel graph as their subgraph, and themselves are subgraphs of the Delaunay triangulation (DT). Notice that, since in single-hop networks every node knows all the information, we can construct DT directly instead of LDT or PDT which are used in the multi-hop case. Other references, applying geometric structures in wireless networks, are surveyed in [20].

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 and multi-hop solutions. In this paper, we only focus on designing scatternet formation algorithms for single-hop networks.

Zaruba, Basagni and Chlamtac [21] 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 [21] 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 [21]. Tan *et al.* [22] proposed a similar method, but are restricted to single-hop scenarios. In the second protocol [21], 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. Notice that the tree topology suffers from a major drawback: the root is a communication bottleneck as it will be overloaded by communications between the different parts of the tree.

Law, Mehta and Siu [23] 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 [24]. However, piconets are not degree limited in that scheme.

Salonidis *et al.* [4] 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 [4]. 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 [25].

Sun, Chang and Lai [26] 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. Bluerings as scatternets are proposed in [27]. Ring structure for Bluetooth has simplicity and easy creation as advantage, but it suffers large diameter (i.e., the maximum number of hops between any two devices) and large number of piconets.

Barriere, Fraigniaud, Narajanan, and Opatrny [28] described a connected degree limited and distributed scatternet formation solution based on projective geometry for single-hop networks. They assume that only slave nodes can act as bridges. They described procedures for adding and deleting nodes from the networks and claimed that it uses $O(\log^4 n \log^4 \log n)$ messages and $O(\log^2 n \log^2 \log n)$ time in local computation, where n is the number of nodes in the network. The degree of the scatternet can be fixed to any $q+1$, where q is a power of a prime number. However, in their method, every node need hold information of the projective plane and the master node who has the "token" needs to know the information of the projective scatternet (which label should be used for the new coming master and which existing nodes need to be connected to it). In [28], the authors did not discuss in detail how to compute the labels for the new master and its slaves, and what will happen when the number of nodes reaches the number of nodes of a complete projective scatternets.

III. Scatternet Formation Algorithms by Li, Stojmenovic and Wang

We now review the localized scatternet formation algorithms from [1], based on sparse geometrical structures. The algorithms have several phases which are shown in following algorithm.

Algorithm 1: Scatternet Formation Algorithms

1) Neighbor discovery and information exchange.
2) Planar subgraph construction (constructing RNG, GG, or PDT), if desirable.
3) Degree information exchange, if desirable.
4) 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.

A. Neighbor discovery and information exchange

Firstly, in the neighbor discovery phase, each node learns about its one-hop or two-hop neighbors. This procedure is called *inquiry procedure* in Bluetooth specifications. One-hop neighbor discovery can be performed by a scheme described in [4], [5]. 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. One modification needed for our application is that nodes exchange their positions in addition to their Bluetooth IDs, which is a trivial addition to the packet content.

B. Planar subgraph construction

This phase is optional. The remaining phases can be applied on the complete graph directly, but will result in non-planar graph. 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, 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. 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. Degree information exchange

This phase is also optional. In our methods, master-slave relations are decided based on a key. Two different keys can be considered. If node's Bluetooth ID is used as a key, this phase can be omitted. If the key is selected as the record (degree, ID), where node degree is primary key, and ID is secondary key, we need to collect degree information from neighbors. The procedure is basically the same procedure needed to collect two-hop information, the only difference again being the packet content. One such Bluetooth compatible procedure has been described in [6] and is applicable here. The idea is after knowing the local list of its neighbors a node can exchanges the degree with its neighbor. Though this phase needs to be done, if we use $degree$ in keys, the number of piconets will be reduced such that the scatternet is expected to function better. Therefore, we use this choice in the sequel and in our experiments.

D. 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 it has less than 7 neighbors. An edge remains in the structure if and only if both endpoints selected it, otherwise it is deleted from the structure. The process of applying Yao structure is done in an $iterative$ way. It works as follows.

Initially all nodes are undecided. In each iteration, undecided nodes with higher keys than any of their undecided neighbors (we shall refer to such nodes as *active*

nodes in the sequel) apply 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 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. 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 its decisions. 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 paragraph) 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. The difference is that communication can be restricted to edges remaining in the graph, so that the information exchange step is faster than neighbor discovery phase.

In 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 $(degree, ID)$, where degree is the number of its neighbors in the topology constructed. 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 $d*$, where $*$ 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. 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. 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 is unassigned. It notices its undecided neighbors to add a slave role. Such decision indicates that the node is creating a piconet.
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.

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. In next section, we will show an example with the detailed iterations of assigning roles. 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.

In [1], Li, Stojmenovic and Wang proved that the scatternet formed remains connected after the iterative application of Yao structure and assigning roles. We have extracted a connected sparse subgraph such that each node has degree at most 7. 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]. Recently, Basagni, Bruno and Petrioli [15] described some results of a ns2-based performance evaluation of our multi-hop scatternet formation method.

IV. Bluetooth Scatternet Formation for Single-hop Networks

In this paper, we adopt our multi-hop scatternet formation algorithms to single-hop ad hoc networks. Recall that, in a single-hop ad hoc network, all wireless devices are in the radio vicinity of each other, e.g., electronic devices in a laboratory, or laptops in a conference room. A single-hop network can be modeled by a complete graph. Our new scatternet creation solutions for single-hop networks apply the same schemes we described in last section for multi-hop networks. However, with the nice property of single-hop networks (each node knows all other nodes information), we can avoid the use of positions information in our algorithms. Also we can use some planar subgraphs of the complete graph, such as MST or DT, which can not be constructed locally in multi-hop networks.

A. Virtual positions

In case of multi-hop networks, these schemes require *exact position* information. Obtaining the precise positions currently poses challenging technological tasks [3] for short range Bluetooth devices. However, when the same schemes are applied to single-hop network, *virtual positions* (random position selected by each node independently and without any hardware requirements) are sufficient. The problem with virtual positions being applied in multi-hop networks is that two nodes which select virtual positions that are close to each other may physically be outside of each other's transmission range. However, in single-hop ad hoc networks, every node can communicate with each other directly, and the problem in multi-hop networks does not occur. Another advantage of using virtual positions for single-hop networks is that our scatternet formation can be used for wireless nodes in three-dimensional space (such as a building) by simply generating 2-dimensional virtual positions in a virtual plane. Thus, in our new scatternet creation solutions for single-hop networks, each node selects independently a random position in the neighbor discovery and information exchange phase.

B. Planar subgraphs

In the second (planar subgraph construction) phase, following methods [1] proposed for multi-hop networks, a planar subgraph is constructed locally and efficiently. Since in single-hop networks every node has all the information needed, we can apply some well-known

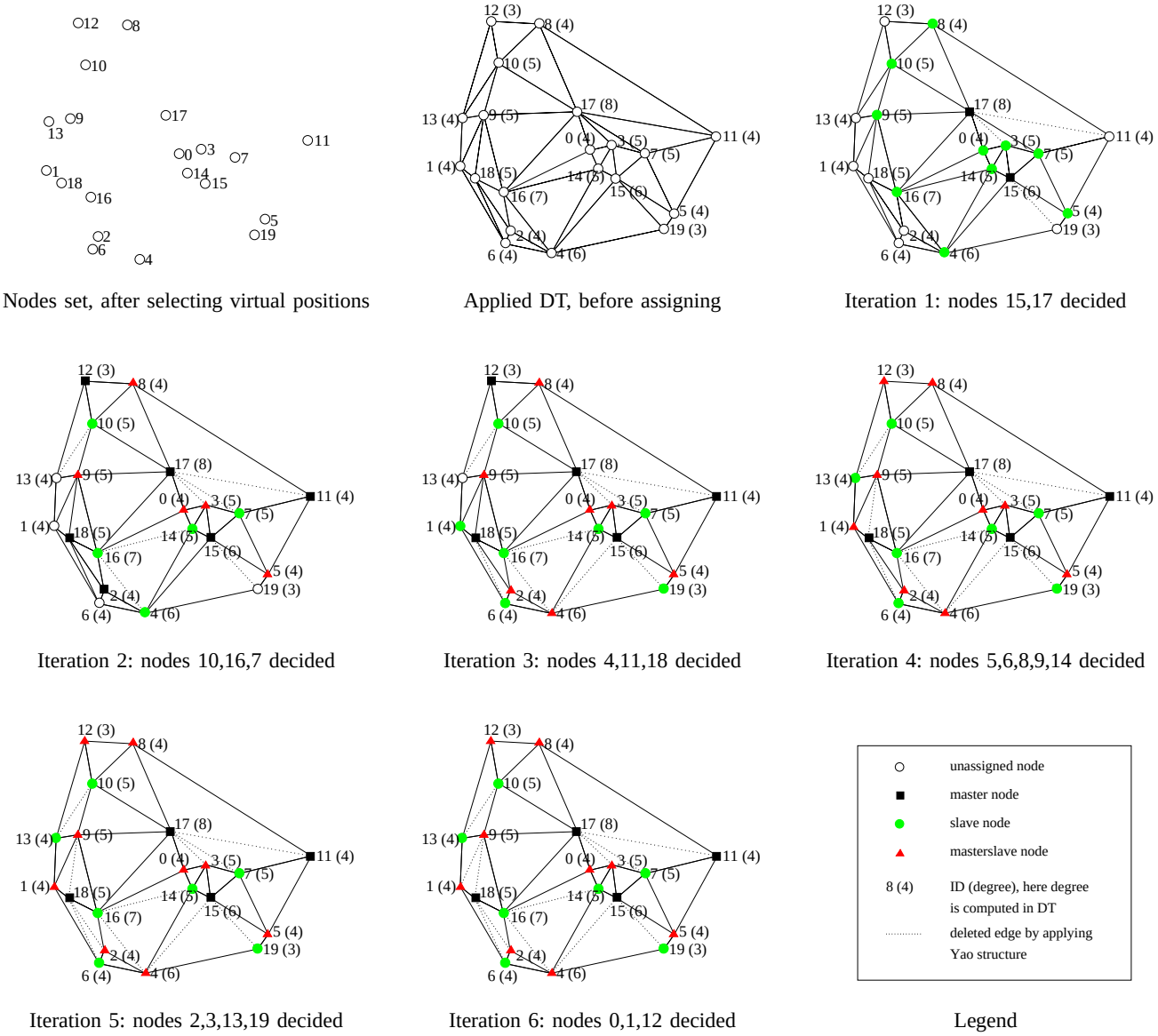

Fig. 1. Assigning roles in single-hop networks: six iterations.

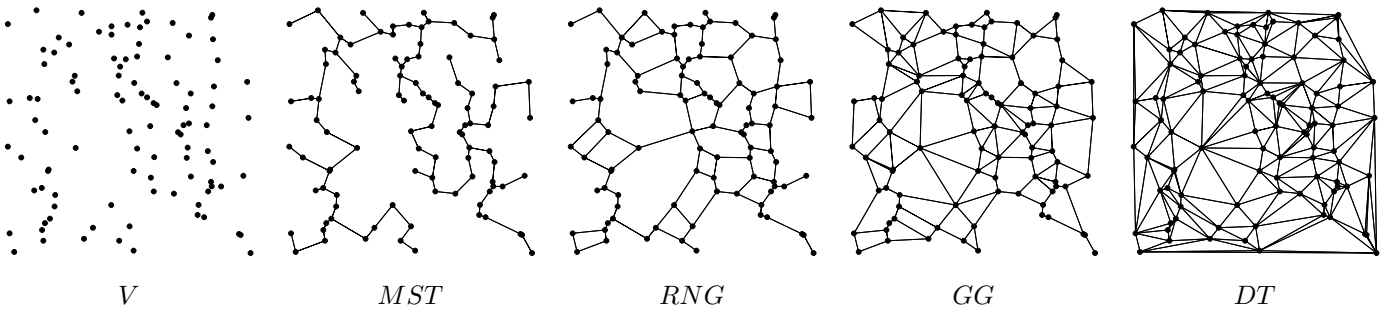

Fig. 2. Point set V and planar subgraphs of the complete graph on V.

global planar substructures in the second phase. For example, we can construct Delaunay triangulation (DT) directly instead of PDT. Notice that PDT is a subgraph of DT and DT is much denser than PDT. Thus, using DT may increase the delivery rate of routing methods on it. In addition, we can also use minimum spanning tree (MST) as the planar topology, which is suitable for broadcasting application. In summary, in the second phase, each node computes which of its incident edges belongs to chosen planar sparse structure, MST, RNG, GG, or DT. Note that each node can make local decisions (after completing neighbor discovery phase). Thus this construction only takes some computation cost without any communication cost.

C. An example

Figure 1 illustrates the procedure of our algorithms (applying planar structure, bounding degree, and assigning roles) in detailed iterations for an example single-hop network.

V. Experiments

In this section, we present our experimental results that compare designed algorithms in terms of various characteristics. We did not include other existing schemes for single-hop networks since each of them has deficiencies (with respect to our scheme) such as significant maintenance overhead, possible disconnection, possible excess number of roles, or lack of termination proof. In the experimental results presented here, we choose total $n = 100$ wireless nodes which are distributed randomly in a square area. Each node is specified by random X and Y coordinate values. These coordinates are virtual, used to mimic geographic position needed to establish Bluetooth scatternet structure. All results are the averages on total 20 wireless nodes sets.

A. Bluetooth Scatternet Formation

All nodes can be divided into several categories, according to the type and number of roles taken in the process. Thus a given node can be: (1) slave only, denoted by *S*, possibly to few piconets, this can be further divided as S_p, where p is the number of piconets where this slave node serves; (2) master only, denoted by M; (3) master of one piconet and slave in other piconets, denoted by MS or in general MS_p, where p is the number of piconets in which this node serves as slave.

Figure 3 illustrates the different Bluetooth structures using CG, MST, RNG, GG, or DT as topologies (shown in Figure 2), bounding degree by applying Yao structure, and assigning node roles by comparing end-nodes degrees 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 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 number of nodes (from 20 to 500). We find that the results are stable, i.e., the portion of the bridge nodes is stable. 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

TABLE I

NUMBER OF SLAVE NODES WITH p MASTERS.

graph	S_1	S_2	S_3	S_4	S_5	S_6	S_7	$S_{>7}$
dCG	0.45	3.70	10.80	7.35	1.60	0.00	0.00	0.00
gCG	0.00	1.45	5.70	7.85	6.70	3.30	0.95	0.00
dMST	23.70	20.35	0.00	0.00	0.00	0.00	0.00	0.00
gMST	7.05	23.65	9.20	0.45	0.00	0.00	0.00	0.00
dRNG	8.25	28.95	2.15	0.00	0.00	0.00	0.00	0.00
gRNG	2.30	19.35	16.55	1.05	0.00	0.00	0.00	0.00
dGG	2.10	13.70	13.20	0.90	0.00	0.00	0.00	0.00
gGG	0.60	7.25	13.95	10.40	2.10	0.15	0.00	0.00
dDT	0.45	5.95	14.15	5.55	0.25	0.00	0.00	0.00
gDT	0.00	2.55	8.50	10.55	5.70	1.50	0.05	0.00

TABLE II

NUMBER OF MS NODES WITH p MASTERS.

graph	M	MS_1	MS_2	MS_3	MS_4	MS_5	MS_6	MS_7	$MS_{>7}$
dCG	9.40	15.90	24.25	20.70	5.45	0.40	0.00	0.00	0.00
gCG	22.95	15.85	16.40	10.30	5.60	2.45	0.50	0.00	0.00
dMST	22.30	32.70	0.95	0.00	0.00	0.00	0.00	0.00	0.00
gMST	46.85	10.35	2.45	0.00	0.00	0.00	0.00	0.00	0.00
dRNG	21.90	30.65	8.10	0.00	0.00	0.00	0.00	0.00	0.00
gRNG	41.70	13.95	4.65	0.45	0.00	0.00	0.00	0.00	0.00
dGG	13.30	27.15	24.65	4.95	0.05	0.00	0.00	0.00	0.00
gGG	32.50	17.00	11.40	4.15	0.50	0.00	0.00	0.00	0.00
dDT	11.40	19.70	27.35	14.05	1.15	0.00	0.00	0.00	0.00
gDT	26.40	16.55	16.05	8.75	3.20	0.15	0.05	0.00	0.00

TABLE III

NUMBER OF PICONETS, BRIDGE NODES, AND SIZE OF PICONETS.

graph	master	slave	masterslave	avg M of S	avg M of MS	avg S of (M+MS)
dCG	9.40	23.90	66.70	3.25	2.25	3.00
gCG	22.95	25.95	51.10	4.29	2.29	3.08
dMST	22.30	44.05	33.65	1.46	1.03	1.77
gMST	46.85	40.35	12.80	2.08	1.20	1.66
dRNG	21.90	39.35	38.75	1.85	1.21	1.97
gRNG	41.70	39.25	19.05	2.42	1.29	1.97
dGG	13.30	29.90	56.80	2.43	1.61	2.34
gGG	32.50	34.45	33.05	3.19	1.64	2.51
dDT	11.40	26.35	62.25	2.97	1.95	2.71
gDT	26.40	28.85	44.75	3.84	1.99	2.80

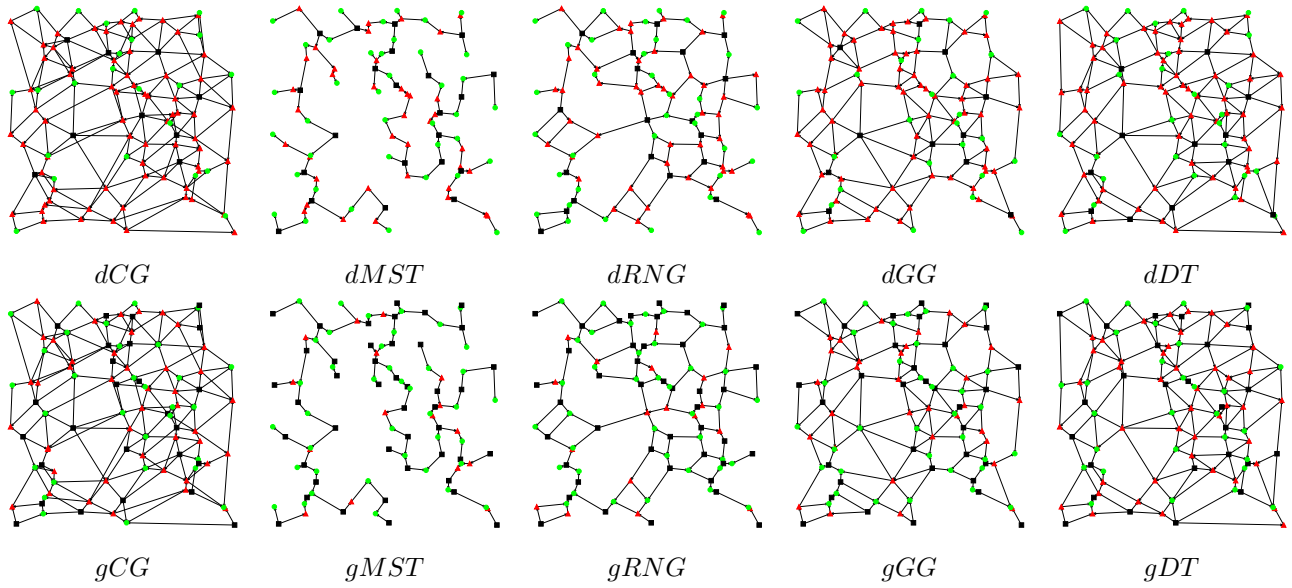

Fig. 3. Geometric structures, bounding node degree, and assigning roles.

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 produces fewer number of slaves to a node with master role. Moreover, it also generates less number of nodes with master-slave role than the other methods.

We found that the complete graph CG 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 (MST, GG, RNG, DT) perform at the same level in terms of the number of piconets generated and the number of piconets a slave node belonging to. We suggest to use DT since it has more edges than other three structures, thus, can sustain more link failures, and have shorter path for some pair of nodes. We also found that scatternets generated based on GG and DT are similar, due to the fact that DT has slightly more edges than GG.

B. Routing in Scatternet

Routing in Bluetooth received little attention so far. Bhagwat and Segall [30] 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 [31] 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). Barriere et al. [28] also proposed a routing method for Bluetooth scatternets formatted by their method using their specific labels. An important problem for scatternet formation algorithms 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. The routing problem in Bluetooth, however, is the last link in a chain

that starts with Bluetooth scatternet formation. Thus, it is interesting to see how our new structures perform in terms of routing efficiency, the quality of the selected routes and so on.

In this subsection, we study some well known geometric localized routing methods on the new structures. Localized routing is sometimes called in the literature *stateless* [32], *online* [33], or *distributed* [34]. Assume a packet is currently at node u, and the destination node is t. Several localized routing algorithms, i.e., find the next node v of u based on t and information of k-hop neighbors of node u, were developed.

- COMPASS ROUTING(CMP): The relay node v forms the smallest angle $\angle vut$ among all neighbors of u. See[35].
- RANDOM COMPASS ROUTING(RCMP): Let v_1 be the node above line ut such that $\angle v_1ut$ is the smallest among all such neighbors of u. Similarly, let v_2 be node below line ut that minimizes the angle $\angle v_2ut$. Then node u randomly choose v_1 or v_2 to forward the packet. See[35].
- GREEDY ROUTING(GRDY): Node u finds neighbor v closest to t as relay node. See [14].
- MOST FORWARDING ROUTING(MFR): Node u finds neighbor v such that $\|v't\|$ is the smallest as relay node, where v' is the projection of v on segment ut. See [34].
- NEAREST NEIGHBOR ROUTING(NN): Given a parameter angle α, node u finds the nearest node v as forwarding node among all neighbors of u such that $\angle vut \leq \alpha$.
- FARTHEST NEIGHBOR ROUTING(FN): Given a parameter angle α, node u finds the farthest node v as forwarding node among all neighbors of u such that $\angle vut \leq \alpha$.
- GREEDY-COMPASS(GCMP): Node u finds the neighbors v_1 and v_2 that forms the smallest clockwise and counter-clockwise angle respectively among all $N_1(u)$ with the segment ut. The packet is forwarded to the node of $\{v_1, v_2\}$ with the minimum distance to t. See [33], [36].

The compass routing, random compass routing and the greedy routing guarantee to deliver the packets if DT is used as network topology [14], [35], [36].

TABLE IV

THE DELIVERY RATE.

	sCG	sMST	sRNG	sGG	sDT
NN	83.8	10.5	33.8	63.3	80.3
FN	80.0	8.8	21.3	72.2	76.7
MFR	79.7	19.3	53.4	88.5	90.3
Cmp	76.6	4.2	18.9	46.0	65.5
RCmp	92.8	15.8	31.9	65.4	81.0
Grdy	100.0	31.3	68.8	100.0	100.0
GCmp	85.2	5.5	22.9	53.3	66.7

We then present our experiments of various routing methods on our different topologies. Again we choose 100 nodes distributed randomly in a square area. Figure 2 and Figure 3 illustrate the well known planar topologies and the final topologies after applying our method. We randomly select 20% of nodes as source; and for each source, we randomly choose 20% of nodes as destination. The statistics are computed over 10 different node sets.

Table IV illustrates the delivery rates. We use $s*$ to denote the bounded degree structures after applying Yao structure, where $*$ denotes the name of the sparse topology from the second phase. For routing methods NN and FN, we choose the next node within $\pi/3$ of the destination direction. Because sDT is denser than sMST, sGG and sRNG, the delivery rates of many routing

methods on it are higher. Recall that sCG is not a planar structure, while other three are. Since sMST, sGG and sRNG are planar graphs, we can apply right hand rule to improve delivery rate. More precisely, delivery can be even guaranteed following method described in [14] (subsequently completed by adding MAC layer in [32]), which applies the greedy routing on Gabriel graph and uses the right hand rule for recovery when greedy mode fails. Table V and Table VI illustrate the maximum and average spanning ratios of the path traversed by the packet from source s to destination t. We define spanning ratio of a path traversed by the packet from source s to destination t as follows: $spanning\ \ ratio = \frac{the\ \ total\ \ length\ \ of\ \ the\ \ path\ \ from\ \ s\ \ to\ \ t}{the\ \ distance\ \ between\ \ s\ \ and\ \ t,\ \ \|st\|}$. Note that the source and destination are within transmission range of each other in a single-hop network, thus ideally message can be delivered in one hop. We are investigating the theoretical reason why the spanning ratios of compass and random compass methods are so large. However, most of other routing methods have small spanning ratios on our topologies.

TABLE V

THE MAXIMUM SPANNING RATIO.

	sCG	sMST	sRNG	sGG	sDT
NN	1.6	1.3	1.6	1.6	1.6
FN	1.8	1.3	1.5	1.9	2.2
MFR	2.0	1.5	2.3	1.9	2.3
Cmp	11.6	1.0	2.1	5.5	11.2
RCmp	25.3	32.4	29.4	53.0	31.0
Grdy	1.5	3.2	2.0	1.8	1.5
GCmp	1.9	1.1	2.0	2.2	1.9

VI. CONCLUSION

We have described a scheme that creates connected degree limited scatternets for single-hop Bluetooth networks. A number of issues remain for future study.

TABLE VI

THE AVERAGE SPANNING RATIO.

	sCG	sMST	sRNG	sGG	sDT
NN	1.2	1.1	1.2	1.1	1.2
FN	1.3	1.1	1.2	1.3	1.3
MFR	1.3	1.1	1.4	1.2	1.3
Cmp	3.3	1.0	1.2	1.5	2.5
RCmp	5.1	5.4	5.6	6.3	5.2
Grdy	1.2	1.3	1.3	1.2	1.2
GCmp	1.3	1.0	1.2	1.5	1.4

One of major desirable properties of the proposed cluster based method 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 in multi-hop networks 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. In single-hop networks, we assume that nodes remain within transmission range of each other, therefore cluster update procedure is not called due to mobility. Nevertheless cluster maintenance is needed when nodes are added or removed from the network. 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. This approach has been investigated for multi-hop networks in [29].

Some other interesting problems include: fast schemes for the neighbors discovery, more suitable routing algorithms for the proposed scatternets, scheduling of

Bluetooth piconets, and capacity assignment based on expected traffic load (recently investigated in [38]).

REFERENCES

[1] X.Y. LI, I. STOJMENOVIC AND Y. WANG, Partial Delaunay Triangulation and Degree Limited Localized Bluetooth Multihop Scatternet Formation, *IEEE Transactions on Parallel and Distributed Systems*, Vol 15, 4, pp 350-361, 2004.

[2] BLUETOOTH SIG, Specification of the Bluetooth system, *http://www.bluetooth.com/*.

[3] J. HIGHTOWER AND G. BORRIELLO, Location systems for ubiquitous computing, *IEEE Computer*, Aug. 2001, 57-66.

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

[5] S. BASAGNI, R. BRUNO AND C. PETRIOLI, Device discovery in Bluetooth networks: A scatternet perspective, *Proc. IFIP-TC6 Networking Conf., Networking 2002,* Italy, May 2002, LNCS 2345.

[6] C. PETRIOLI AND S. BASAGNI, Degree-constrained multihop scatternet formation for Bluetooth networks, *Proc. IEEE GLOBECOM 2002,* Taipei, Taiwan, November 2002.

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

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

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

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

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

[12] J. WU AND W. LOU, Forward-node-set-based broadcast in clustered mobile ad hoc networks, Proc. GLOBECOM, 2002.

[13] L. BARRIERE, P FRAIGNIAUD, L. NARAJANAN AND J. OPATRNY, Robust position based routing in wireless ad hoc networks with unstable transmission ranges, *Proc. ACM DIALM01*, Rome, Italy, July 2001, 19-27.

[14] P. BOSE, P. MORIN, I. STOJMENOVIC AND J. URRUTIA, Routing with guaranteed delivery in ad hoc wireless networks, *Proc. ACM DIALM99*, Seattle, August 1999, 48-55; *ACM/Kluwer Wireless Networks*, 7, 6, November 2001, 609-616.

[15] S. BASAGNI, R. BRUNO AND C. PETRIOLI, A Performance Comparison of Scatternet Formation Protocols for Networks of Bluetooth Devices, *Proc. IEEE International Conference on Pervasive Computing and Communications (PerCom)* Texas, Mar. 2003.

[16] J. GAO, L.J. GUIBAS, J. HERSHBERGER, L. ZHANG AND A. ZHU, Geometric spanner for routing in mobile networks, *Proc. ACM Symposium on Mobile Ad Hoc Networking and Computing MobiHoc*, October 2001, 45-55.

[17] 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, 37-44.

[18] X.Y. LI, P.J. WAN, Y. WANG AND O. FRIEDER, Sparse power efficient topology for wireless networks, *Proc. Hawaii Int. Conf. on System Sciences*, 2002.

[19] X.Y. LI, G. CALINESCU, P.J. WAN AND Y. WANG, Localized Delaunay Triangulation with Application in Ad Hoc Wireless Networks, *IEEE Transactions on Parallel and Distributed Systems*, , vol. 14, no. 10, 2003, 1035-1047.

[20] X.Y. LI , Algorithmic, geometric and graphs issues in wireless networks, *Wireless Communications and Mobile Computing*, 3, 2, 2003, 119-140.

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

[22] G. TAN, A. MIU, J. GUTTAG AND H. BALAKRISHNAN, Forming scatternets from Bluetooth personal area networks, *MIT-LCS-TR-826*, October 2001; First Annual Student Oxygen Workshop, Gloucester, MA, July 2001.

[23] C. LAW, A.K. MEHTA AND K.Y. SIU, Performance of a new Bluetooth scatternet formation protocol, *Proc. ACM MobiHoc01*, October 2001, 183-192.

[24] S. BAATZ, S. BIESCHKE, M. FRANK, P. MARTINI, C. SCHOLZ AND C. KUHL, Building Efficient Bluetooth Scatternet Topologies from 1-Factors, *Proc. IASTED Wireless and Optical Communications WOC*, Banff, Canada, July 2002.

[25] D. MIORANDI AND A. ZANELLA, On the optimal topology of Bluetooth piconets: Roles swapping algorithms, *Proc. Mediterranean Conf. on Ad Hoc Networks*, Sardinia, Italy, Sept. 2002.

[26] M.T. SUN, C.K. CHANG AND T.H. LAI, A self-routing topology for Bluetooth scatternets, *Proc. ISPAN*, Philipini, May 2002.

[27] F. CGUN-CHOONG AND C. KEE-CHAING, Bluerings - Bluetooth scatternets with ring structure, *Proc. IASTED Wireless and Optical Communications WOC*, Banff, Canada, July 2002.

[28] L. BARRIERE, P FRAIGNIAUD, L. NARAJANAN AND J. OPATRNY, Dynamic construction of Bluetooth scatternets of fixed degree and low diameter, *Proc. ACM SODA*, 2003.

[29] I. STOJMENOVIC, Dominating set based Bluetooth scatternet formation with localized maintenance, CD Proc. IEEE IPDPS Workshops, Fort Lauderdale, April 2002.

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

[31] B.J. PRABHU AND A. CHOCKALINGAM, A routing protocol and energy efficent techniques in Bluetooth scatternets, *Proc. IEEE ICC 2002* .

[32] B. KARP AND H. T. KUNG, "GPRS: Greedy perimeter stateless routing for wireless networks," in *ACM/IEEE International Conference on Mobile Computing and Networking*, 2000.

[33] P. BOSE AND P. MORIN, "Online routing in triangulations," in *Proc. of the 10 th Annual Int. Symp. on Algorithms and Computation ISAAC*, 1999.

[34] IVAN STOJMENOVIC AND XU LIN, "Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 10, 2001.

[35] E. KRANAKIS, H. SINGH, AND J. URRUTIA, "Compass routing on geometric networks," in *Proc. 11 th Canadian Conference on Computational Geometry*, 1999, pp. 51–54.

[36] P. MORIN, *Online routing in geometric graphs*, Ph.D. thesis, Carleton University School of Computer Science, 2001.

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

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

[39] Y. WANG, I. STOJMENOVIC AND X.Y. LI, Bluetooth Scatternet Formation for Single-hop Ad Hoc Networks Based on Virtual Positions, in *Proc. 9th IEEE Symposium on Computers and Communications ISCC'2004*, Alexandria, Egypt, Jun. 2004.