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

Yu Wang
Dept. of Computer Science
Illinois Institute of Technology
Chicago, USA
wangyu1@iit.edu

Ivan Stojmenovic
Schl. of Info. Tech. & Engr.
University of Ottawa
Ottawa, Canada
ivan@site.uottawa.ca

Xiang-Yang Li
Dept. of Computer Science
Illinois Institute of Technology
Chicago, USA
xli@cs.iit.edu

Abstract

This paper addresses the problem of scatternet formation for single-hop Bluetooth based personal area and ad hoc networks, with minimal communication overhead. 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. In this article we apply their methods 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 in [1] using such virtual positions instead of real ones. In addition, we use Delaunay triangulation instead of partial Delaunay triangulation proposed in [1], since each node has all the information needed. 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.

1 Introduction

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. 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. The standard also allows multiple roles for the same device. A node can be a 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 should minimize 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 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 *et al.* [1] for multi-hop networks. In multi-hop networks, these schemes require *exact position* information. Obtaining the precise positions currently poses challenging technological tasks 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 networks, *virtual positions* (random positions selected by each node independently) are sufficient. The problem with applying virtual positions in multi-hop networks is that two nodes with close virtual positions may physically be outside of each other's transmission range. One advantage of using virtual positions is that our scatternet formation can be used for wireless nodes in three-dimensional space (such as a building).

Previous literature on scatternet formation assumed that devices cannot communicate unless they have previously discovered each other by synchronizing their frequency hopping patterns. Synchronizing the frequency hopping patterns is apparently a time consuming and pseudo-random process [2]. In this paper we assume that the problem of discovering all neighbors in transmission radius is resolved by separate Bluetooth protocols such as those in [2, 3].

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. Section 3 presents Bluetooth formation algorithms from [1], while Section 4 describes our new algorithms for single-hop networks. Section 5 describes the experimental results of our algorithms. Section 6 concludes our paper.

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. 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 $CG(S)$.

The *Gabriel graph* $GG(S)$ contains edge uv iff the disk with diameter uv contains no other points of S. The *relative neighborhood graph* $RNG(S)$ 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. $GG(S)$ and RNG are planar graphs (that is, no two edges cross each other). Obviously, they can be constructed locally. In other words, a node can compute its incident edges in $GG(S)$ or $RNG(s)$ by using only 1-hop neighbors. Both $GG(S)$ and $RNG(S)$ are connected and contain the Euclidean MST of S.

The Yao graph [4] 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, only keep the link $\overrightarrow{uv}$ to the closest node v within the transmission range of u, if there is any. The remaining edges are deleted from the graph.

We continue with the definition of the Delaunay triangulation. 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. Obviously, $GG(S)$ and $RNG(S)$ are subgraphs of $DT(S)$.

Sparse geometric structures that can be defined locally have been applied in wireless networks for localized routing and broadcasting algorithms. GG was used in [5, 6] in order to define planar subgraph used for recovery routing to guarantee delivery, when simple heuristics fail. RNG was used in [8] to provide efficient localized broadcasting for one-to-one models of wireless communications. Li *et al.* [9] proposed to use GGs, RNGs, and Yao graphs to construct sparse power efficient networks. To improve the graph connectivity of planar graphs, Li *et al.* [10] 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 GG as their subgraph, and themselves are subgraphs of 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.

2.2 Literature Review on Scatternet Formation

Scatternet formation algorithms 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.

In [11, 12, 13], authors proposed several algorithms that create connected degree bounded scatternets. The final structures are all tree like topologies, which limit efficiency and robustness. Note 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. A single-hop Bluetooth scatternet formation scheme based on 1-factors is described in [14]. However, piconets are not degree limited in that scheme. Recently, Salonidis *et al.* [2] proposed a centralized topology construction algorithm which is not scalable and not localized. Moreover, how to assign the roles is not elaborated in [2]. 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 [15]. Sun *et. al* [16] 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. It relies on a sophisticated scatternet merge procedure with significant communication overhead for creation and maintenance. Bluerings as scatternets are proposed in [17]. 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 *et al.* [18]

described a connected degree limited and distributed scatternet formation solution based on projective geometry for single-hop networks. However, in their method, every node need hold information of the projective plane and the master node with "token" needs to know the information of the projective scatternet. In [18], they 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.

3 Scatternet Formation Algorithms in [1]

We now review the localized scatternet formation algorithms from [1], which have several phases as follows.

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 (called *active node*), 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.

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. The second and third phases are optional. If planarity is a desirable property in some cases, e.g., routing with guaranteed delivery, each node computes which of its incident edges belongs to chosen planar sparse structure. If node degree is used as part of the key when assigning master-slave role, we need to collect degree information from neighbors in the third phase. In the last (mandatory) phase, the degree of each node is limited to 7 by applying Yao structure in an *iterative* way, 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. In each iteration, active nodes decide master-slave roles at each undeleted edge, and communicate the decision to the other node at each edge. In [1], they describe two different ways to decide the roles: node with initially higher key

(ID or both node degree and ID, in our experiments we use the latter) is master, and cluster based. Both methods keep all links "saved" by Yao structure but converts them to directed edges, so that one node on each edge is master, and the other is slave. In [1], Li *et. al* proved that the scatternet formed remains connected after the iterative application of Yao structure and assigning roles, and 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. Recently, Basagni *et. al* [7] described results of a ns2-based performance evaluation of our multi-hop scatternet formation method.

4 New Bluetooth Scatternet Formation for Single-hop Ad Hoc 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. A single-hop network is modeled by a complete graph here. 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.

4.1 Virtual Positions

In case of multi-hop networks, these schemes require *exact position* information. Obtaining the precise positions currently poses challenging technological tasks 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 is that our scatternet formation can be used for wireless nodes in three-dimensional space 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.
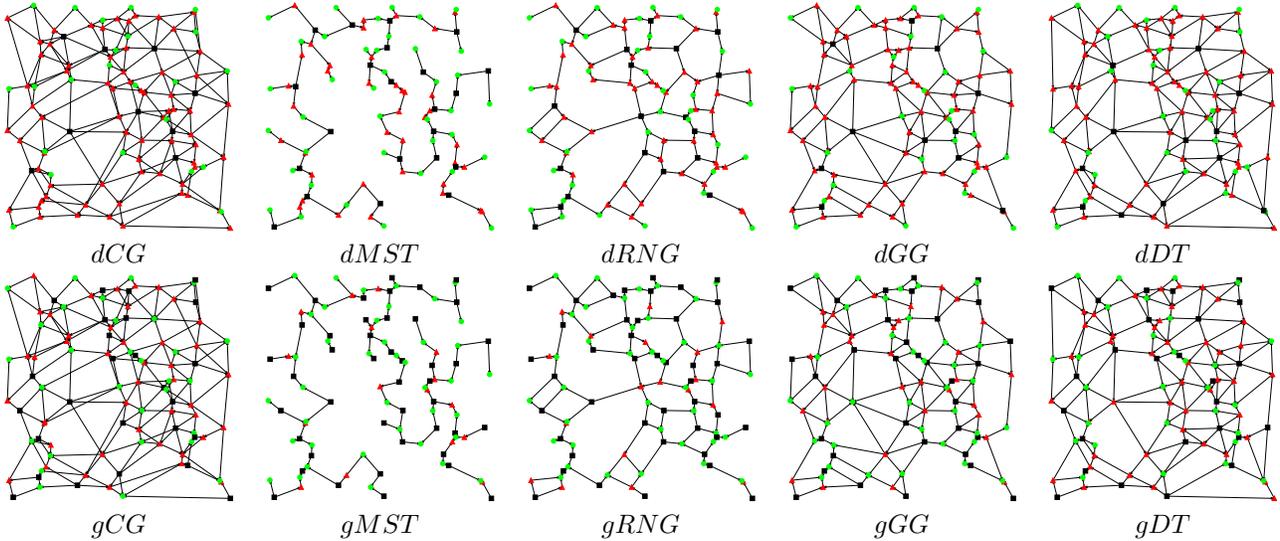
Figure 1. Geometric structures, bounding node degree, and assigning roles.

4.2 Planar Subgraphs

In the second (planar subgraph construction) phase, following methods [1] proposed for multi-hop networks, a planar subgraph is constructed locally. Since in single-hop networks every node has all the information needed, we can apply some well-known 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 the construction only takes some computation cost without any communication cost.

5 Experiments

In this section, we present our experimental results that compare designed algorithms in terms of various characteristics. We do 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, or possible excess number of roles. In the experiments, we choose 100 wireless nodes randomly distributed in a square area. The coordinates of each node are virtual, used to mimic geographic position needed to establish Bluetooth scatternet. All results are the averages on total 20 wireless nodes sets.

5.1 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 (also called *bridge*), denoted by MS or in general MS_p, where p is the number of piconets in which this node serves as slave.

Figure 1 illustrates the different Bluetooth structures using CG, MST, RNG, GG, or DT as topologies, 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*). Hereafter, $*$ denotes the name of the sparse topology from the second phase. 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 1 lists the number of slave/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 2 presents the average number of slave nodes assigned to a node with master role, i.e., a master node or a master-slave node. The fifth column represents the average number of piconets assigned to a node with slave roles only. The sixth column represents the average number of piconets

Table 1. Number of master(M) nodes and number of slave(S)/master-slave(MS) nodes with p masters.

graph	M	S_1/MS_1	S_2/MS_2	S_3/MS_3	S_4/MS_4	S_5/MS_5	S_6/MS_6	S_7/MS_7	$S_{>7}/MS_{>7}$
dCG	9.40	0.45/15.9	3.70/24.2	10.8/20.7	7.35/5.45	1.60/0.40	0.00/0.00	0.00/0.00	0.00/0.00
gCG	22.9	0.00/15.8	1.45/16.4	5.70/10.3	7.85/5.60	6.70/2.45	3.30/0.50	0.95/0.00	0.00/0.00
dMST	22.3	23.7/32.7	20.3/0.95	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
gMST	46.8	7.05/10.3	23.6/2.45	9.20/0.00	0.45/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
dRNG	21.9	8.25/30.6	28.9/8.10	2.15/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
gRNG	41.7	2.30/13.9	19.3/4.65	16.5/0.45	1.05/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
dGG	13.3	2.10/27.1	13.7/24.6	13.2/4.95	0.90/0.05	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
gGG	32.5	0.60/17.0	7.25/11.4	13.9/4.15	10.4/0.50	2.10/0.00	0.15/0.00	0.00/0.00	0.00/0.00
dDT	11.4	0.45/19.7	5.95/27.3	14.1/14.0	5.55/1.15	0.25/0.00	0.00/0.00	0.00/0.00	0.00/0.00
gDT	26.4	0.00/16.5	2.55/16.0	8.50/8.75	10.5/3.20	5.70/0.15	1.50/0.05	0.05/0.00	0.00/0.00

Table 2. 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/gCG	9.40/22.95	23.90/25.95	66.70/51.10	3.25/4.29	2.25/2.29	3.00/3.08
dMST/gMST	22.30/46.85	44.05/40.35	33.65/12.80	1.46/2.08	1.03/1.20	1.77/1.66
dRNG/gRNG	21.90/41.70	39.35/39.25	38.75/19.05	1.85/2.42	1.21/1.29	1.97/1.97
dGG/gGG	13.30/32.50	29.90/34.45	56.80/33.05	2.43/2.97	1.61/1.95	2.34/2.71
dDT/gDT	11.40/26.40	26.35/28.85	62.25/44.75	2.97/3.84	1.95/1.99	2.71/2.80

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 method.

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.

5.2 Routing in Scatternet

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. Most designed structures are planar and therefore suitable for routing with guaranteed delivery [6], 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 effi-

ciency, 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. 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 [6, 19, 20]. In our experiments, we test the following routing algorithms: compass routing(Cmp), random compass routing(RCmp), greedy routing(Grdy), most forwarding routing(MFR), nearest neighbor routing(NN), farthest neighbor routing(FN), and greedy-compass routing(GCmp). The compass routing, random compass routing and the greedy routing guarantee to deliver the packets if DT is used as the underlying network topology [6, 19, 20]. Again, we choose 100 nodes distributed randomly in a square area. Figure 1 illustrates 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 3 illustrates the delivery rates. We use $s*$ to denote the bounded degree structures after applying Yao structure. 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, sRNG and sDT are planar graphs, we can apply right hand rule to improve delivery rate. More pre-

Table 3. 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

Table 4. Maximum/average spanning ratio.

	sCG	sMST	sRNG	sGG	sDT
NN	1.6/1.2	1.3/1.1	1.6/1.2	1.6/1.1	1.6/1.2
FN	1.8/1.3	1.3/1.1	1.5/1.2	1.9/1.3	2.2/1.3
MFR	2.0/1.3	1.5/1.1	2.3/1.4	1.9/1.2	2.3/1.3
Cmp	12/3.3	1.0/1.0	2.1/1.2	5.5/1.5	11/2.5
RCmp	25/5.1	32/5.4	29/5.6	53/6.3	31/5.2
Grdy	1.5/1.2	3.2/1.3	2.0/1.3	1.8/1.2	1.5/1.2
GCmp	1.9/1.3	1.1/1.0	2.0/1.2	2.2/1.5	1.9/1.4

cisely, delivery can be even guaranteed following method described in [6]. Table 4 illustrates the maximum and average spanning ratios of the path traversed by the packet from s to t. We define spanning ratio of a path traversed by the packet from s to 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 Cmp and RCmp methods are so large. However, most of other routing methods have small spanning ratios on our topologies.

6 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, such as efficient cluster algorithms for dynamic networks, fast schemes for neighbors discovery, more suitable routing algorithms for the proposed scatternets, scheduling of Bluetooth piconets, and capacity assignment based on expected traffic load.

References

[1] X.Y. LI, I. STOJMENOVIC AND Y. WANG, Partial Delaunay triangulation and degree limited localized Bluetooth multihop scatternet formation, *IEEE TPDS*, Vol 15, 4, pp 350-361, 2004.

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

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

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

[5] 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.

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

[7] S. BASAGNI, R. BRUNO AND C. PETRIOLI, A performance comparison of scatternet formation protocols for networks of Bluetooth devices, *Proc. IEEE PerCom* Texas, 2003.

[8] 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.

[9] 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.

[10] X.Y. LI, G. CALINESCU, P.J. WAN AND Y. WANG, Localized Delaunay triangulation with application in ad hoc wireless networks, *IEEE TPDS*, vol. 14, no. 10, 2003, 1035-1047.

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

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

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

[14] 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.

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

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

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

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

[19] E. KRANAKIS, H. SINGH, AND J. URRUTIA, "Compass routing on geometric networks," in *Proc. CCCG*, 1999.

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