Hierarchical Self-routing Scatternet for Multihop Bluetooth Networks

Wen-Zhan Song [†] Xiang-Yang Li [*]

Abstract— **The paper proposes a strategy for each Bluetooth device selecting proper communication neighbors and assigning proper label, hence all nodes together form a hierarchical self-routing Bluetooth networks. Both the scatternet formation and routing protocols do not require any geometric information, and the final network topology has the following attractive properties: (1) the diameter of the scatternet is $O(\log(n))$ and the backbone is a hop spanner; (2) the degree of each master node is bounded by a constant 7; (3) the number of piconets is close to optimal; (4) each cluster has self-routing property. Moreover, the network topology can be maintained dynamically and locally with low communication cost, and the message delivery is guaranteed even during structure updating in clusters. The network supports efficient IP-based routing through Distributed Hash Tables(DHTs). The actual routing performance on the structure is evaluated through extensive simulations, the result shows the average communication hops are indeed around $\log(n)$.**

Keywords— **Bluetooth networks, scatternet formation, multi-hop, de Bruijn, connected dominating set, low diameter, hop spanner, degree-bounded,**

I. Introduction

Bluetooth [1] is a promising low cost and low power wireless technology, which enables portable devices to form short-range wireless ad hoc networks based on a frequency hopping physical layer. Bluetooth operates in the unlicensed 2.4GHz ISM band, with the frequency hopping technique to alleviate the effects of the interference. The nominal bit rate of transmission is 1Mbps. It has been widely predicted that Bluetooth will be the major technology for short range wireless networks and wireless personal area networks. Bluetooth ad-hoc networking also presents new technical challenges, such as scheduling, network forming and routing. According to the Bluetooth specification, 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*. 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. Several criteria are known as the objectives in forming scatternet. First of all, the protocol should create degree limited scatternets, to avoid parking any node. Secondly, the formation and maintenance of scatternet should have small communication overhead. Thirdly, the diameter of the scatternet should be small, i.e., the maximum number of hops between any two devices must be small.

Many scatternet formation algorithms have been proposed recently, and all of them can only meet part of the preferred criteria. A greedy centralized multi-hop algorithm, where a hypothetical central entity knows the complete topology has been proposed in [13]. Distributed algorithms have also been proposed in [13], which assume 2-hop neighborhood information. They applied a variant of clustering algorithm with limiting number of nodes in each cluster to seven, in accordance to Bluetooth restriction. However, there are examples where the scatternet is disconnected, which may occur when two cluster-heads were originally connected but formed clusters and 'erased' their link without leaving alternate connection between their piconets. References [16], [17] essentially propose variants of clustering based scatternet formation scheme, where clustering process are made at random. However, it also does not always lead to connected structure. The counterexample is the same that applies to [13]. On a positive side, [16] proposes two excellent measures for the performance of scatternets: average shortest-path length and maximum traffic flow. Zaruba, Basagni and Chlamtac [14] 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. 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 [14] observed that if a node in unit disk graph has more than five neighbors, then at least two of them must be connected . Tan *et al.* [15] proposed a similar method, but are restricted to single-hop scenarios. Basagni and Petrioli [18], [4] described multi-hop scatternet formation scheme, taking into account several Bluetooth issues which do not pertain to clustering. Clusterhead (master role) decisions are based on node weights (instead of node IDs, which express their suitability to become masters), following a variant of clustering method described in [19]. All clusterhead nodes are declared master nodes in a piconet, with all nodes belonging to their clusters as their slaves. Some of the slaves become masters of additional piconets, following [6], to assure connectivity. However, piconets may have more than seven slaves. The scheme described by Petrioli and Basagni [20] does not require position information, but instead the local information is extended to two-hop information, with two rounds device discovery phase for obtaining necessary information. Scatternet formation proceeds in iterations. However, the method may show weaknesses on some other metrics, especially about the worst case number of slave roles a node can assume.

[†] School of Engineering and Computer Science, Washington State University, Vancouver WA 98686. Email: song@vancouver.wsu.edu

[*] Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616. Email: xli@cs.iit.edu

This paper proposes both the scatternet formation and routing algorithms for multi-hop Bluetooth networks. We first propose a novel communication efficient method to build a *connected dominating set* (CDS) as the backbone of multi-hop Bluetooth network, then construct the a scalable scatternet in each cluster based on de Bruijn graph, which makes self-routing within the cluster possible. A cluster is defined by a dominator node and all its dominatee nodes. Several methods are proposed to form piconets in and between clusters to meet the preferred criteria. We then proposes an efficient IP-based routing protocol for the multihop network through Distributed Hash Tables (DHTs). The actual routing performance on the structure is evaluated through extensive simulations, where the average communication hops are indeed around $\log(n)$.

The rest of the paper is organized as follows. II, we describe our innovative multihop scatternet formation algorithm, which integrates the CDS based backbone and de Bruijn based cluster structure together seamlessly, and hence enjoys many nice properties. In Section III, we discuss in detail the IP-based routing solution based on DHTs. We evaluate our structures in Section IV by conducting extensive simulations. Finally, we conclude our paper in Section V.

II. Scatternet Formation and Labelling

Our scatternet formation algorithm has two phases: first, all nodes together elect proper nodes to form the *connected dominating set* (CDS) as the backbone of the scatternet; then, each cluster self-forms an efficient routing structure and assigns address to each node distributively. For simplicity, we omit the operation details and describe our scatternet formation algorithms conceptually.

A. Scatternet Backbone Formation

A subset S of V is a *dominating set* if any node u in V is either in S or is adjacent to some node in S. Nodes in S are called dominators, while nodes in $V - S$ are called dominatees. Once some nodes, hereafter called *connectors*, are selected to form a connected graph together with dominators, the final structure is called *connected dominating set*. Wan *et al.* [7], [5] proposed a communication efficient algorithm to find a set of dominators with the following property: the backbone by connecting each pair of dominators separated by two hops is connected. Their method uses a carefully chosen rank definition. The ranking of nodes is induced by an arbitrary spanning tree T rooted at a leader. The message complexity of their method is $O(n)$ if a leader is already known and $O(n \log n)$ if leader election is needed. Unlike previous CDS construction methods[1], it guarantees the network connectivity if all 2-hop adjacent dominators are connected. However, in their method, two adjacent dominators are not necessarily connected by a connector. In the worst case, two adjacent dominators may be connected by a path with $O(k)$ hops, where k is the number of dominators found. In other words, the connected dominating set derived from the spanning tree is not a hop-spanner. To facilitate the inter-cluster routing, in our algorithm, we adopt their method to find dominators. But, for *any* pair of 2-hop adjacent dominators, we find a dominatee node to connect them to generate a hop-spanner as backbone. We try to minimize the number of connectors and communication cost during construction while keeping at least one connector for each pair of adjacent dominators. Thus, a connector could be used to connect many pairs of dominators in our method. In all previous methods [7], [5], [9] to find connectors, they adopt the broadcast communication model to build CDS graph. It is well known that local broadcast cannot be performed efficiently in practice, due to the constraint in MAC layer. Simultaneous broadcast by dominatees could cause massive signal interference so that large latency is unavoidable. In our algorithm, we actually reduce the communication cost significantly by using unicast instead of broadcast.

[1]Previous methods need connect 3-hop adjacent dominators in some situation to ensure the connectivity.

In Algorithm 1, each dominator maintains two lists: *adjacentDominatorList* and *dominateeList*, which are initially empty. Here *adjacentDominatorList* records all adjacent dominators of this node, in addition, the connection flag is set for each pair of adjacent dominators acknowledged by a connector; *dominateeList* records all dominatees dominated by this node, which is reserved for dBBlue scatternet construction as will see later. Each dominatee also maintains two lists: *blackList* and *neighborDominatorList*. *blackList* is initialized as the list of known dominator neighbors, and *neighborDominatorList* stores the dominator neighbors which need be connected by itself, if this node is a connector.

Algorithm 1: Finding Connectors

1. Each dominatee selects the neighboring dominator with the smallest ID in *blackList* and sends it a TRYCONNECTOR message, which includes *blackList*.
2. Once a dominator gets the TRYCONNECTOR message from a dominatee node, it first adds the sender to its *dominateeList*, then performs the following two steps:
   (a) Adds those unknown adjacent dominators (if exist) in *blackList* of the TRYCONNECTOR message into *adjacentDominatorList*.
   (b) Sets connection flag for each new pair of adjacent dominators (if exist) acknowledged by the sender. Simultaneously, generates a *confirmList* including all new pairs of adjacent dominators that will require this dominatee node to connect.
   Finally, if *confirmList* is non-empty, it confirms the sender with a CONFIRMCONNECTOR message, which includes the *confirmList*.
3. Once a dominatee gets the CONFIRMCONNECTOR message, it will copy *confirmList* to its *neighborDominatorList*, and announces itself as connector by sending all dominators (except the sender) in the *confirmList* a IAMCONNECTOR message including the *neighborDominatorList*.
4. Once a dominator gets the IAMCONNECTOR message from a connector, it simply adds all unknown adjacent dominators in the *neighborDominatorList* of the message to its *adjacentDominatorList* and sets flag for each unknown adjacent pairs if necessary.

It is easy to show that each 2-hop adjacent dominator pair is finally connected, since the connection will be acknowledged by some dominatee in our algorithm. Figure 1(a) illustrates a backbone topology formed by our algorithm, in which each adjacent dominator pair is connected by exactly one connector. One

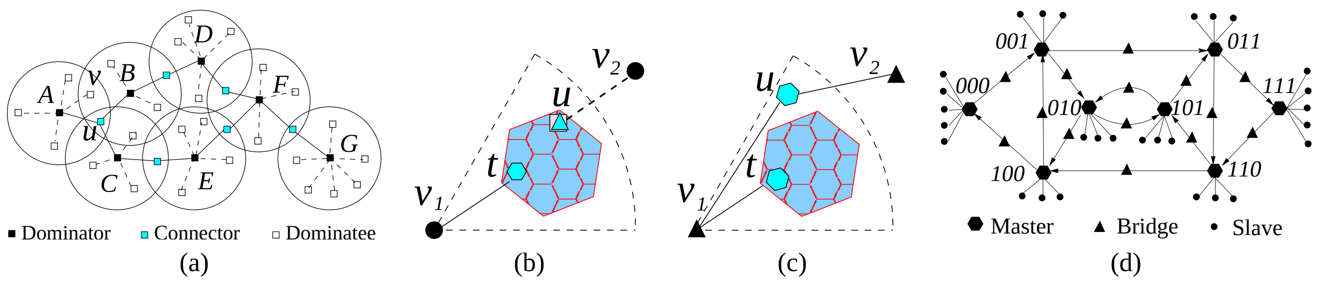


Fig. 1. Illustration of Scatternet Formation Procedure (a) Algorithm 1 (b) Algorithm 2 (c) Algorithm 3 (d) dBBlue Scatternet

connector could be used to connect several dominators. For instance, node u serves as the connector for three dominators A, B and C, hence node v will not be selected as dominators. Using CDS as the backbone of Bluetooth networks is not our innovation, priori arts [18], [4] also adopted CDS forming backbone. But, the CDS formation algorithms presented here has more nice properties than all previous approaches. Firstly, the algorithm for finding connectors uses unicasting instead of broadcasting, which is more communication efficient. Secondly, the backbone is a hop spanner, any 2-hop adjacent dominator pairs are connected through a connector, while connectors only connect with dominators. In [18], [4], after the backbone is formed, all dominatees are connected to dominators directly, hence the node degree of dominators could be very large, such as $O(n)$ in worst case.

B. Scatternet Formation in Clusters

As mentioned before, a cluster may have many nodes, obviously it is inefficient to connect all dominatees to the dominator. The node degree is preferred to be bounded by a constant number, where *seven* is the best match to Bluetooth specifications. One-hop Bluetooth scatternet formation has been well studied in [10], [2], [3], [11]. The dBBlue protocol proposed in [2] enjoys many nice properties such as low-diameter, single-role, bounded-degree and self-routing. In addition, the scatternet can be easily updated due to a scalable MAC assignment mechanism. In [2], Song *et. al* adopt the well-known de Bruijn graph to build a self-routing scatternet with low-diameter $O(\log n)$ and bounded node-degrees. Each master has at most seven slaves and each slave node exists in at most two piconets, and no node assumes both master and slave roles. They also presented a scalable MAC assignment mechanism and a vigorous method to *locally* update the structure *dBBlue* using at most $O(\log n)$ communications when a node joins or leaves the network. The computation cost is $O(n)$ for static construction. Figure 1(d) illustrates a dBBlue scatternet containing 48 nodes based on $B(2,3)$ graph. The dBBlue protocol only works for one-hop Bluetooth network.

Unfortunately, each cluster (composed of a dominator node and all its dominatees) may be *not* an one-hop Bluetooth network, i.e., some dominatees pair could not communicate directly at all. Thus, any one-hop scatternet formation algorithm can not be applied here directly. There are two possible solutions here: (1) make sure that all dominatees of a dominator node can communicate directly. (2) partition the dominatee nodes into cliques such that the dominatee nodes of each clique can communicate directly. Figure 1(b)(c) illustrates the proposed two algorithms to applying the dBBlue scatternet formation protocol in each clique. In both algorithms, we will let the dominator nodes be slaves(or bridges) of piconets. Notice that this approach is different from all previous scatternet formation methods based on the connected dominating set, in which the dominator is naturally assigned master role instead. We will show that assigning dominator slave roles actually produces scatternet with several nice properties.

Algorithm 2: Cluster Scatternet Formation Method 1
The dominator divides each cluster into 6 cliques. In each clique,
1. The dominator randomly selects a *dominatee* as leader, which initiates the dBBlue scatternet formation on all nodes in the clique including the *dominator* and *connectors*.
2. The dominator always assumes the pure slave of the leader with MAC 100.
3. All connectors have higher priority to be pure slaves or bridge slaves than other dominatees. Notice that the average number of connectors in each cone is $24/6 = 4$ and the leader could have up to 7 slave nodes. Thus, a connector will assume the slave role of the leader with high probability.

Figure 1(b) illustrates the algorithm. Notice that we made some special treatments with the dominator and connectors in the algorithm, because we need diminish the probability to assign them *master* roles in the dBBlue structure. Consequently, their node degree can be bounded by 7 and they do not need switch roles between *bridge* and *master*. Though we can not avoid the role switch absolutely, it is not difficult to show that, at most one connector need assume dual roles, even in the worst situation that there are no dominatees in the clique, which rarely happens in practice as will see in our simulation results,

Theorem 1: In the scatternet built by Algorithm 2, node degree is bounded by a constant 7 with high probability.

Proof: There are three kinds of nodes in the multi-hop scatternet: dominator, dominatee and connector. We consider them case by case.

Case 1: a dominator node. According to Algorithm 2, the dominator assumes pure slave in the one-hop dBBlue scatternet and there are at most 6 dBBlue structures in a cluster, so its degree is bounded by 6.

Case 2: a connector node. We notice that the connector could exist in at most 5 clusters, as long as there is at least one dominatee node in a clique. We can let that the dominatee assume *master* role and the connector be its pure slave. Hence its degree is bounded by 5.

Case 3: a dominatee node. Its node degree is obviously bounded by 7 according to the property of dBBlue scatternet. ■

Another approach is to exclude the connectors from participating in the one-hop dBBlue scatternet formation in each clique, so that *no* nodes need assume both master and slave roles. Figure 1(c) illustrates the algorithm which is described in detail as follows.

Algorithm 3: Cluster Scatternet Formation Method 2
The dominator divides each cluster into 6 cliques. In each clique,

1. The dominator assumes *slave* role and the connectors assumes *master* role in the CDS-based backbone.
2. The dominator randomly selects a *dominatee* as leader, which initiates the dBBlue scatternet formation on all nodes in the clique including the *dominator* and *connectors*.
3. The dominator always assumes the pure slave of the leader with MAC 100.

Theorem 2: In the scatternet built by Algorithm 3, no nodes assume dual roles, and the degrees of all dominatees and connectors are at most 7, while dominators have degree at most 30 in pessimistic estimation.

Proof: For a connector, it does not participate in one-hop dBBlue construction, so all its neighbors must be dominators, which is at most 5. For a dominatee, it only participates in one-hop dBBlue formation, so its degree is always bounded by 7. For a dominator, it assumes the bridge slave role for the backbone. It is in at most ℓ_2 piconets for the backbone since it has at most $\ell_2 = 24$ neighboring connectors. Additionally, it is in at most 6 one-hop dBBlue scatternet since we can partition its dominatees into 6 cliques at most (when the direction of a node can be found exactly; otherwise the number of cliques found will be slight larger). Consequently, its degree is at most $\ell_2 + 6 \leq 30$ under pessimistic estimation. As will see later in simulation results, the average node degree of dominators is much lower. ■

Algorithm 2 builds a degree-7 Bluetooth scatternet as long as there is one dominatee in each clique. While the structure built by Algorithm 3 is easier to be dynamically updated since CDS-based backbone and clusters are independent of each other. To evaluate the performance of the two algorithms, simulation is conducted in the section IV.

Notice that, in both algorithm 2and 3, the label addressing is carefully chosen. In other words, as will see late, our strategy is not just forming a network topology, but also build a DHT overlay in the network hence enable the self-routing in clusters.

III. Routing in Multi-hop Scatternet

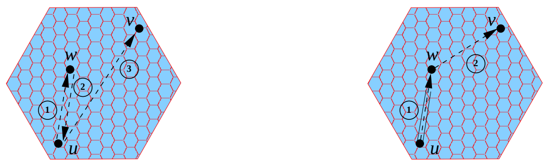
Position-based routing for wireless ad hoc networks has drawn considerable attention recently. However, it is not suitable for Bluetooth based personnel area networks since it requires additional GPS equipments hence increases the cost of Bluetooth devices. Moreover, Bluetooth networks are usually regarded as the extension of Internet, where IP-based routing is dominating. In this section, we propose a complete IP-based routing mechanism to integrate the proposed multi-hop scatternet with Internet seamlessly.

In the multi-hop scatternet, each cluster is assigned a network number, and every node in the cluster is dynamically assigned an IP address with same network number. The routing over the backbone could also be table driven protocols used in Internet. Since such kind of routings are well-studied, we omit the details of routing along the backbone here. For the packets targeting a node in other cliques in the cluster, the dBBlue protocol will first forward them to the cluster dominator, which then forwards the packets to the target clique. Here suppose that the dominator keeps an IP address range table for each clique. Eventually the packet will reach the target through the intra-clique routing as described late.

In each clique of a cluster, we adopt the self-routing mechanism of de Bruijn graph and applied the DHT (Distributed Hash Table) to ensure efficient IP-based routing. dBBlue structure [2] intrinsically provides the self-routing mechanism based on the labels derived from a pseudo-balanced de Bruijn graph. To enable the IP-based routing in a clique, we need map the IP address to the corresponding label. Given the IP address of the target node, the source node need know the label of the target node. One possible approach to solve this is to store all pairs of (IP, label) for all nodes in a node, e.g., the dominator node of the cluster. The source node always queries the dominator node for the label of the target node. Notice that such queries can be conducted using self-routing since the label of the dominator node is always fixed in our dBBlue structure. This centralized approach is simple, however, it suffers several disadvantages: the traffic storm problem to the dominator node, the single failure of the dominator node breaks the network, and so on.

We propose to use a distributed storage of the (IP, label) pairs. Each master node u in the dBBlue structure manages a lookup table, which stores the (IP, Label) pairs of those nodes whose key has u's piconet ID as prefix. Notice that the label is generated when we construct the dBBlue scatternet for each clique. The key of a node is some value computed from its IP, e.g., the hash value of its IP.

Assume that the length of every piconet ID in the dBBlue scatternet is between m and $m+1$. Each node first maps its host address, the suffix of its IP address, to a binary *key* with length $m+1$. The mapping technique could adopt any hashing function or simply translate its host address to binary format which is then abbreviated or extended to $(m+1)$-bits key. The node then forwards its key and (IP, label) pair to the target master node through the label-based routing in dBBlue scatternet. Notice that the target master node in which the pair will be stored has a label being a prefix of the key. Since the labels of the nodes are universal prefix free, the target master node is unique.



(a) Search and Forwarding (b) Packet-in-tunnel Forwarding

Fig. 2. Label based intradomain routing.

Consider the case that a node u wants to send packets to target node v in the same dBBlue scatternet while only IP address of node v is known. W.l.o.g., suppose the master node w holds the (IP, label) pair of node v. Node u first maps the IP address of node v to a key, say k. Two options , which are illustrated by Figure 2, could be used to send out the packet:

1. **Search and forwarding**. Node u queries the dBBlue backbone based on key k and gets the label of node v from the master node w. Node u then forwards the packet targeting v through dBBlue routing protocol. Figure 2(a) illustrates the mechanism.
2. **Packet-in-tunnel forwarding**. Node u adds an additional header with the key k to the packet then sends the packet out. The routing of the packet is based on the label of k. Once the master node w gets packet, it strips out the header and relays the packet to node v according to node v's label in its lookup table. Figure 2(b) illustrates the mechanism.

Remember the path between any two nodes in dBBlue structure is at most $2m + 2$ hops. The former approach can reduce the overall workload of dBBlue structure since the data packet travels through the network at most $2m+2$ hops, while the data packet travels at most $4m + 4$ hops in the latter case. But the latter approach does not need keep the packet before getting the target label as in the former approach, and the packet can reach the target faster if the time difference between transmitting different size packets is neglectful, because the total communication path is at most $6m + 6$ hops in the former while at most $4m + 4$ hops in the latter. On the other hand, the latter approach can keep the anonymity of node w hence increase security.

IV. Performance Evaluation

We conduct extensive simulations to study the performance of different multi-hop scatternet structures proposed in this paper.

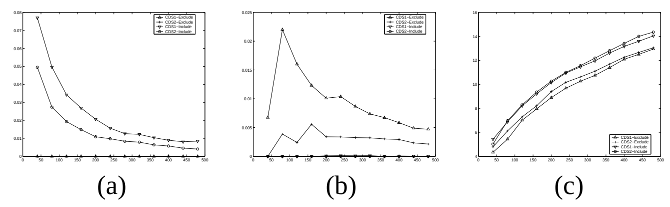


Fig. 3. The number of nodes varies in the range $[60, 500]$ while the node transmission range is fixed at 8. (a)Proportion of nodes with dual roles. (b)Proportion of nodes with degree exceeding 7. (c)Average communication hops.

All experiment results are shown in Figure 3. In the figures, we use *CDS1* to denote the CDS constructed by Algorithm 1, and *CDS2* to denote the CDS constructed by the algorithm from Wan *et., al* [7]. To also distinguish our two different scatternet formation methods for clusters, we use *Exclude* to denote Algorithm 3, and *Include* to denote Algorithm 2. Figure 3 (a)(b)(c) illustrate the performance variation when the number of wireless nodes varies in $[60, 500]$, while the transmission range of each node is fixed at 8 units. The first observation is that the diameter of scatternet is about $\log n$ as we expected, where n is the number of wireless nodes. The average degree in backbone keeps almost constant after the density reaches some extent. In Figure 3, the proportion of nodes with dual roles, and the proportion of nodes with degree exceeding 7 drop when the network density increases. As we expected, the scatternet backbone based on *CDS1* method does provide smaller average hops between any pair of nodes than based on *CDS2* method. Thus, more energy is saved. The multi-hop scatternet constructed by *Exclude* method has lower diameter than that produced by *Include* method. In addition, no nodes assume dual role in the scatternet formed by the *Exclude* method. While the *Include* method has its own advantage: almost no node has degree more than 7.

V. Conclusion

In this paper, we proposed a novel solution for multi-hop Bluetooth scatternet formation and self-routing protocol. We proposed a novel communication efficient method to build a connected dominating set (CDS) as the backbone of multi-hop Bluetooth network. Then *dBBlue* scatternet is formed for each cluster. Our method does not need any position information for scatternet construction and routing.

References

[1] J. C. Haartsen, "The bluetooth radio system," *IEEE Personal Communications*, vol. 7, pp. 28–36, 2000.
[2] W.-Z. Song, X.-Y. Li, Y. Wang, and W. Wang, "dBBlue: Low diameter and self-routing bluetooth scatternet," in *DialM-POMC*, 2003.
[3] T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaire, "Distributed topology construction of bluetooth personal area networks," in *Proc. IEEE INFOCOM*, 2001.
[4] C. Petrioli, S. Basagni, and I. Chlamtac, "Configuring bluestars:multihop scatternet formation for bluetooth networks," *IEEE Transactions on Computers*, vol. 52, no. 6, pp. 779–790, 2003.
[5] K. M. Alzoubi, *Virtual Backbone in Wireless Ad Hoc Networks*, Ph.D. thesis, Illinois Institute of Technology, 2002.
[6] I. Chlamtac and A. Farago, "A new approach to design and analysis of peer to peer mobile networks," *Wireless Networks*, vol. 5, pp. 149–156, 1999.
[7] P.-J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," in *IEEE INFOCOM*, 2002.
[8] J.Wu and H.L. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless network," in *3rd ACM international workshop on Discrete algorithms and methods for mobile computing and communications*, 1999, pp. 7–14.
[9] Y. Wang and X.-Y. Li, "Geometric spanners for wireless ad hoc networks," in *Proc. of 22nd IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2002.
[10] L. Barriere, P Fraigniaud, L. Narajanan, and J. Opatrny, "Dynamic construction of bluetooth scatternets of fixed degree and low diameter," in *14th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2003, pp. 781–790.
[11] C. Law, A.K. Mehta, and K.Y. Siu, "Performance of a new bluetooth scatternet formation protocol," in *Proc. ACM Symposium on Mobile Ad Hoc Networking and Computing MobiHoc*, 2001, pp. 183–192.
[12] Z. Lei and T. Lim, "Estimation of directions of arrival of multipath signals in cdma systems," *IEEE Trans. Communications*, vol. 48, no. 6, pp. 1022–1028, June 2000.
[13] K. Balaji, S. Kapoor, A.A. Nanavati, and L. Ramachandran, "Scatternet formation algorithms in the bluetooth network," 2001.
[14] G.V. Zaruba, S. Basagni, and I. Chlamtac, "Bluetrees - scatternet formation to enable bluetooth based ad hoc networks," in *Proc. IEEE ICC*, 2001.
[15] G. Tan, A. Miu, J. Guttag, and H. Balakrishnan, "Forming scatternets from bluetooth personal area networks," Tech. Rep. MIT-LCS-TR-826, MIT, 2001.
[16] Z. Wang, R.J. Thomas, and Z. Haas, ," in *HICSS*, 2002.
[17] R. Guerin, E. Kim, and S. Sarkar, "Bluetooth technology key challenges and initial research," in *Proc. SCS Communication Networks and Distributed Systems Modeling and Simulation CNDS*, 2002, pp. 157–163.
[18] S. Basagni and C. Petrioli, "A scatternet formation protocol for ad hoc networks of bluetooth devices," in *IEEE Vehicular Technology Conference, VTC Spring*, 2002.
[19] S. Basagni, "Distributed clustering for ad hoc networks," in *Int. Symp. Parallel Algorithms, Architectures and Networks ISPAN*, 1999, pp. 310–315.
[20] C. Petrioli and S. Basagni, "Degree-constrained multihop scatternet formation for bluetooth networks," in *Proc. IEEE GLOBECOM*, 2002.