SCATTERNET FORMATION AND SELF-ROUTING IN BLUETOOTH NETWORKS

YU WANG*, WEN-ZHAN SONG†, AND XIANG-YANG LI†

Abstract. In this chapter, we first summarize the criteria of satternet design for Bluetooth network, and review different scatternet formation algorithms for both single-hop and multi-hop networks. Then, we survey Bluetooth routing algorithms and review several scatternet topologies which have self-routing properties.

Key words. Bluetooth, scatternet formation, routing, neighbor discovery

1. Introduction. Bluetooth [12] 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. 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 a major technology for short range wireless networks and wireless personal area networks. This chapter deals with the problem of building ad hoc networks using Bluetooth technology.

According to the Bluetooth 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 several piconets is called a *scatternet* (as shown in Figure 1.1). 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 the master in one piconet and a slave in other piconets (as node a in Figure 1.1, which is the master of piconet I and a slave of piconet II) or be slaves in multiple piconets (as nodes b and c in Figure 1.1). A node with multiple roles acts as *bridge* or *gateway* between the piconets to which it belongs. 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 the one that minimizes the roles assigned to the nodes, without losing network connectivity.

While several solutions and commercial products have been introduced for Bluetooth communication, the Bluetooth specification does not indicate any method for scatternet formation. The problem of scatternet formation has not been dealt with until very recently. The solutions proposed in the literature can be divided into single-hop and multi-hop solutions. Several criteria could be set as the objectives in forming scatternet. First of all, the formatted scatternets should keep the network connectivity, i.e., the scatternets are connected if the original communication graph is

*Department of Computer Science, University of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, NC 28223 (ywang32@uncc.edu).

†Department of Computer Science, Illinois Institute of Technology, 10 West 31st Street, Chicago, IL 60616 (songwen@iit.edu, xli@cs.iit.edu)

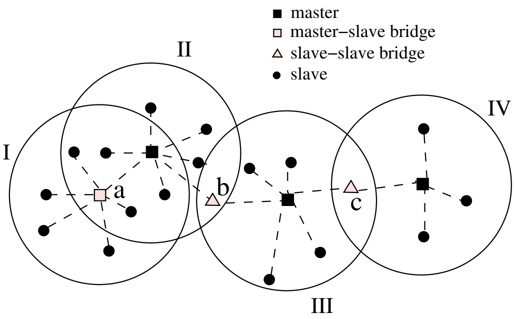


FIG. 1.1. *Scatternet formed by 4 piconets.*

connected. Secondly, the protocol should create degree limited scatternets, to avoid parking any slave node. Thirdly, the number of piconets should be minimized to reduce the inter-piconet scheduling and communication cost. Fourthly, the formation and maintenance of scatternet should have small communication overhead. Fifthly, the diameter of the scatternet should be small, i.e., the maximum number of hops between any two devices must be small to provide faster routing. Sixthly, the scatternet formation may enable efficient self-routing algorithms in the scatternets. In this chapter, we survey the solutions for scatternet formation and self-routing for both single-hop and multi-hop ad hoc networks.

The rest of this chapter is organized as follows. Section 2 discusses a closely related problem of scatternet formation: *neighbor discovery* in Bluetooth networks. Section 3 surveys solutions that for generating scatternets for single-hop and multihop networks. Section 4 describes some self-routing methods for several proposed scatternets. Finally, we conclude our chapter in Section 5.

2. Neighbor Discovery. 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 [58]. Most of the scatternet formation algorithms have *device discovery* procedure to learn about devices in its neighborhood. The device discovery procedure is also called *inquiry procedure* in Bluetooth specifications.

Bluetooth devices use the inquiry procedure to discover nearby devices, or to be discovered by devices in their locality. The inquiry procedure is asymmetrical. The inquiry procedure uses a special physical channel *inquiry scan channel* for the inquiry requests and responses. A Bluetooth device that tries to find other nearby devices is known as an *inquiring device* and actively sends inquiry requests. It iterates (hops) through all possible inquiry scan channel frequencies in a pseudo-random fashion, sending an inquiry request on each frequency and listening for any response. Bluetooth devices that are available to be found are known as *discoverable devices* and listen for these inquiry requests on their inquiry scan channel and send responses to those requests.

In [58, 9, 17, 59], the device discovery is performed as follows. Each device alternates between *inquiry* mode (as the inquiring device) and *inquiry scan* mode (as the discoverable device), remaining in each mode for a time selected randomly and uniformly in a predefined time range. Inquiry nodes select a repeated pattern of 32 frequencies in inquiry scan channel and send inquiry request on selected frequency. Inquiry scan nodes also select a frequency at random in each frequency of the the inquiry scan channel and listen to the requests. When two devices in opposite inquiry modes handshake (frequency-matching), they set up a temporary piconet that lasts only the time necessary to exchange their ID and other information necessary for the scatternet formation. In [13], the authors conduct real-world measurements and simulations, derive the optimal parameters for symmetric ad hoc neighbor discovery.

Recently, several methods have been proposed to improve the Bluetooth device discovery procedure [27, 69, 15, 68, 33, 49, 60, 57]. In [27], the authors let each device perform the device discovery protocol until it is connected with c neighbors, where c is a constant between 5 and 7. Their experiments show that the resulting graph is connected with high probability if the original communication graph is connected. In [68], the authors suggest three possible changes to the Bluetooth specification: eliminating or decreasing the random backoff delay in INQUIRY SCAN, using a single frequency train instead of two in INQUIRY, and a combination of the two. Their experiments show that these methods can improve the connection setup time up to 75% without deteriorating the overall system performance. A hardware empirical testbed is developed to verify these methods in [49]; the result suggests that a single train with no backoff has the best performance. In [33], the authors also proposed three methods to speed up the device discovery: half inquiry interval (HII), dual inquiry scan (DIS), and combination of HII and DIS. The result shows a reduction of average frequency-matching time from 23.55 seconds to 11.38 seconds. In [60], it is pointed out that the scalability of Bluetooth inquiry procedure is not sufficient if many devices are present. As a result of this observation, an adaptive protocol for cooperative device discovery is proposed to allow devices to exchange their knowledge of nearby devices, such as addresses and clocks, to reduce energy consumption and improve scalability for environments with many devices. In [69, 15], it is suggested to use auxiliary devices, such as IrDA interfaces or RFID transponders, to facilitate connection setup. Recently, [57] propose a simple neighbour discovery (SND) procedure instead of Bluetooth inquiry procedure for Bluetooth, which is suited to systems where peer nodes are communicating.

3. Bluetooth Scatternet Formation. Given n nodes currently distributed in the network, Bluetooth scatternet formation algorithms group the nodes into piconets and join the piconets into a connected scatternet. After the neighbor discovery, the Bluetooth devices know the information of its neighbors in the communication graph. Here, the communication graph is a graph in which there is a link between any two devices who are in each other's transmission ranges. If we assume that all devices have the same transmission ranges, the communication graph is modelled by a *unit disk graph* in which there is a link between two nodes whose Euclidean distance is less than or equal to one. Then the problem of scatternet formation becomes to construct a connected subgraph of the unit disk graph and to select piconets (and assign master and slave roles to nodes in each piconet) so that the resulting scatternet has some desirable properties.

3.1. Criteria of Scatternet Design. There are various desirable properties [63, 8, 51] for scatternets which have been used by different scatternet formation algorithms. We summarize the criteria of scatternet design as follows:

- Guarantee of Connectivity. If the communication graph from device discovery phase is connected, the scatternet formed by scatternet formation algorithm should also be connected. Connectivity is the most basic feature of the network topology, it guarantees that there exist at least one path from one device to any other devices.

- Single Master Role. Master node is belong to exactly one piconet, there is no master-master bridge. This is a requirement in the Bluetooth specifications.
- Minimal Number of Roles. The standard allows multiple roles for the same device, but one node can be active only in one piconet. To operate as a member of another piconet, a node has to switch its hopping frequency. The switch operation causes delays and big overheads. So an efficient scatternet formation algorithm may minimize the number of roles assigned to each node. Some algorithms even only allow bridges connect to at most two piconets.
- Minimal Number of Piconets. The number of piconets (i.e., the number of nodes with master role) should be minimized to provide faster routing and keep maintenance overhead small. Notice that the worse case we can have all nodes as masters of their neighbors.
- Limited Piconet Size. Though each piconet can have more than eight devices, only eight of them can be active as one master and seven slaves in the same time, other devices are forced to be parked. In order to communicate with all nodes, the master node need to park and unpark its slaves. This will significantly reduce the bandwidth and throughput of the network. Therefore, the size of piconet is expected to be limited by eight, so that the nodes can communicate with each other without parking and unparking operations. In other words, we hope the scatternet have node degree bounded by eight.
- Minimal Number of Master-Slave Bridge. When a device serves as a master-slave bridges (as node a in Figure 1.1) between two piconets, if it acts as the slave in one piconet, all the communication in the other piconet (where it serves as the master) will be "frozen". This reduces the throughput of the network. Therefore, comparing to master-slave bridges, slave-slave bridges (as nodes b and c in Figure 1.1) are to be preferred by scatternet formation.
- Minimal Scatternet Diameter. The diameter of the resulting scatternet is the number of hops of the longest path between any two devices in the networks. If the diameter is bounded by $f(n)$, then we can find a route with at most $f(n)$ hops for every pair of devices. For example, the diameter of dBBlue [61] is bounded by $O(\log n)$, which means the length of route is at most $O(\log n)$ for any pair of source and target.
- Efficient Routing. Several proposed scatternet formation algorithms [61, 64] also enable efficient self-routing in which device does not need to maintain routing table. Some routing protocols [14, 36] ask the topology be planar so that they can guarantee the delivery, then planar scatternets are constructed in [66, 40]. In addition, the scatternet should have multiple routes between any pairs of devices to keep the routing robustness.
- Easy To Formate, Update. Due to the limited resources and high dynamics (e.g. node leaving, node joining or node moving) of the wireless nodes, it is preferred that the scatternet can be constructed and maintained in a distributed (or even localized) manner. Here, in localized scatternet formation, each node makes formation decisions solely based on the local information from its neighbors. When a node move, appear or disappear from the network, the scatternet should be updated easily by scatternet maintenance (or

self-healing) protocols.

- Resource-based Master Selection. Notice that a master need to handle and operate all the communication with its slaves in the piconet. So it will cost more resources in master node than in slave nodes. Therefore, scatternet formation algorithm may consider the available resources in each node when selecting the master node.
- QoS: Delay, Throughput and Capacity. Many algorithms [48, 47, 44, 2] also consider the QoS criteria during the scatternet formation. For example, given the traffic matrix of the network, find the scatternet that can minimize the average packet delay or maximize the network capacity.

Some of these properties are contradictive and hard to achieve together, but an efficient scatternet formation protocol can achieve most of them or at least several of them. The solutions proposed in literature can be divided into single-hop and multi-hop solutions.

3.2. Scatternet Formation Algorithms for Single-hop 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. In this subsection, we review several scatternet formation algorithms for single-hop networks.

3.2.1. Central Decision Methods. Salonidis *et al.* [58] proposed a topology construction algorithm based on leader election, called *Bluetooth Topology Construction Protocol* (BTCP). 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 first process, one for each connected component. In the second phase, leader then collects the information about the whole network, decides the roles for each node, and distributes back the roles to all nodes. In other words, basically, it is a centralized approach, and the decision is made by a central super node. Thus, the solution is not scalable (for dynamic environments where devices can join and leave after the scatternet is formed), and not localized, the time complexity is large. Moreover, how to assign the roles is not elaborated in [58]. They also assume up to 36 nodes in the network.

3.2.2. Tree Based Methods. Law, Mehta and Siu [38] described an randomized and distributed algorithm that creates connected degree bounded scatternet in single-hop networks. Every node starts out as a leader. Each leaser flips a coin to see whether it goes into *scan* or *seek* mode. When two leaders are connected, one must *retire* and the components will be merged. The authors gave five cases to handle the merge. The final structure is a tree like scatternet, which limits efficiency and robustness. They proved that the algorithm achieves O(log n) time complexity and O(n) message complexity. The scatternets formed by their protocol have the following properties: (1) any device is a member of at most two piconets, and (2) the number of piconets is close to be optimal. They validated the theoretical results by simulations, which also show that the scatternets formed have O(log n) diameter.

Tan *et al.* [65] proposed a distributed Tree Scatternet Formation (TSF) protocol for single-hop networks, which is similar with the multi-hop methods in [71]. TSF connects nodes in a tree structure that simplifies packet routing and scheduling. At any point in time, the TSF-generated scatternet is a forest consisting of connected tree components. Every root node in one component elects a single coordinator responsible for discovering other tree scatternets. The coordinator is prefer to be leaf nodes, since leaf nodes are not communication bottlenecks and have more spare capacity for discovering neighboring devices. TSF allows nodes to arrive and leave at arbitrary times, incrementally building a tree topology and healing partitions when they occur. The extensive simulation results indicated relatively short scatternet formation latency. However, TSF is not designed to minimize the number of piconets. The simulation results suggest that each master usually has fewer than 3 slaves.

Sun, Chang and Lai [64] described a self-routing topology for single-hop Bluetooth networks where the routing overhead is kept to a minimum. Nodes are organized and maintained in a search tree structure, with Bluetooth ID's as keys (these keys are also used for routing). It requires only fix-sized message header and no routing table at each node regardless of the size of the scatternet. These properties make the solution scalable to deal with networks of large sizes. It relies on a sophisticated scatternet merge procedure with significant communication overhead for creation and maintenance.

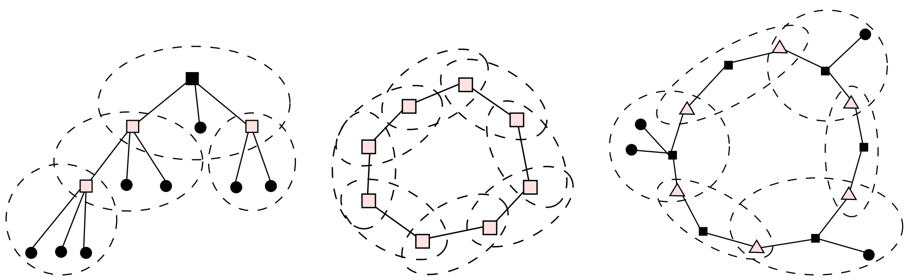
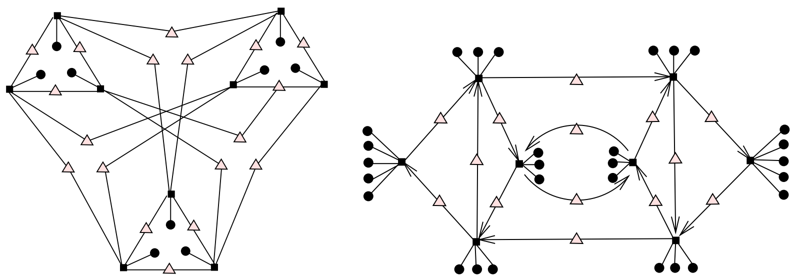


Fig. 3.1. *Different Scatternets: (a) BlueTree; (b) BlueRing with master-slave bridges; (c) BlueRing with slave-slave bridges.*

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. Figure 3.1(a) illustrates a BlueTree formed by four piconets, where the leaves are pure slaves, the root is a master, and other nodes are master-slave bridges.

3.2.3. Ring Based Methods. Bluerings as scatternets are proposed in [28, 42]. Ring structure for Bluetooth has simplicity, easy creation and easy routing as advantage, but it suffers large diameter (i.e., the maximum number of hops between any two devices can be as bad as $O(n/2)$) and large number of piconets. In the ring structure from [28], each device acts as a master-slave bridge to connect itself to the two neighbors in the ring. See Figure 3.1(b) for illustration. However, in [42], the authors used slave-slave bridges and masters to form the ring, as in Figure 3.1(c). In [42], they also addressed in detail the formation, routing and topology maintenance for the ring structure. Due to the self-routing and easy to maintain, ring structure is good for small-size or median-size scatternets.

3.2.4. Other Well-known Structures Based Methods. Barriere, Fraigniaud, Narajanan, and Opatrny [3] 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, in other words there are only slave-slave bridges. Figure 3.2(a) illustrates an example of the scatternet based on projective geometry with 36 nodes. 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. The diameter of the scatternet is bounded by $O(\log^2 n \log^2 \log n)$. In addition, the connectivity of masters is high (i.e., for any pair of master, the number of edge-disjoint paths connecting them in the network is large than some constant). 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 [3], 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.



(a) and (b)

Fig. 3.2. *Different Scatternets: (a) Projective scatternet; (b) dBBlue based on de Bruijn graph.*

Song *et al.* [61] adopted the well-known structure *de Bruijn graph* to form the backbone of Bluetooth scatternet, called *dBBlue*, such that every master node has at most seven slaves, every slave node is in at most two piconets, and *no* node assumes both master and slave roles. Figure 3.2(b) illustrates an example of *dbBlue* based on the de Bruijn graph $B(2,3)$. Their structure *dBBlue* also enjoys a nice routing property: the diameter of the graph is $O(\log n)$, s.t., it can find a path with at most $O(\log n)$ hops between every pair of nodes without any routing table. Moreover, the network congestion is at most $O(\log n/n)$, assuming that a unit total traffic demand is evenly distributed among all pair of nodes. They also proposed a vigorous method to *locally* update the structure *dBBlue* using at most $O(\log n)$ communications when a node joins or leaves the network. In most cases, the cost of updating the scatternet is actually $O(1)$ since a node can join or leave without affecting the remaining scatternet. The number of affected nodes is always bounded from above by a constant when a node joins or leaves the network. Their method can construct the structure *dBBlue* incrementally when the nodes join the network one by one. In addition, the structure formed by their method can sustain the faults of 2 nodes and the network is still guaranteed to be connected. If a node detects a fault of some neighboring master node or bridge slave node, it can dynamically re-route the packets and the path traveled by the packet is still at most $O(\log n)$ hops. By designing a novel method for assigning MAC addresses to nodes, dBBlue structure can enable the self-routing even during the updating procedures when node leaves or joins the network. We will review it in Section 4.

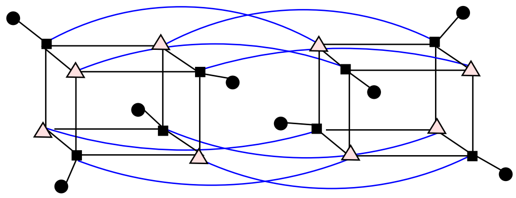


FIG. 3.3. *Scatternet: BlueCube.*

Chang *et al.* [16] presented a three-stage distributed construction protocol to automatically construct a hypercube based scatternet, called *BlueCube*. Figure 3.3 illustrates an example of *BlueCube*. The proposed protocol tackles the link construction, role assignment, scatternet formation and network management problems, to construct efficiently a hypercube structure. The construction of the scatternet has three phases: first form a ring scatternet, then switch roles for some nodes to reduce the number of piconets and connect some unconnected devices, at last form the hypercube for all devices in the ring. The proposed protocol enables Bluetooth devices easily to construct a routing path, tolerate faults and create disjoint paths.

3.2.5. Cluster Based Methods. Ramachandran *et al.* [56] proposed a two-stage distributed $O(n)$ randomized cluster algorithm for a n node single-hop network, that always finds the minimum number of star-shaped clusters, which have maximum size 8. The first stage of the algorithm is randomized, at the end of which each node either becomes a master-designate or a slave-designate. The second stage corrects the effect of the randomness introduced in the previous stage by using a deterministic algorithm to decide on the final set of masters and slaves, and to efficiently assign slaves to masters. A super-master is elected, which counts the actual number of masters and collects information about all the nodes. The super-master can then run any centralized algorithm to form a network of desired topology (selecting the bridges). The election of the super-master is interleaved with the cluster formation, which speeds up the ad hoc network formation. In [56], they also proposed a deterministic distributed algorithm for the same model which achieves the same purpose. The basic idea of this algorithm is that nodes discovering each other form a tree of responses, the root of each tree being a master, all other nodes in the tree being its slaves. Each tree form a cluster. Then the second half of the algorithm involves the election of a super-master among the masters. They applied the same method (form a tree of responses) among all the masters. And again the super-master will decide the final scatternet.

3.2.6. QoS Based Methods. Baatz *et al.* [2] proposed a single-hop Bluetooth scatternet formation scheme based on 1-factors which allow a maximum aggregated throughput in the corresponding scatternet. They first elaborated on Bluetooth scatternet capacity with special respect to co-channel interference. Then they introduced a class of Bluetooth scatternet topologies (constructed from one-factorizations) with optimal aggregated throughput that are easy to schedule in a fair manner. In other words, the scatternet allows a maximum number of simultaneously active piconets. As a variable number k of 1-factors may be used to build a topology, one is able to find a tradeoff between scheduling overhead on the one hand and robustness and average path length on the other hand. Due to the chosen construction, the 1-factors can be computed easily for a given number of nodes and a given k (in time linear to the number of links). However, piconets are not degree limited in their scheme.

Miorandi and Zanella [48] investigated the impact of the master choice on the performance of a Bluetooth piconet. They assumed the end-to-end traffic matrix of the single-hop network is given. They proposed an optimal and a suboptimal criterion for the choice of the master unit to minimize the average packet delay. However, the optimization requires high computational capability. In [47], Miorandi, Trainito and Zanella further studied the relationship between capacity and topology for Bluetooth scatternets. They discussed how Bluetooth nodes should be organized to build up a scatternet, where the efficiency of the resulting configuration is measured in terms of network capacity instead of packet delay. They presented a theoretical study of intrinsic capacity limits of a scatternet, where the maximum throughput may be achieved under local traffic. They first discussed some conditions to achieve efficient piconets interconnection. Then, they investigated the performance achieved by some specific scatternet topologies, both "planar" and "solid", in case of uniform traffic matrix, that is, assuming that every node in the network generates an equal amount of traffic towards any other node. In [44], Marsan *et al.* also studied how to construct the optimal topology that provides full network connectivity, fulfills the traffic requirements and the constraints posed by the system specification, and minimizes the traffic load of the most congested node in the network, or equivalently its energy consumption. By using a min-max formulation, they provided a solution based on integer linear programming. Due to the problem complexity, the optimal solution is attained in a centralized manner, which is the limitation of their method.

3.2.7. Virtual Position Based Methods. In [66], Wang *et al.* applied the position-based scheme proposed by Li *et al.* [40] 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 [31] for short range Bluetooth devices, aimed primarily at home and office environments. However, 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 their method [66], nodes first randomly select their virtual positions, then based on these positions, a planar structure (minimum spanning tree, Gabriel graph, relative neighborhood graph or Delaunay triangulation) can be built. As in [40], then they bound the degree by applying Yao graph on the structure, and assign the roles for the scatternet. We will review the detailed method of [40] in Section 3.3.3.

3.2.8. Genetic Methods. Recently, Sreenivas and Ali [62] proposed an evolutionary approach to scatternet construction, wherein they used a genetic algorithm to find a global optimum: the best, or fittest, combination of masters, slaves and bridges in a given Bluetooth network. Their solution considered only slave-slave bridges. The algorithm executes in two phases, role determination and connection establishment. In the first phase, the genetic algorithm selects random groups of nodes: these groups constitute the initial population. Each group corresponds to a combination of masters, slaves and bridge nodes. The fitness of each group of nodes is evaluated based on the number of master nodes (or piconets), slave nodes and bridge nodes in the network. A desirable property of scatternets is to minimize the number of masters, maximize the number of slaves and ensure that the bridge nodes are not too few such that bottlenecks are created. Taking this property into consideration, a fitness value is derived for each group in the population. Then a new population will be generated repeatedly by the genetic algorithm, until the end condition based on the universal lower bound for the number of piconets is satisfied. They showed that their scatternet formation algorithm produces scatternets with certain desirable characteristics: minimal delay to the end-users during scatternet formation (i.e. the number of generations that the genetic algorithm in first phase iterates through), minimal number of piconets in order to reduce inter-piconet interference during communication and bounded number of slaves to minimize the overhead associated with slave parking and unparking operations.

3.3. Scatternet Formation Algorithms for Multi-hop Networks. In this section we review the solutions of scatternet formation for multihop networks. In a singlehop topology, all devices are in the radio vicinity of each other, which is not always the case in realistic scenarios.

3.3.1. Tree Based Methods. Zaruba, Basagni and Chlamtac [71] proposed two distributed tree-based protocols for forming connected scatternet. In both cases, the resulting topology is termed as *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 then 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). See Figure 3.1(a) for illustration. In order to limit the number of slaves, they [71] 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 five slaves. If a master node has more than five 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 [71]. In the second protocol [71], 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. Remember that the tree topology suffers from a major drawback: the root is a communication bottleneck. In addition, dynamic updating that preserves correct routing is not discussed in these protocols. There are several modified versions of BlueTree, such as [26, 32], to improve the communication overhead or increase the connectivity. Cuomo et al. [21] also proposed a tree-based scatternet formation algorithm *SHAPER* for multi-hop network, which focuses on the self-healing behavior of the tree structure: i.e., it is able to dynamically reconfigure the scatternet after topological variations due to mobility or failure of nodes.

Guerin et al. [30] proposed depth/breath first search and MST-based scatternet formation schemes for unit graphs in two and three dimensions. They construct a tree where all nodes at one level are either masters or slaves (i.e., they construct bipartite graphs). Their construction does not guarantee maximum degree bound unless the structure itself provides the bound. For example, MST in two dimensions has a maximum degree of five, but in three dimensions, some nodes can have degrees up to 13. The schemes are also not localized.

3.3.2. Cluster Based Methods. Basagni, Petrioli and Chlamtac [53, 6] described a multihop scatternet formation scheme based on clustering scheme [41]. The constructed scatternet is called *BlueStars*. The protocol proceeds in three phases: device discovery, partitioning of the network into piconets (stars) by clustering, and interconnection of the piconets to connected scatternet. In the second phase, *BlueStars Formation*, clusterhead (master role) decisions are based on node weights (instead of node IDs, as used in [41]), that express their suitability to become masters, following a variant of the clustering method described in [4]. All clusterhead nodes are declared master nodes in a piconet, with all nodes belonging to their clusters as their slaves. Then in the third phase, *BlueConstellation*, some of the slaves become masters of additional piconets, i.e. become master-slave bridges, to assure the connectivity of the scatternet. However, piconets in the scatternet may have more than seven slaves. This may result in performance degradation, as slaves need to be parked and unparked in order for them to communicate with their master. A performance evaluation of the clustering-based scatternet formation scheme [6] is given in [5, 53].

To fix the unbound slave number, Basagni, Petrioli and Chlamtac [54, 52] modified their protocol [53, 6] and proposed a new scatternet called *BlueMesh*. The idea of bounding the slave number in BlueMesh is again based on the observation (also used in [71]) that if a node in unit disk graph has more than five neighbors then at least two of them must be connected. Same with *BlueStars*, the selection of the masters is based on the node weights. However, the selection of slaves is performed in such a way that if a master has more than 7 neighbors, it only chooses 7 slaves among them so that via them it can reach all the others. This phase proceeds in iterations. Initially all nodes are undecided. Nodes that participate in a given iteration perform the above modified clustering process (deciding the master and slave roles). In each iteration, the decided nodes and links will be removed from the next iteration. After the roles decided for all nodes, finally, the bridges are chosen to connect all piconets to the connected scatternet. The selection of bridges is same as in BlueStars, bridges are chosen so that there is an inter-piconet route between all masters that are at most three hops away.

Variants of clustering-based scatternet formation schemes were proposed in [67, 29]. Wang, Thomas and Haas [67] proposed a scatternet formation shceme, called *Bluenet*. It is a 3-phase algorithm. In the first phase, nodes enter page state randomly, trying to invite less than seven neighbors to join its piconet. Once a node becomes a slave, it will stop paging or answering pages. When phase-1 is finished, several separate piconets are formed, and there are also some isolated nodes. In the second phase, isolated nodes will page all of its neighbors and try to connect them to some initial piconets built in phase-1. In last phase, master of each piconet instructs their slaves to set up outgoing links, so that piconets are connected to form a scatternet. Guerin, Kim and Sarkar [29] also proposed a distributed cluster algorithm for scatternet formation. Initially all nodes have unassigned states, and nodes discover other nodes randomly. When two nodes meet for the first time and both are unassigned, the one with the highest ID becomes master, the other becomes its slave. When two nodes meet and one is unassigned while the other is master, the unassigned node become a slave of the master, if the master has less than 7 slaves. When two nodes meet and one is unassigned while the other is slave, the unassigned node becomes master, and the slave becomes a bridge node. When two nodes meet and both are masters, nothing changes. When a master node meets a slave node, the slave will join the master's piconet, and becomes a bridge node. Both clustering processes [67, 29] follow a random fashion. Initial connections are made by nodes entering scan or inquiry scan phases at random. Already existing master nodes have priority in attracting more slaves, up to the limit. After each node is assigned master or slave role, or is unable to join any piconet or attract any neighbor as its slave to create its own piconet, some bridge piconets are added to connect the scatternet. However, both methods [67, 29] do not always lead to a connected structure.

3.3.3. Position Based Methods. In [40, 39], Li, Stojmenovic and Wang proposed the first schemes that construct degree limited (a node has at most seven slaves) and connected piconets in multihop networks, without parking any node. Notice that the schemes in [54, 52] can also achieve bounded degree scatternet. Their neat scheme does not require position information, but instead the local information is extended to two hop information, with a two round device discovery phase for obtaining necessary information. In Li *et al.*'s solution, nodes know their positions and are able to establish connections with other nodes within their transmission radius in the neighbor discovery phase. The second phase of the proposed formation algorithm is optional, and can be applied to construct a sparse planar geometric structure, such as Gabriel graph (GG), relative neighborhood graph (RNG) or partial Delaunay triangulation (PDT). Note that each node can make local decisions about each of its edges in these graphs without any message being exchanged with any of its neighbors. Thus this construction has basically no cost involved. In the third mandatory phase, the degree of each node is limited to 7 by applying Yao structure [1], 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. The authors considered 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. The creation and maintenance of the scatternets require small overhead in addition to maintaining accurate location information for one-hop neighbors. The experiments confirmed good functionality of created Bluetooth networks in addition to their fast creation and straightforward maintenance.

ALGORITHM 1. *Postion-based Scatternet Formation Algorithm*

1. Neighbor discovery and information exchange (collecting the node degree information).
2. Planar subgraph construction (constructing RNG, GG, or PDT and degree information exchange), if desirable.

[1]The Yao graph [70] 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.

3. Bounding degree and assigning roles (consisting of several iterations).
   Initially all nodes are undecided. In each iteration, if a undecided node u has the highest degree among its all undecided neighbors, it runs the following steps:
   (a) Bound its degree (applying Yao structure).
   (b) Assign role to itself (based on the information on each link or using cluster based method).
   (c) Mark itself decided, and notice the deleted edges and its status to its undecided neighbors.

   Repeat the iterations, until all nodes are decided.

Recently, Basagni et al. [10, 7] described the results of an ns2-based comparative performance evaluation among four major solutions for forming multihop scatternet: [40, 53, 71, 67]. They found that device discovery is the most time-consuming operation, independently of the particular protocol to which it is applied. The comparative performance evaluation showed that due to the simplicity of its operations BlueStars [53] is by far the fastest protocol for scatternet formation. However, BlueStars produces scatternets with an unbounded, possibly large number of slaves per piconet, which imposes the use of potentially inefficient Bluetooth operations. They proposed a combined solution by applying a Yao structure as described here on each piconet, to limit the degree of each master node to seven. This is a variant of the clustering-based scheme presented in this article, with degree limitation applied at the end instead of during the scatternet creation process.

3.3.4. On-demand Methods. Most above scatternet formation protocols tend to interconnect all Bluetooth devices at the initial network startup stage and maintain all Bluetooth links thereafter. The master or bridge nodes in the resulting scatternet may become the traffic bottleneck and reduce network throughput. To make the scatternet structure more suitable to serve in mobile ad hoc networks, recently several on-demand methods [43, 37, 50, 20] (to build scatternets only along the multihop routes with traffic demands and eliminate unnecessary link and route maintenances) are proposed.

Liu, Lee and Saadawi [43] proposed a scatternet-route structure to combine the scatternet formation with on-demand routing, thus build scatternets only along the multihop routes with traffic demands. This route-type scatternet is called scatternet route. As the scatternet routes survive along with the on-going traffic flows, no unnecessary Bluetooth link maintenance is needed. The scatternet route is designed to have a special master-slave alternate structure to enable the devices along the route to connect together via Bluetooth links. The formation of the scatternet route is similar to the common on-demand routing protocols. They introduced an extended ID (EID) connectionless broadcast scheme to expedite the route discovery and construction. To remove the piconet switch overhead suffered by the bridge devices inside the scatternet route, they proposed to align the time slot of all piconets along each scatternet route. The synchronized scatternet route is shown to reach higher network throughput and undergo shorter data transmission delays. Finally, in order to enable fair and efficient packet transmissions over scatternet routes, they also designed a route-based scatternet scheduling algorithm.

Kawamoto *et al.* [37] proposed a Two-Phase Scatternet Formation (TPSF) protocol with the aim of supporting dynamic topology changes while maintaining a high aggregate throughput. In the first phase, a control scatternet is constructed for control purposes (i.e., to support dynamic join/leave, route discovery, etc). After the control scatternet formation in the first phase, each master node maintains all the information of its slaves and bridges nodes within its piconet and adjacent piconets. This information is exploited during the second phase. The second phase is invoked whenever a node needs to initiate data communications with another node. A dedicated piconet/scatternet is constructed on-demand between the communicating nodes, using any on-demand source routing protocols proposed for wireless mobile ad hoc networks. In [37], they use the dynamic source routing (DSR) protocol [34] for route selection. Since the on-demand scatternet can dedicate all the time slots to a single communication session, it has the capability to provide a high throughput and a small end-to-end data transfer delay. The on-demand scatternet is torn down when the data transmissions are finished. The simulation results showed that the proposed TPSF protocol achieves a higher average aggregate throughput when compared with BTCP [58].

Pagani *et al.* [50] also proposed an On-Demand Bluetooth scatternet formation algorithm (ODBT). ODBT characterizes an ad hoc infrastructure with a tree topology. It is able to cope with topology changes due to either leaving or moving Bluetooth devices, as well as with devices that dynamically join the scatternet. The authors described in detail how ODBT can be implemented in the Bluetooth protocol stack, and analyzed its performance by simulations.

3.3.5. QoS Based Methods. Marsan *et al.* [44] studied how to construct the optimal topology that provides full network connectivity, fulfills the traffic requirements and the constraints posed by the system specification, and minimizes the traffic load of the most congested node in the network, or equivalently its energy consumption. By using a min-max formulation, they provided a centralized solution based on integer linear programming.

Recently, Chiasserini *et al.* [18] extended the work in [44] and enhanced the optimization problem by adding the constraints on the network capacity. As in [44], they gave a min-max formalization of the topology formation problem. They assume that just one route is used for each source destination pair. For each traffic source, they took the average traffic rate as an input parameter to the problem. The min-max problem is solved in a centralized manner due to its complexity and the large number of parameters involved. The solution provides topologies which minimize the traffic load of the most congested node in the network while meeting the constraints on the scatternet formation and network capacity. By varying the maximum number of piconets that can be created, they derived the performance of the attained solutions as the requirements on the throughput and on the role played by some of the network nodes change. The results can be used to find the optimal trade-off between system complexity and network efficiency. However, the optimization problem requires detailed system information and is not suited for a distributed implementation both for the algorithm characteristics and for its intrinsic complexity. It can only deal with a limited number of network nodes. Then, to overcome such a limitation, They discussed the key building blocks for a distributed solution approach to the scatternet formation problem. They outlined two procedures to handle the insertion and the removal of a node in/from the scatternet, in a distributed fashion. Both procedures aim at satisfying the Bluetooth technology constraints, while providing full network connectivity, high throughput, and reduced overhead due to control messages. Although these procedures may generate sub-optimal topologies, they can be easily implemented and are designed to deal with a large number of nodes.

Augel and Knoor [1] proposed a new approach of scatternet formation in which the formation is dependent on the QoS requirements of the applications. In their solution, to avoid larger degree which may cause bad influence on throughput, nodes with high degree stop paging and instructs a neighbor with a low degree to start paging instead. Each device may try to influence the topology depending on the QoS requirements. They describe a greneral scatternet formation design guidelines for QoS applications, and they did not present any particular scatternet formation protocol.

In [45, 23, 46], the scatternet formation issue in Bluetooth was discussed, by setting a framework for scatternet analysis based on a matrix representation, which allows developing and applying different metrics. They identified several metrics (capacity, average load, or path length) both in a traffic independent and in a traffic dependent context, and showed the relevant numerical results. In the traffic independent case, the scatternet is formed without knowledge of traffic relationships among involved devices. The scenario is described only by means of the adjacency matrix. If traffic relationships between nodes (e.g., flows at given data rates) have to be taken into account, they can be conveniently described by a traffic matrix. Then, a distributed algorithm for scatternet topology optimization, Distributed Scatternet Optimization Algorithm (DSOA), is introduced, that supports the formation of a locally optimal scatternet based on a selected metric. Numerical results obtained by adopting this distributed approach to optimize the network topology are shown to be close to the global optimum.

Cuomo, Melodia and Akyildiz [24, 22] extended their work and provided an integrated approach for scatternet formation and quality-of-service support (called SHAPER-OPT) by combining the tree-based scatternet formation algorithm *SHAPER* [21] and the distributed scatternet optimization algorithm (DSOA) [45, 23, 46]. The approach produces a meshed topology by applying DSOA on the network built by SHAPER. Performance evaluation of the proposed algorithms, and of the accordingly created scatternets, is carried out by using ns2 simulation. Devices are shown to be able to join or leave the scatternet at any time, without compromising the long term connectivity. Delay for network setup and reconfiguration in dynamic environments is shown to be within acceptable bounds. DSOA is also shown to be easy to implement and to improve the overall network performance.

4. Self-Routing in Bluetooth Scatternet. Routing in Bluetooth network received little attention so far. Though considerable research has been done in the area of routing in ad hoc networks, the direct application of this may be inefficient to Bluetooth scatternets. Some routing schemes [11, 55, 19, 35] for scatternets have been proposed recently.

Bhagwat and Segall [11] proposed a routing method, Routing Vector Scheme (RVM), in Bluetooth based on a concept of route vector. In their scheme, the complete path is carried in the header (i.e., source routing) and Bluetooth addresses are expressed very efficiently. They described the protocols for on-demand route discovery and packet forwarding. Their design illustrates three design compromises, namely minimization of soft-state, protocol simplicity, and bandwidth conservation, all of which are crucial for efficient operation over small size Bluetooth scatternets. However, due to carrying the complete path in the source routing, their scheme will lead to a large packet overhead in large Bluetooth scatternets, particularly since Bluetooth packets are very small.

Prabhu and Chockalingam [55] proposed a routing protocol that employs flooding to obtain battery levels of nodes. The protocol uses the available battery power in the Bluetooth devices as a cost metric in choosing the routes. We propose two techniques, namely a) battery power level based master-slave switch and b) distance based power control, to increase the network lifetime in scatternets. The master-slave switch technique is motivated by the fact that a piconet master has to handle the packet transmissions to/from all its slaves, and hence may drain its battery soon. By role switching, each device in a piconet may have to play the master role depending on its available battery power. In the second technique, devices choose their transmit powers based on their distances from their respective masters. Their performance results show that a considerable gain in network lifetime can be achieved using these two power saving techniques.

There are also some Bluetooth routing protocols by applying ad hoc routing protocols with adjustments to fit Bluetooth own characteristics. Choon-sik Choi and Hae-Wook Choi [19] proposed a Bluetooth routing protocol based on DSR (Dynamic Source Routing) [34]. Kapoor and Gerla [35] presented a routing scheme for Bluetooth scatternets based on ZRP (Zone Routing Protocol) [72] by customizing the scheme for use in Bluetooth scatternets.

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). Several scatternet formation algorithms consider routing issue in design. In the following, we will review those scatternet formation algorithms with self-routing properties.

4.1. Blue-tree Scatternet. Tree shaped scatternets are promising in terms of minimizing the number of piconets and allowing simpler routing protocols. BlueTree in [64] is capable of self-routing 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).

Let $ADDR(x)$ be node x's Bluetooth MAC address as defined in the Bluetooth Specification. Define $child(x)$ to be the set of children for a node x. Assuming the $ADDR$ has an total order, node x is before (after) node y if and only if $ADDR(x) < ADDR(y)$ ($ADDR(x) > ADDR(y)$). The $min(x)$ and $max(x)$ are defined to be the smallest and the largest $ADDR$ of the nodes in the subtree rooted at x. The $range(x) = (min(x), max(x))$ is defined as the address range of nodes in the subtree rooted at x. The $dist(x, range(y))$ is defined to be 0 if address of x falls in $range(y)$, i.e. $min(y) \leq ADDR(x) \leq max(y)$. Otherwise, it is defined to be the minimum of $(|ADDR(x) - min(y)|, |ADDR(x) - max(y)|)$.

The Blue-tree can be implemented by maintaining the following information at any node x:

1. An array of range for all x's children.
2. Its own $range(x) = (min(x), max(x))$.

The concept of Blue-tree is the extension of the binary search tree. If we consider the destination address as the key, then routing is just like doing a search. In message routing, an unique $ADDR$ is given and one is required to find the device that has this address along the tree. If we map a Blue-tree into a scatternet, each of the internal node is the master and its immediate children are its slaves. Based on the limit imposed by Bluetooth piconet, the maximal number of children is 7. The second property divides the nodes into subtrees in a special way, making the self-routing possible. If a scatternet is a Blue-tree, as long as each node maintains the ranges of itself and its children, routing can be done easily using the protocol similar as binary search. Please refer to [64] for more details.

Worth to mention that, Blue-tree scatternet relies on a sophisticated scatternet merge procedure with significant communication overhead for creation and maintenance.

4.2. BlueRing Scatternet. Bluerings as scatternets are proposed in [28, 42]. Ring structure for Bluetooth has simplicity, easy creation and easy routing as advantage. Routing on BlueRing is stateless in the sense that no routing information needs to be kept by any host once the ring is formed. This would be facorable for environments such as Smart Homes where computing capability is limited. In [42], the author presented the detailed routing protocol, which supports both unicasting and broadcasting on BlueRing. In the protocol, data packets will be routed foloowing the direction of the BlueRing. Since a packet flowing around the ring will eventually reach its destination piconet, no route discovery process is required. For detailed protocol, please refer to [42]. Notice, due to large diameter (i.e., the packet may reach its destination after travelling the whole ring), ring structure is only good for small-size or median-size scatternets.

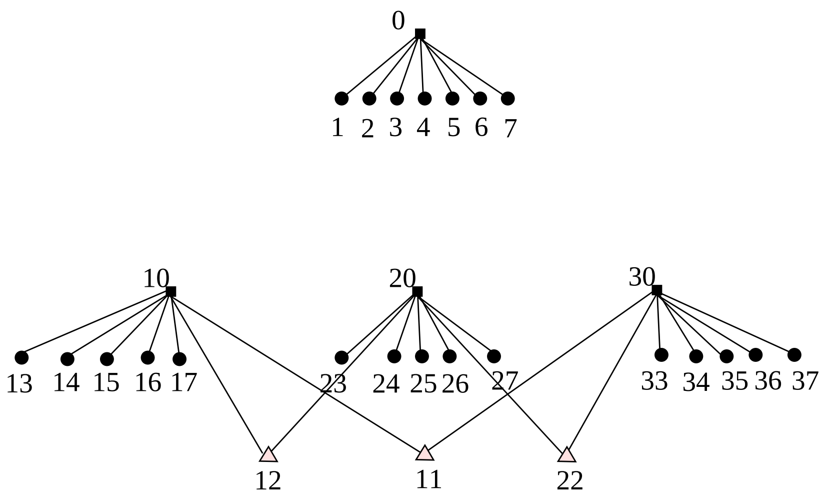


Fig. 4.1. *Node labeling for the projective scatternet.*

4.3. Projective Scatternet. Barriere et al. [3] also proposed a routing method for Bluetooth scatternets formatted by their method using their specific labels. The projective scatternet is constructed by using a basic procedure called Q-plication of scatternet S using a base scatternet N. The complete projective scatternet consists of many layers of scatternets. Each layer contains a finite number of scatternets, any one is obtained by a repeated Q-plication of the last element of the preceding level with a fixed base scatternet also taken from the preceding layer. The labeling rule of the scatternet is as follows. The master of a piconet is labeled 0, and the slaves are labeled arbitrarily from 1 to $p-1$, where p is a integer larger than q (a power of a prime). Figure 4.1 shows the illustration, where $p=8$ and $q=1$. Given the labeling of the nodes in a scatternet S, a node x of the Q-plication of S receives as label $L(x)$ a pair (i, l) where i indicates the index of the copy of S where x is located, $0 \leq i \leq Q-1$, and l is the label of x in this copy. A node resulting from the identification of $q+1$ free slaves receives a unique label (i, l) which is the smallest label among those $q+1$ slaves, according to the standard lexicographic ordering of the labels. In Figure 4.1, the upper figure shows the labels for a piconet, which is the base scatternet. The lower figure shows the labels for projective scatternet formed by Q-plication of the base scatternet. For example, for the node with label 11, it is slave for both nodes 10 and 30. So its label is the smallest label between 11 (its label in the 10's piconet) and 31 (its label in the 30's piconet).

Routing in the scatternet is based on following the path suggested by labels. Routing in a complete scatternet from a node x with label $a_k a_{k-1} \cdots a_1$ to a node y with label $b_k b_{k-1} \cdots b_1$ consists of routing from $a_k a_{k-1} \cdots a_1$ to a node $b_k c_{k-1} \cdots c_1$ to node $b_k b_{k-1} d_{k-2} \cdots d_1$, etc., each time matching one of the components in the label of the destination from the left until finally reaching the node $b_k b_{k-1} \cdots b_1$. Thus all we have to describe is how the routing proceeds from $a_k a_{k-1} \cdots a_1$ to a node $b_k c_{k-1} \cdots c_1$. Notice that the length of the label, k, is the level number in the construction of the scatternet. The routing algorithm works as follows. 1) Node x determines, using the block design information, the position of a slave z in its base scatternet of level k, that is shared by the two copies labeled by a_k and b_k. 2) x sends the message for y to z within the base scatternet shared by x and z. 3) z forwards the message to y within the subscatternet of level $k-1$ recursively. For example, if we want to send a packet from 27 to 15. First, we find the slave 12 in its base scatternet that shared by piconets 2 and 1. Then 27 sends the message to 12 via 20 within the base scatternet piconet 2. At last, the 12 send the message to 15 via 10 within the subscatternet of level 0. For more detailed labeling rules and routing method, please refer to [3].

4.3.1. dBBlue Scatternet. The dBBlue [61] scatternet first builds a backbone based on the well-known de Bruijn graph [25], then adds the remained nodes into the network with a flexible MAC assignment scheme to enable the self-routing in Bluetooth networks.

The de Bruijn graph, denoted by $B(d,k)$, is a directed graph with d^k nodes. Assume that each node is assigned a unique label of length k on the alphabet $\{0, \cdots, d-1\}$. There is an edge in $B(d,k)$ from a node with label $x_1 x_2 \cdots x_k$ to any node with label $x_2 \cdots x_k y$, where $y \in \{0, \cdots, d-1\}$. Figure 4.2 illustrates $B(2,3)$. It is well-known that the de Bruijn graph enables self-routing intrinsically. The self-routing path from the source with label $x_1 x_2 \cdots x_k$ to the target with label $y_1 y_2 \cdots y_k$ is $x_1 x_2 \cdots x_k \rightarrow x_2 \cdots x_k y_1 \rightarrow x_3 \cdots x_k y_1 y_2 \rightarrow \cdots \rightarrow x_k y_1 \cdots y_{k-1} \rightarrow y_1 \cdots y_k$. Observe that, we could find a shorter route by looking for the longest sequence that is both a suffix of $x_1 x_2 \cdots x_k$ and a prefix of $y_1 y_2 \cdots y_k$. Suppose that $x_i \cdots x_k = y_1 \cdots y_{k-i+1}$ is such longest sequence. The shortest path between the source and the target is $x_1 \cdots x_k \rightarrow x_2 \cdots x_k y_{k-i+2} \rightarrow x_3 \cdots x_k y_{k-i+2} y_{k-i+3} \rightarrow \cdots \rightarrow x_{i-1} \cdots x_k y_{k-i+2} \cdots y_{k-1} \rightarrow y_1 \cdots y_k$. Clearly, the route between any two nodes is at most k hops, i.e., $B(d,k)$ has diameter $k = \log_d n$, where $n = d^k$ is the number of nodes of the graph.

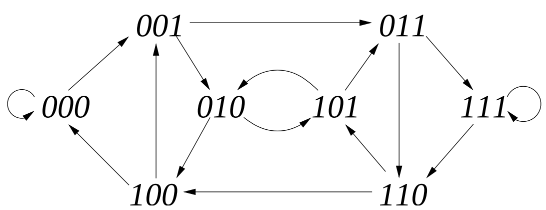


FIG. 4.2. *The de Bruijn graph $B(2,3)$.*

The classical de Bruijn graph is *balanced* in the sense that the labels of all nodes have the same length. A generalized de Bruijn graph is *pseudo-balanced* if the lengths of the labels are different by at most one. In dBBlue[61], they used a pseudo-balanced de Bruijn graph to be the backbone and handle the node leaving and joining. In a pseudo-balanced de Bruijn graph $B(2,k)$, each node has at most 4 out-neighbors and 2 in-neighbors. To route a packet from a node u with label $x_1x_2\cdots x_{s-1}x_s$ to another node v with label $y_1y_2\cdots y_{t-1}y_t$, where $s,t\in[k,k+1]$. Node u will forward the packet to its neighbor node with label $x_2\cdots x_{s-1}x_s$, or $x_2\cdots x_{s-1}x_sy_1$, or $x_2\cdots x_{s-1}x_sy_1y_2$. Notice that since the labels of the nodes are a universal prefix set, we know that *exactly* one of these three labels does exist. The following nodes keep forwarding the packet similarly until it reaches node v. Consequently, the diameter of pseudo-balanced de Bruijn graph is still $O(\log n)$. dBBlue[61] proposes a scalable scatternet structure based on pseudo-balanced de Bruijn graph $B(2,k)$.

In [61], the authors presented a novel rule of assigning the MAC address in a piconet. In the dBBlue scatternet, when we route a packet to a destination node v, we only know the piconet ID of node v, say $y_1y_2\cdots y_k$, which is same as the label of its master node, and the MAC address, say $z_1z_2z_3$, of this node in that piconet. When some node joins or leaves the scatternet, we often have to reorganize some piconets and thus re-assign the MACs of some nodes. The method of assigning MAC addresses in a piconet and reorganizing the piconets guarantees that the new piconet (even the new MAC address) can be found by a simple appending or deleting the least significant bit, which keeps the label prefix of updating nodes unchanged so that even the delivery of the packets on the way to those updating nodes will not be interrupted.

In a piconet, MAC 000 is always reserved by the master node. For simplicity, they omit the MAC address of a master node hereafter while representing its label, i.e., the master node with label $x_1x_2\cdots x_{m-1}x_m$ actually has a label $(x_1x_2\cdots x_{m-1}x_m,000)$ if consistent labels with slave nodes are needed. Remember that, in a pseudo-balanced de Bruijn graph, any node has 2 in-neighbors (except 0^m and 1^m) and at most 4 out-neighbors, so MAC addresses 011 and 111 are always reserved for the two bridge slaves to in-neighbors, MAC 010, 101, 001 and 110 are reserved for bridge slaves to out-neighbors if they exist, and 100 is reserved for the 7th slave (it must be a pure slave) if it exists. Figure 4.3 illustrates all four possibilities for the piconet MAC address assignment according to the number of out-neighbors in scatternet backbone. In the figure, for simplicity, we use $y_1y_2\cdots y_{m-1}y_m(y)$ to denote a node with label $y_1y_2\cdots y_{m-1}y_m$ or $y_1y_2\cdots y_{m-1}y_my$, whichever exists in the network. Notice that a master node in the constructed scatternet based on a pseudo-balanced de Bruijn graph $B(2,m)$ always has two incoming neighbors. For example, a master node $x_1x_2\cdots x_m$ in level m can have incoming neighbor $0x_1x_2\cdots x_{m-1}$ or $0x_1x_2\cdots x_m$, but not both since the de Bruijn graph is built upon a universal prefix set; similarly another incoming neighbor is $1x_1x_2\cdots x_{m-1}(x_m)$. Analogously, a master node $x_1x_2\cdots x_mx_{m+1}$ in level $m+1$ has incoming neighbors $0x_1x_2\cdots x_{m-1}(x_m)$ and $1x_1x_2\cdots x_{m-1}(x_m)$. On the other hand, the number of out-neighbors of a node in the pseudo-balanced de Bruijn graph $B(2,m)$ could be $1,2,3,4$. Only the node at level m could have 3 or 4 out-neighbors and only the node at level $m+1$ could have 1 out-neighbor (except nodes 0^m and 1^m if they exist).

Table 4.1 summarizes the rule of assigning the MAC address to the bridge slave nodes in a piconet. Their MAC addresses can be decided uniquely according to the label bit difference between current piconet and neighbor piconet IDs. For example, if the master u is labeled $x_1x_2\cdots x_s$ and its out-neighbor v is labeled $x_2\cdots x_sy_1y_2$, then the MAC addresses of their bridge slave is $y_1y_2\overline{y_2}$ assigned by u, and x_111 assigned

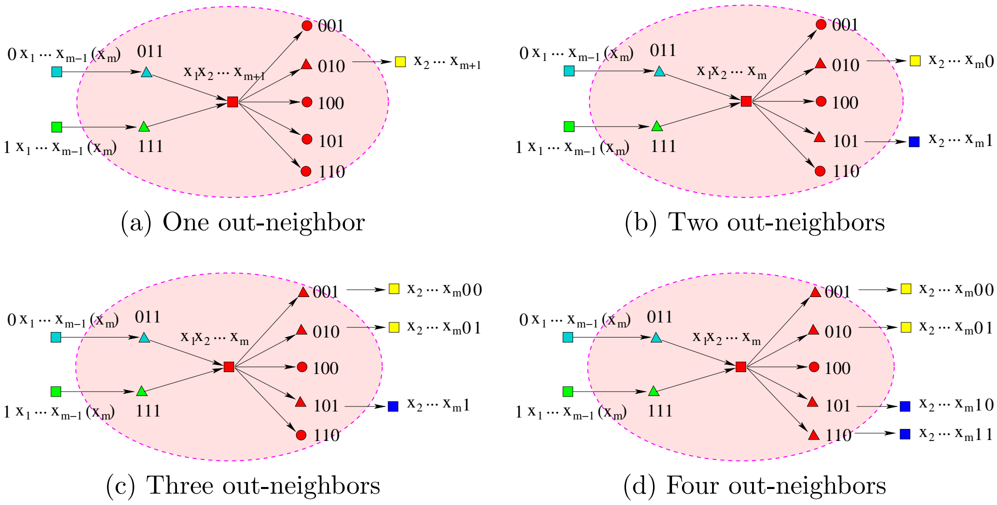


FIG. 4.3. *MAC address assignment for a piconet.*

by v. Remember that every bridge slave has one MAC address in each of the two piconets it resides.

TABLE 4.1
The rule to assign MAC address to bridge slave nodes.

Node	In-Neighbor	Out-Neighbor		
	$yx_1 \cdots x_r$	$x_2 \cdots x_s$	$x_2 \cdots x_s y_1$	$x_2 \cdots x_s y_1 y_2$
$x_1 \cdots x_s$	$y11$	010	$y_1 \overline{y_1} y_1$	$y_1 y_2 \overline{y_2}$

Notice that, in bluetooth scatternet, the bridge slave nodes have two independent piconet IDs and MAC addresses in two piconets respectively. However, since the routing mechanism in de Bruijn is directional, only their piconet ID and MAC address assigned by their in-master is public and meaningful for routing, saying *label* in the remaining paper, and the other one is only used for inter-communication in a piconet.

The updating of scatternet due to node joining or leaving is to maintain the pseudo-balanced de Bruijn based backbone, conforming the labelling rule as described before. More details, please refer to [61].

5. Conclusion. In this chapter, we reviewed different scatternet formation algorithms for both single-hop and multi-hop networks, and surveyed some Bluetooth routing algorithms and several scatternet topologies which have self-routing properties.

REFERENCES

[1] Markus Augel and Rudi Knorr. Bluetooth scatternet formation - state of the art and a new approach. In *Proc. of the 17th International Conference on Architecture of Computing Systems (ARCS), LNCS 2981*, Augsburg, Germany, 2004.

[2] Simon Baatz, Christoph Bieschke, Matthias Frank, Carmen Kuhl, Peter Martini, and Christoph Scholz. Building efficient Bluetooth scatternet topologies from 1-factors. In *Proceedings of the IASTED International Conference on Wireless and Optical Communications, WOC 2002*, Banff, Alberta, Canada, July 2002.

[3] Lali Barriere, Pierre Fraigniaud, Lata Narayanan, and Jaroslav Opatrny. Dynamic construction of Bluetooth scatternets of fixed degree and low diameter. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 781–790. Society for Industrial and Applied Mathematics, 2003.

[4] S. Basagni. Distributed clustering for ad hoc networks. In A. Y. Zomaya, D. F. Hsu, O. Ibarra, S. Origuchi, D. Nassimi, and M. Palis, editors, *Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN'99)*, pages 310–315, Perth/Fremantle, Australia, June 23–25 1999. IEEE Computer Society.

[5] S. Basagni, R. Bruno, and C. Petrioli. Performance evaluation of a new scatternet formation protocol for multi-hop Bluetooth networks. In *Proceedings of the 5th International Symposium on Personal Wireless Multimedia Communications, WPMC 2002*, pages 208–212, Honolulu, Hawaii, October 27–30 2002.

[6] S. Basagni and C. Petrioli. A scatternet formation protocol for ad hoc networks of Bluetooth devices. In *Proceedings of the IEEE Semiannual Vehicular Technology Conference, VTC Spring 2002*, Birmingham, AL, May 6–9 2002.

[7] Stefano Basagni, Raffaele Bruno, Gabriele Mambrini, and Chiara Petrioli. Comparative performance evaluation of scatternet formation protocols for networks of Bluetooth devices. *Wirel. Netw.*, 10(2):197–213, 2004.

[8] Stefano Basagni, Raffaele Bruno, and Chiara Petrioli. Scatternet formation in Bluetooth networks. in: Mobile Ad Hoc Networking, S. Basagni, M. Conti, S. Giordano and I. Stojmenovic (eds.), IEEE Press, Wiley Interscience.

[9] Stefano Basagni, Raffaele Bruno, and Chiara Petrioli. Device discovery in Bluetooth networks: A scatternet perspective. In *Proceedings of the Second International IFIP-TC6 Networking Conference on Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; and Mobile and Wireless Communications*, pages 1087–1092. Springer-Verlag, 2002.

[10] Stefano Basagni, Raffaele Bruno, and Chiara Petrioli. A performance comparison of scatternet formation protocols for networks of Bluetooth devices. In *Proc. IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2003.

[11] P. Bhagwat and A. Segall. A routing vector (RVM) for routing in Bluetooth scatternets. In *Proc. IEEE Int. Workshop on Mobile Multimedia Communications MoMuC*, 1999.

[12] Bluetooth SIG. Specification of the Bluetooth system. http://www.bluetooth.com/.

[13] Diego Bohman, Matthias Frank, Peter Martini, and Christoph Scholz. Performace of symmetric neighbor discovery in Bluetooth ad hoc networks. In *2nd German Workshop on Mobile Ad-hoc Networking (WMAN 2004)*, Ulm, Germany.

[14] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *3rd int. Workshop on Discrete Algorithms and methods for mobile computing and communications*, 1999.

[15] A. Busboom, Ian Herwono, M. Schuba, and G. Zavagli. Unambiguous device identification and fast connection setup in Bluetooth. In *Proceedings of the European Wireless 2002*, Florence, Italy, 2002.

[16] Chao-Tsun Chang, Chih-Yung Chang, and Jang-Ping Sheu. Bluecube: Constructing a hypercube parallel computing and communication environment over Bluetooth radio system. In *Proc. 2003 International Conference on Parallel Processing*, Kaohsiung, Taiwan, 2003.

[17] K. Cheolgi, M. Joongsoo, and L. Joonwon. A random inquiry procedure using Bluetooth. *IEICE Trans. Communications*, E-82(1), 1999.

[18] Carla-Fabiana Chiasserini, Marco Ajmone Marsan, Elena Baralis, and Paolo Garza. Towards feasible distributed topology formation algorithms for Bluetooth-based wpans. In *Proc. of 36th Hawaii International Conference on System Science (HICSS-36)*, Big Island, Hawaii, 2003.

[19] C.S. Choi and C.W. Choi. DSR Based Bluetooth Scatternet. In *ITC-CSCC 2002*, 2002.

[20] Ming-Te Chou and Ruay-Shiung Chang. Blueline: A distributed Bluetooth scatternet formation and routing algorithm. 2004.

[21] F. Cuomo, G. di Bacco, and T. Melodia. Shaper: a self-healing algorithm producing multi-hop Bluetooth scatternets. In *Proceedings of IEEE Globecom 2003*, San Francisco, CA, USA, December 2003.

[22] F. Cuomo, G. di Bacco, and T. Melodia. Optimized scatternet topologies for personal area networking in dynamic environments. In *Proceedings of IEEE IEEE International Conference on Communications (ICC 2004)*, Paris, France, June 2004.

[23] F. Cuomo and T. Melodia. A general methodology and key metrics for scatternet formation in Bluetooth. In *Proceedings of IEEE Globecom 2002*, Taipei, Taiwan, November 2002.

[24] F. Cuomo, T. Melodia, and I. F. Akyildiz. Distributed self-healing and variable topology

optimization algorithms for qos provisioning in scatternets. *IEEE Journal on Selected Areas in Communications*, 22(7):1220–1236, 2004.

[25] N.G. de Bruijn. A combinatorial problem. In *Koninklijke Nederlandsche Akademie van Wetenschappen*, July 1946.

[26] Yuhong Dong and Jie Wu. Three Bluetree formations for constructing efficient scatternets in Bluetooth. In *Proc. of the 7th Joint Conference on Information Sciences*, 2003.

[27] Fabrizio Ferraguto, Gabriele Mambrini, Alessandro Panconesi, and Chiara Petrioli. A new approach to device discovery and scatternet formation in Bluetooth networks. In *Proc. IEEE 18th International Parallel and Distributed Processing Symposium (IPDPS04)*, 2004.

[28] Cgun-Choong Foo and Kee-Chaing Chua. Bluerings - Bluetooth scatternets with ring structure. In *Proceedings of the IASTED International Conference on Wireless and Optical Communications, WOC 2002*, Banff, Alberta, Canada, July 2002.

[29] R. Guerin, E. Kim, and S. Sarkar. Bluetooth technology: Key challenges and initial research. In *Conference on Network and Distributed Simulations*, 2002.

[30] R. Guerin, J. Rank, S. Sarkar, and E. Vergetis. Forming connected topologies in Bluetooth adhoc networks. In *Proceedings of ITC'18*, Berlin, 2003.

[31] Jeffrey Hightower and Gaetano Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, 2001.

[32] Tsung-Chuan Huang, Chu-Sing Yang, Chao-Chieh Huang, and Shen-Wen Bai. Hierarchical grown Bluetrees (HGB) - an effective topology for Bluetooth scatternets. In *Proc. of International Symposium on Parallel and Distributed Processing and Applications (ISPA 2003), LNCS 2745*, Aizu, Japan, 2003.

[33] J.-R. Jiang, B.-R. Lin, and Y.-C. Tseng. Analysis of Bluetooth device discovery and some speedup mechanisms. *Int'l J. of Electrical Engineering*, 2004.

[34] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[35] Rohit Kapoor and Mario Gerla. A Zone Routing Protocol for Bluetooth scatternets. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC03)*, New Orleans, LA, 2003.

[36] Brad Karp and H.T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, 2000.

[37] Yoji Kawamoto, Vincent W.S. Wong, and Victor C.M. Leung. A two-phase scatternet formation protocol for Bluetooth wireless personal area networks. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC03)*, New Orleans, LA, 2003.

[38] 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*, pages 183–192, 2001.

[39] Xiang-Yang Li and Ivan Stojmenovic. Partial delaunay triangulation and degree limited localized Bluetooth scatternet formation. In *AdHocNow*, 2002.

[40] Xiang-Yang Li, Ivan Stojmenovic, and Yu Wang. Partial delaunay triangulation and degree limited localized Bluetooth multihop scatternet formation. *IEEE Transaction on Parallel and Distributed Systems*, 15(4):350–361, 2004. The short version appeared at AdHocNow 2002.

[41] Chunhung Richard Lin and Mario Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal of Selected Areas in Communications*, 15(7):1265–1275, 1997.

[42] T. Y. Lin, Y. C. Tseng, and K. M. Chang. A new Bluering scatternet topology for Bluetooth with its formation, routing, and maintenance protocols. *Wireless Communications and Mobile Computing*, 3(4):517–537, June 2003.

[43] Yong Liu, M.J. Lee, and T.N. Saadawi. A Bluetooth scatternet-route structure for multihop ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 21(2):229– 239, 2003.

[44] M.A. Marsan, C.F. Chiasserini, A. Nucci, G. Carello, and L. de Giovanni. Optimizing the topology of Bluetooth wireless personal area networks. In *INFOCOM*, 2002.

[45] T. Melodia and F. Cuomo. Ad hoc networking with Bluetooth: Key metrics and distributed protocols for scatternet formation. *Ad Hoc Networks (Elsevier)*, 2(2):185–202, April 2004.

[46] T. Melodia and F. Cuomo. Locally optimal scatternet topologies for Bluetooth ad hoc networks. In *Proceedings of First Working Conference on Wireless On-demand Network Systems (WONS 2004)*, Madonna di Campiglio, Italy, January 2004.

[47] D. Miorandi, Arianna Trainito, and A. Zanella. On efficient topologies for Bluetooth scatternets. In *Proc. IFIP-TC6 8th International Conference, PWC 2003, LNCS 2775*, Venice, Italy, 2003.

[48] D. Miorandi and A. Zanella. On the optimal topology of Bluetooth piconets: Roles swapping algorithms. In *Proc. Mediterranean Conference on Ad Hoc Networks (MedHoc)*, Sardinia, Italy, 2002.

[49] P. Murphy, E. Welsh, and P. Frantz. Using Bluetooth for Short-Term Ad-Hoc Connections Between Moving Vehicles: A Feasibility Study. In *IEEE Vehicular Technology Conference (VTC)*, volume 1, Birmingham, AL, May 2002.

[50] Elena Pagani, Gian Paolo Rossi, and Stefano Tebaldi. An on-demand Bluetooth scatternet formation algorithm. In *Proceedings of First Working Conference on Wireless On-demand Network Systems (WONS 2004)*, Madonna di Campiglio, Italy, January 2004.

[51] K. Persson, D. Manivannan, and M. Singhal. Bluetooth scatternet formation: Criteria, models, and classification. In *IEEE Consumer Communications and Networking Conference 2004 (CCNC'04)*, Las Vegas, NV, 2004.

[52] C. Petrioli and S. Basagni. Degree-constrained multihop scatternet formation for Bluetooth networks. In *Proceedings of IEEE Globecom 2002*, pages 222–226, Taipei, Taiwan, R.O.C., November 17–21 2002.

[53] C. Petrioli, S. Basagni, and I. Chlamtac. Configuring Bluestars: Multihop scatternet formation for Bluetooth networks. *IEEE Transactions on Computers*, 52(6):779–790, 2003.

[54] C. Petrioli, S. Basagni, and I. Chlamtac. Bluemesh: degree-constrained multi-hop scatternet formation for Bluetooth networks. *Mobile Networks and Aplications*, 9(1):33–47, 2004.

[55] Balakrishna J. Prabhu and A. Chockalingam. A Routing Protocol and Energy Efficient Techniques in Bluetooth Scatternets. In *IEEE International Conference on Communications (ICC 2002)*, New York, NY, April 2002.

[56] Lakshmi Ramachandran, Manika Kapoor, Abhinanda Sarkar, and Alok Aggarwal. Clustering algorithms for wireless ad hoc networks. In *Proceedings of the 4th international workshop on Discrete algorithms and methods for mobile computing and communications*, pages 54–63. ACM Press, 2000.

[57] Miklos Aurel Ronai and Eszter Kail. A simple neighbour discovery procedure for Bluetooth ad hoc networks. In *proceedings of IEEE GlobeCom 2003*, San Francisco, CA, USA.

[58] T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaire. Distributed topology construction of Bluetooth personal area networks. In *Proc. IEEE INFOCOM*, 2001.

[59] Theodoros Salonidis, Pravin Bhagwat, and Leandros Tassiulas. Proximity awareness and fast connection establishment in Bluetooth. In *Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*, pages 141–142. IEEE Press, 2000.

[60] Frank Siegemund and Michael Rohs. Rendezvous layer protocols for Bluetooth-enabled smart devices. *Personal Ubiquitous Comput.*, 7(2):91–101, 2003.

[61] Wen-Zhan Song, Xiang-Yang Li, Yu Wang, and Weizhao Wang. dBBlue: Low diameter and self-routing Bluetooth scatternet. *Elsevier Journal of Parallel and Distributed Computing (JPDC)*, 2004. accepted for publication. The short version appeared at ACM DIALM 2003.

[62] Hiranmayi Sreenivas and Hesham Ali. An evolutionary Bluetooth scatternet formation protocol. In *Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 9*, page 90306.3. IEEE Computer Society, 2004.

[63] I. Stojmenovic and N. Zaguia. Bluetooth scatternet formation in ad hoc wireless networks. in: Performance Modeling and Analysis of Bluetooth Networks: Network Formation, Polling, Scheduling, and Traffic Control, J. Misic and V. Misic (eds.), CRC Press.

[64] M.T. Sun, C.K. Chang, and T.H. Lai. A self-routing topology for Bluetooth scatternets. In *Proc. IEEE International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '02)*, Philipini, 2002.

[65] G. Tan, A. Miu, J. Guttag, and H. Balakrishnan. Forming scatternets from Bluetooth personal area networks. Technical Report MIT-LCS-TR-826, MIT, 2001. Also in First Annual Student Oxygen Workshop, Gloucester, MA, July 2001.

[66] Yu Wang, Ivan Stojmenovic, and Xiang-Yang Li. Bluetooth scatternet formation for single-hop ad hoc networks based on virtual positions. 2004. Journal of Internet Technology (JIT), accepted for publication. The short version appeared at IEEE ISCC 2004.

[67] Z. Wang, R.J. Thomas, and Z. Haas. Bluenet – a new scatternet formation scheme. In *Proceedings of the 35th Hawaii International Conference on System Science (HICSS-35)*, Big Island, Hawaii, 2002.

[68] E. Welsh, P. Murphy, and P. Frantz. Improving Connection Times for Bluetooth Devices in Mobile Environments. In *International Conference on Fundamentals of Electronics Communications and Computer Sciences (ICFS)*, March 2002.

[69] Ryan Woodings, Derek Joos, Trevor Clifton, and Charles D. Knutson. Rapid heterogeneous connection establishment: Accelerating Bluetooth inquiry using IrDA. In *Proceedings of the Third Annual IEEE Wireless Communications and Networking Conference (WCNC*

'02), 2002.

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

[71] G.V. Zaruba, S. Basagni, and I. Chlamtac. Bluetrees - scatternet formation to enable Bluetooth based ad hoc networks. In *Proc. IEEE International Conference on Communications(ICC)*, 2001.

[72] Haas Z.J. and Pearlman M.R. The zone routing protocol(zrp) for ad hoc networks. In *Internet draft - Mobile Ad hoc NETworking (MANET), Working Group of the Internet Engineering Task Force (IETF)*, Novermber 1997.