Bluetooth scatternet formation
in ad hoc wireless networks

Ivan Stojmenovic and **Nejib Zaguia**

SITE, University of Ottawa
Ottawa, Ontario K1N 6N5, Canada
ivan@site.uottawa.ca zaguia@site.uottawa.ca

1. Introduction

1.1. Summary

Bluetooth standard allows the creation of piconets, with one node serving as its master and up to seven nodes serving as slaves. Additional slaves must be parked, with significant overhead involved for parking and unparking them. Although the standard allows for the creation of a collection of connected piconets, called scatternet,, it does not give any particular protocol for it. In a unit disk graph, two nodes can communicate with each other if and only if the distance between them is at most *R*, where *R* is transmission radius which is equal for all nodes. Given a set of Bluetooth nodes which are positioned so that their unit disk graph is connected, the *Bluetooth scatternet formation* (BSF) problem is to select piconets, and master and slave roles in each piconet, so that the obtained scatternet is connected, has some desirable properties and with a good performance with respect to some metrics. This chapter surveys the solutions proposed so far in the literature for the BSF problem.

1.2. Related scheduling problem

When two Bluetooth devices establish communication, one of them assumes the role of a *master* node while the other is a *slave* node. Two nodes in a scatternet can communicate by finding a route between them, where each hop is a master-slave pair of nodes from the same piconet. A node may serve as the master in at most one piconet, and as a slave in unlimited number of other piconets. However, while it serves as slave in other piconet, its own piconet (if it has master role in any of them) will be idle. Obviously, there are problems of *scheduling* transmissions and the time division for each master and slave node so that the overall operation is synchronized and the delay is minimized. The scatternet characteristics will have a direct impact on performance of scheduling protocols. Miklos et all [MRTVJ] concluded that piconet switching poses a significant overhead and has a major impact on system performance. It is therefore important for overall scatternet performance not only that scatternet topology is carefully constructed, but also that piconet switching is scheduled as efficiently as possible. This chapter is concerned only with the Bluetooth scatternet formation (BSF) problem. The scheduling problem is a non-trivial one and it is discussed elsewhere in the same book as this chapter.

1.3. Preliminary taxonomy of BSF protocols

We will describe now a classification taxonomy for the most known Bluetooth scatternet formation (BSF) algorithms. The main criteria used to evaluate these protocols will be on how they achieved the main mission, of providing a connected and a degree limited scatternet topology. Starting from a connected unit disk communication network and assuming that each node is aware of all its neighbours within a communication range, the algorithms will be classified into those that guarantee connectivity and those that do not guarantee connectivity,. Obviously, there is tiny probability that two nodes will never find each other, therefore connectivity cannot be guaranteed in that sense even for a network consisting of two nearby nodes. Thus the assumption is natural, and connectivity is judged subject to established neighbours knowledge.

The next classification is based on observing degree limitation. The protocols will be divided into those that guarantee degree limitation for each created piconet (that is, always no more than seven slaves for each master) and those that do not guarantee degree limitation.

Existing protocols may also be divided into those that work properly and are designed for the single-hop scenarios only, where each device is within communication range of any other device in the network. In this case, and using the graph terminology, the unit disk communication graph is a complete graph. More general protocols can be applied for single-hop scenarios as well, but they are designed to work properly for arbitrary type of unit disk graphs, that is, for multi-hop scenarios, where some nodes are not within transmission range of each other, but are connected via other nodes in multi-hop fashion. This chapter will primarily classify the existing protocols into these two categories, and describe them in separate sections.

The protocols can also be divided into those that require that each node learns about all its neighbours for proper functioning, and those that decide about scatternet links after learning about some of its neighbours. More classification criteria will be listed in the sequel.

1.4. Device discovery

A closely related problem is *neighbour discovery (*or *device discovery),* that is, how two nodes find each other and establish communication. Almost all proposed solutions assume that Bluetooth technology is used for both neighbour discovery and data communication in the created scatternets. Most BSF protocols use the following device discovery scheme which is described in [SBTL]. The device discovery is performed by each node randomly entering into an *inquiry* or an *inquiry scan mode* (with equal probabilities), and randomly selecting the time length for being in the mode repeatedly until a timeout expires (the timeout should be carefully selected to enable one hop information with high probability, but within reasonable time). Inquiry nodes select a repeated pattern of 32 frequencies (out of a total of nearly hundred available frequencies) and send signal on selected frequency in given spot. Inquiry scan nodes also select a frequency at random in each spot and listen to the transmission at the selected frequency. The discovery (and establishment of master-slave relationship) occurs when both sender and receiver nodes are at the same frequency. The timeout for overall device discovery protocol (for finding sufficient number of neighbors) is experimentally determined to have the best value of about 8 seconds. A different 'recipe' is proposed in [FMPP] where each device executes the device discovery protocol until it is connected with *c* neighbors (*c* is between 5 and 7). If the visibility graph (normally it is the unit disk graph) is connected, then the resulting graph is experimentally shown to be connected with high probability.

Basagni, Bruno and Petrioli [BBP5] studied the reasons for the Bluetooth based device discovery being so inefficient. The overall connectivity is established fairly quickly, but the full awareness of all neighbors is slow. By means of a thorough ns2 based performance evaluations the authors identified at least two major features of the Bluetooth specifications that are responsible for this lack of performance. First the overly long backoff intervals adds a considerable time to every handshake, and second the impossibility of the node in inquiry mode to identify itself in the ID packet, resulting in handshakes between nodes that have already discovered each other.

Joung and Huang [JH] proposed to modify device discovery protocol by using node's ID to decide a pseudo-random sequence, so that scan or inquiry state is decided deterministically. When two nodes discover each other, smaller ID node replaces its ID with the ID of the other node. They discussed timeout duration to guarantee the connectivity, which depends on diameter, since one node with largest ID will propagate its ID to all the other nodes in the network.

Bluetooth and Wi-Fi are two widely adopted technologies for wireless communication between two nodes. They are competing and complementary at the same time, since Bluetooth is more suitable for short communications and provides direct communication between any two nodes, while Wi-Fi requires access point and generally is applied on somewhat longer distances. Many equipments use both Bluetooth and Wi-Fi as two independent applications, which may cause interference when both are running. Some latest products allow them to be synchronized based on time slicing technique. Further step would be to provide softwares that will further synchronize the two technologies and allow, for instance, that two nodes discover each other using Wi-Fi technology and then communicate by Bluetooth chips. This device discovery protocols has been proposed in [LSW]. One of route discovery schemes for on-demand BSF construction in [LLS] also uses single channel communication for broadcasting the destination search packet. The use of WiFi or another single channel medium access technology for device discovery is a predicable approach since Bluetooth based neighbour discovery is proven to be slow, especially when each device is required to discover all its neighbours before a good scatternet can be created. Therefore neighbour discovery phase of scatternet formation can be classified as being Bluetooth or Wi-Fi based.

1.5. Communication and time requirements

A formation algorithm should not depend on a central component for otherwise it will contradict the character of ad hoc networks. When Bluetooth is applied on a hybrid ad hoc network, that is, a network attached to a fixed infrastructure (e.g. Internet) the point of attachment may run a centralized algorithm and distribute master-slave decisions to each node. A single node in ad hoc networks may also collect all the information and run a centralized algorithm. A better approach is to design a formation algorithm in a decentralized and distributed manner. Distributed algorithms can be further classified into globalized and localized protocols. In a *localized protocol*, each node makes formation decisions solely based on the information from its neighbours (possibly also 2-hop neighbours). Localized protocols are further divided into local and quazi-local based on maintenance cost. The algorithms should support mobile devices, and devices entering and leaving the network. When nodes move, appear or disappear from the network, the scatternet maintenance (or self-healing) protocol is required. If the changes made by a single node have impact only on piconets associated with 1-hop and 2-hop neighbours then the protocol is local. Otherwise (e.g. when local change may sometimes trigger global updates, that is, cause a 'chain' effect) the protocol is quazi-local. An implicit solution, as in [LSW], is to apply localized schemes that provide local maintenance,.

Further division can be made according to *message complexity*, *time complexity* and *memory requirements.* In general, the protocols along these lines can be classified as being constant and non-constant size with respect to the number of nodes in the network. The message complexity is particularly important, since it is directly related to the energy needs of the algorithm. The QoS requirements of the application may also be considered. The topology could be set in such a way that the QoS requirements of the user applications in the network are met. The scatternet formation problem appears very difficult even without QoS considerations. Existing work on creating scatternets that provide QoS guarantees appears either centralized or has considerable communication overhead. Therefore QoS provision largely remains for future studies.

1.6. Scatternet design criteria

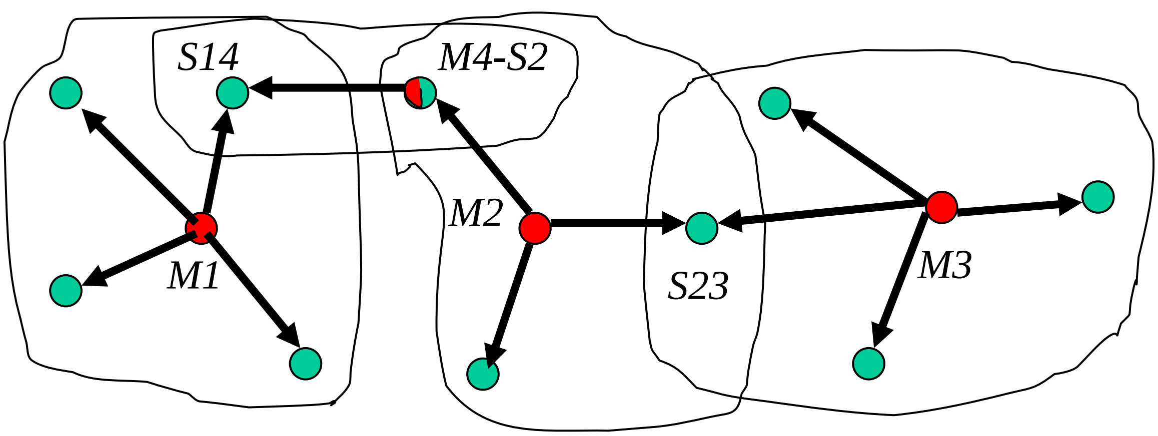

Figure 1. A scatternet consisting of four piconets

Figure 1 illustrates a scatternet with four piconets. Master nodes are labelled *M1-M4* and the slave nodes have indices that correspond to the piconet numbers (for example *S14* is a slave of piconets 1 and 4). Common slave *S23* serves as a bridge to connect piconets 2 and 3. In order to connect piconets 1 and 2, a (new) piconet 4 is created. Node *M4-S2* is its master, and plays also the role of a bridge node for two piconets and serving as slave node in piconet 2.

The bridges participate in piconets on time division basis. It is therefore anticipated that scatternet formation protocols should have (at least) the following goals in mind:

- minimization of the number of piconets, and therefore the number of master nodes;
- minimization of the number of slave roles for each node (a rough division is into protocols with <u>constant</u> and <u>non-constant maximum number of slave roles</u> for each node);
- minimization of the number of master-slave bridge nodes; a rough division is into those that <u>do not have master-slave bridges</u> (bipartite graphs) and those that <u>allow master-slave nodes</u>).

There are various metrics that can be used for evaluating scatternets. For example, Melodia and Cuomo [MC1] favoured scatternets with maximum capacity, scatternets with maximum residual capacity or minimum average load, and metrics associated with path lengths (average path length, average path capacity). Capacity related metrics may require a priori knowledge of traffic demands. Persson, Manivannan and Singhal [PMS] listed the following criteria for constructing scatternets: complete scatternet connectivity, maximized aggregate bandwidth, minimized average routing path length, maximized average node availability, minimized bridge switching overhead, communication group clustering, self-healing, multi-hop node participation, and on-demand scatternet formation. This survey concentrates on traffic independent measures when evaluating the performance of scatternets. Hodge and Whitaker [HW] listed the following such measurements: number of piconets, average number of slaves per piconet, average number of roles per device, average number of bridges per piconet, average number of bridges between piconets, number of master-slave bridges, average shortest-path length, bottleneck and average path latency.

2. BSF in single-hop networks

In a single-hop ad hoc network (or a complete unit disk graph), all wireless devices are in the radio vicinity of each other, e.g., electronic devices in a laboratory, or laptops in a conference room. In this section, we only focus on designing scatternet formation algorithms for single-hop networks. Note that the initial single-hop network, after creating scatternet, is converted into a multi-hop scatternet.

2.1. Centralized BSF protocols for complete graphs

2.1.1. Traffic and capacity based scatternets

Miorandi, Trainito, and Zanella [MTZ] investigated the relationship between the network capacity and the topology for Bluetooth scatternets. They started by considering the intrinsic capacity limits of a scatternet structure, and showed that limiting capacity may be achieved for very local traffics and under specific conditions on the scatternet structure. Then, they provided a description of the performance achievable with two basic scatternet configurations, namely star and closed–loop topologies, and then showed the role played by inter–piconet interference in the choice of efficient configurations. Finally, they presented some efficient topologies, based on Platonic solids structures. A centralized BSF solution for single-hop networks, where the traffic between any pair of nodes is known a priori, is described by Miorandi and Zanella [MZ].

2.1.2. Super-master election for central decisions

Bhagwat and Rao [BR] described an efficient technique for enumerating all feasible Bluetooth scatternet topologies as well as several constrained subsets of topologies. These results are useful in the design of optimization algorithms for Bluetooth networks.

Salonidis, Bhagwat, Tassiulas, and LaMaire [SBTL] proposed a BSF topology construction algorithm which 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. In the process, one leader , for each connected component is elected. Then the leader collects the information about the whole network, decides the roles for each node, and distributes back the roles. Since it is a centralized approach this the solution is not scalable, and not localized. Moreover, [SBTL] did not elaborate in their paper how to assign the roles in. And they also assumed that the network could have up to 36 nodes.

Huang, Chen, Sivakumar, Kashima, and Sezaki [HCSKS], argue that Bluetooth characteristics preclude the formation of very big network, and propose a centralized scatternet formation scheme that is optimized for conference scenarios. Only one node, super-master, will do inquiry continuously while all the other nodes will be continuously in an inquiry scan mode. The super-master makes topology decisions based on a load metric instead of the hop count metric, and communicates them to the other nodes. It needs to now the number of nodes in the network (and enter it at the beginning of the protocol) to create scatternet.

Tree structure is not a desirable scatternet topology since there is exactly one path between any two nodes, and failure of a single node on the path will disconnect the network, moreover the root node is likely to be a bottleneck.

2.1.3. Ring topology

Lin, Tseng and Chang [LTC] described a *ring topology* for scatternet formation in single-hop networks. A ring is created by the master nodes and the slave-slave bridges.

Each master can have further slaves outside the ring. The protocol uses park mode, and it is centralized. Therefore it is not scalable and the park mode introduces long message delays. Foo and Chua [FC] presented a similar ring structure consisting of master-slave bridges only, which simplifies the routing and offers larger fraction of bandwidth to each device. Ring structure for Bluetooth has the simplicity and the easy creation as an advantage, but it suffers from a large diameter (i.e., the maximum number of hops between any two devices) and a large number of piconets.

2.2. Distributed BSF protocols for complete graphs

2.2.1. Tree scatternet structure

Law, Mehta and Siu [LMS] described an algorithm that creates connected degree bounded scatternet in single-hop networks. The final structure is a tree like scatternet, which limits efficiency and robustness.

Sun, Chang and Lai [SCL] described a self-routing topology for single-hop Bluetooth networks. Nodes are organized and maintained in a search tree structure, with Bluetooth ID’s as keys (these keys are also used for routing). It relies on a sophisticated scatternet merge procedure with significant communication overhead for creation and maintenance. The procedure in [SCL] generates maximally filled piconets.

Tan, Miu, Guttag and Balakrishnan [TMGP] proposed a method for multi-hop networks and which is restricted to single-hop scenarios and where every node is active in at most two piconets. This method is similar to the one proposed in [ZBC] (described below),

2.2.2. Mimicking known topologies

A single-hop Bluetooth scatternet formation scheme based on 1-factors is described by Baatz, Bieschke, Frank, Martini, Scholz and Kuhl [BBFMS]. However, piconets are not degree limited in that scheme.

Barriere, Fraigniaud, Narajanan, and Opatrny [BFNO] described a connected degree limited and distributed scatternet formation solution based on projective geometry for single-hop networks. They assume that only slave nodes can act as bridges. They described procedures for adding and deleting nodes from the networks and claimed that it uses $O(\log^4 n \log^4 \log n)$ messages and $O(\log^2 n \log^2 \log n)$ time in local computation, where n is the number of nodes in the network. The degree of the scatternet can be fixed to any $q + 1$, where q is a power of a prime number. However, in their method, every node need to hold the information of the projective plane, and the master node who has the ”token” needs to know the information of the projective scatternet (i.e. which label should be used for the new coming master and which existing nodes need to be connected to it). In [BFNO], 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. Also, notice that the method has large overhead for construction and maintenance.

Daptardar [D] proposed to use cube structure in two and three dimensions for creating scatternets in single-hop scenarios. The author [D] argues that the structure provides higher connectivity, lower diameter, less node contention, multiple paths between any two nodes, in-built routing, easy inter-piconet scheduling and the ability to reconfigure for dynamic environments.

Song, Li, Wang and Wang, [SLWW] adopted the well-known *de Bruijn graph* structure 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 of the master and slave roles. Their structure *BBlue* also enjoys a nice routing property: the diameter of the graph is O(log *n*) and there exists a path with at most O(log *n*) hops for every pair of nodes without any routing table. Moreover, the congestion of every node is at most O((log *n*)/*n*), assuming that a unit of total traffic demand is equally distributed among all pair of nodes. In the same paper, the authors [SLWW] discuss in detail 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 nodes affected when a node joins or leaves the network is always bounded from above by a constant. To facilitate self-routing and easy updating, the authors design a scalable MAC assigning mechanism for piconet, which guarantees the packet delivery during scatternet updating. The *dBBlue* scatternet can be constructed incrementally when the nodes join the network one by one. The proposed method therefore has a number of desirable characteristics.

2.2.3. Minimal spanning tree based scatternets

Wang, Stojmenovic and Li [WSL] addressed the problem of scatternet formation for single-hop Bluetooth based personal area and ad hoc networks, with a minimal communication overhead. The scatternet formation schemes by Li, Stojmenovic and Wang [LSW] (described below) are position based and were applied for multi-hop networks. These schemes are localized and can construct degree limited and connected piconets, without parking any node. They also limit to 7 the number of slave roles in one piconet. The creation and maintenance require small overhead in addition to maintaining location information for one-hop neighbors. In [WSL], the authors apply this method to single-hop networks, by showing that position information is then not needed. Each node can simply select a virtual position, and communicate it to all neighbors in the neighbor discovery phase. Nodes then act according to the scheme [LSW] using such virtual positions instead of real ones. In addition, [WSL] used Delaunay triangulation instead of partial Delaunay triangulation proposed in [LSW], since each node has all the information needed. Likewise, [WSL] applied minimum spanning tree (MST) as the planar topology in their schemes. The experiments [WSL] confirm good functionality of the created Bluetooth networks in addition to their fast creation and straightforward maintenance. If MST is used as the scatternet topology, some long edges can be added to provide shorter routes, following the suggestions given in [S1].

2.2.4. Loop scatternet structure

Zhang, Hou and Sha [ZHS] proposed a BSF scheme for single-hop scenario. The new loop scatternet structure [ZHS] preserves connectivity and maximum node degree, and minimizes number of piconets. Additionally, it incurs a much smaller network diameter and much smaller maximum node contention. The main idea in [ZHS] is to create smaller scatternet structures, and to make some changes in master-slave relations whenever two such scatternets merge into one. The goal is to create a loop rather than a tree like structure. The final structure has a form of a loop, with a number of additional slave nodes attached to loop masters. In the first phase, piconets are created, with at most six slaves each. In the second phase, a slave from each piconet is explicitly selected and shared with another piconet to reduce the diameter. The protocol creates only slave-slave bridges. The authors did clearly show, theoretically and experimentally, the advantages of this new method.

2.2.5. On-demand scatternet formation and maintenance

Sivakumar, Chen and Huang [SCH] proposed a framework for continuously optimizing the network topology in order to produce the best suitable one for current data streams. The optimization process also takes care of the network maintenance to accommodate the node mobility. A high level description of a protocol is designed for single-hop networks to follow the framework (however a precise protocol description is missing).

3. BSF in multi-hop networks

3.1. Centralized algorithms

Ajmone-Marsan et al. [ACNCG] described a centralized solution for finding a Bluetooth topology (for multi-hop case) 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. The solution is based on a linear optimization formulation, and the formulation leads to an NP-complete problem suited only for small and stationary networks.

Sreenivas and Ali [SA] described a centralized BSF protocol based on a genetic algorithm, which selects random groups of nodes as an initial population. Each group corresponds to a combination of masters, slaves and bridge nodes and is represented by a string. The goal is to minimize the number of piconets created. The protocol is described on a half page without sufficient details on the population coding (master-slave relations are apparently not coded according to given description), crossover, mutation, and fitness function used. Genetic algorithm approach to scatternet formation is also suggested in [HW].

Mehta and El Zarki [MeZ] outlined an approach, centered on the Bluetooth technology, to support a sensor network composed of fixed wireless sensors for health monitoring of highways, bridges and other civil infrastructures. They present a topology formation scheme that not only takes into account the traffic generated by different sensors but also the associated link strengths and buffer capacities. The algorithm makes no particular assumptions as to the placement of nodes, nor the assumption that nodes need to be in radio proximity of each other. The output is a tree shaped scatternet rooted at the sensor hub (data logger), that is balanced in terms of traffic carried on each of the links. The solution is centralized (data logger collects network information and makes all decisions), and is based on a combinatorial optimization formulation followed by simulated annealing based solution.

Yun, Kim, Kim and Ma [YKKM] presented an approach that forms master-slave mesh topology. Their Bluestars approach models the discovery neighbourhood as an inquiry graph *I*, with $2^{|I|}$ topology subsets available. The solution requires a costly determination of the optimal topology subset (presumably in a centralized fashion) before the scatternet is formed.

Ramachandran, Kapoor, Sarkar, and Aggarwal [RKSA] proposed a BSF algorithm based on growing a tree from the root, where master node is not always directly connected to its slave node. They presented a deterministic as well as a randomized algorithm. Both approaches involve a leader election of a super-master, which subsequently forms the actual topology in a centralized manner.

A preliminary account on how to deal with changing network topology has been presented by Chiasserini, Ajmone-Marsan, Baralis, and Garza [CABG]. They presented an optimized approach for scatternet formation that attempts to minimize the traffic load. The authors assume that traffic patterns and routes are known a priori and formalize the topology formation as a min-max problem, which finds a bottleneck node and minimizes the traffic load at that node. They also discussed a distributed approach.

3.2. Growing tree based distributed BSF

Zaruba, Basagni and Chlamtac [ZBC] proposed two protocols for forming connected scatternet. In both cases, the resulting topology is termed a bluetree. The number of roles each node can assume is limited to two or three. The first protocol is initiated by a single node, called the blueroot, which will be the root of the bluetree. A rooted spanning tree is built as follows. The root will be assigned the role of a master node. Every one hop neighbor of the root will be its slave. The children of the root will now be 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 a slave for only one master, the one that paged it first. Each internal node of the tree is a master on one piconet, and a slave of another master (its parent in the initial tree). In order to limit the number of slaves, the authors [ZBC] observed that if a node in the unit disk graph has more than five neighbors, then at least two of them must be connected. This observation is used to reconfigure 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 the master of s_1, and then disconnects s_2 from itself. Such branch reorganization is carried throughout the network.

In [DW], Dong and Wu proposed three modifications to the Bluetree algorithm in [ZBC]. These modifications aim to minimize the overheads introduced by Bluetooth's piconet and multi-hop scatternet. Their modified algorithms use the neighbor's neighbor set and/or neighbor's location to construct the Bluetree to efficiently balance two conflicting goals between the number of piconets and the average shortest path ratio. The modifications are to select as a bridge the slave node that has maximal degree, to select closest slave and instruct it to become master, and then to apply the Yao structure (similar to the one in [LSW]).

Pagani, Rossi and Tebaldi [PRT] proposed an improvement to the Bluetree algorithm in [ZBC], in order to address the issues involved in practical implementation, and to describe mechanisms to support mobility, joining and leaving the network. Their proposed solution is an on-demand BSF algorithm. The main improvement is to start the scatternet formation when needed, by an initiator node that becomes the tree root, while the other nodes progressively join the tree so that overall structure is optimized with respect to the latency in the data forwarding. and on-demand formation of scatternet.

Huang, Yang, Bai and Huang [HYBH] modified bluetree scatternet formation scheme [ZBC] by limiting the number of children slaves of each node to 5, and using two more slave roles for up to two siblings. Thus additional links between nodes on the same level are added in the tree. This creates a structure with less critical links for connectivity, but the maintenance is still expensive like in the bluetree construction scheme [ZBC]. Also, the scheme in [HYBH] seems to be working properly only for single-hop networks, since otherwise links between nodes on the same level may not exist.

In the second protocol in [ZBC], several roots are initially selected. As in the first protocol each of them then creates its own scatternet. In the second phase, subtree scatternets are connected into one scatternet spanning the entire network. Each node is active in up to three piconets.

Bhatnagar and Kesidis [BK] proposed to run first a leader election process to first find a root for the whole network, and then to use the proximity information of this election protocol to create a tree scatternet by merging subtrees.

Pamuk and Karasan [PK] described a Bluetooth scatternet formation scheme which creates a tree, with nodes selected as masters using a mesure based on device characteristics (that is a measure combining battery type, battery level, and traffic generation rate). Nodes that have better device characteristics are preferred in master-slave decisions and therefore end at levels closer to the root (which is the node with the best device characteristics). In several existing tree based approaches, master nodes take the lead in selecting the slaves. This has contributed to a counterexample for connectivity of the final scatternet (see Figure 2). In [PK], however, each slave node takes the lead and selects one other node as its master. Each node, however, accepts up to 6 requests from potential slaves. During the first phase of neighbour discovery process, the selected master is the discovered neighbour that accepted the slave and has best device characteristics among such neighbours (once accepted, a master can be replaced in the process if a better one is discovered later). This phase is run for a certain predefined time. In the next phase, only roots of created trees participate in completing the scatternet. In case of a single-hop network, roots enter the second phase which runs in the same way as in the first phase (with fewer nodes, since only the roots will participate). This phase runs until a single root remains. In case of multi-hop networks, each root needs to label all its descendants slaves in the corresponding tree. The labels are then exchanged with all neighbours discovered in the first phase, in order to identify all bridges (nodes with endpoints having different labels, that is, roots). To connect the two trees, master-slave relations on the path from the node with label of lower priority (which also becomes the slave of other node) toward its root are reversed. Scatternet is then connected, since each node may receive at most one additional slave role in the second phase, keeping the maximum to 7. We observe that the initial maximum can be lower than 6, allowing the addition of more links and the creation of a mesh rather than a tree, by using extra slave connections.

Guerin et al. [GSV] proposed depth first search (DFS), breath first search (BFS), and MST-based scatternet formation schemes for unit disk graphs in two and three dimensions. They construct a tree where all nodes at each level (the tree they construct is seen as a bipartite graphs) are either masters or slaves. Their construction does not guarantee the 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 not localized.

The communication overhead in growing tree based BSF algorithms [ZBC, RKSA, PK, GSV, BK, PRT, DW, HYBH] is significant, especially when the appropriate maintenance procedures are designed and added to the protocol.

3.3. Context, on-demand and QoS based distributed BFS

3.3.1. Context based BFS

Siegemund [Si] discusses context based scatternet formation for sensor networks. Sensor information is included in the formation process, and nodes with the same context are included in the same piconet. For instance, nodes with the same temperature and noise are assumed to be in the same context and are arranged in the same piconet to enhance their mutual communication.

Gonzalez-Valenzuela, Vuong, and Leung [VVL] proposed BlueScouts, a on-demand scatternet formation protocol based on mobile agents. Their protocol runs in a fully asynchronous fashion, with device discovery being decoupled from actual topology formation. Agents are spread through the existing links in a controlled fashion and recursively signal back the state of the last computation’s outcome, leading up to the further replication of the mobile process or its termination. They conduct a coordinated spatial depth-first search over a logical backbone (excluding leaf nodes) in an attempt to reconfigure the role of a new device. The proposed ‘programmable’ approach introduces unmatched flexibility by allowing context-aware topology formation.

A scatternet queuing model was developed in [KSG, MM] and used to compare the delay and throughput characteristics of various topologies. It was found that the best topology is application dependent.

3.3.2. On-demand scatternet formation

Liu, Lee and Saadawi [LLS] proposed to combine the scatternet formation with on-demand routing, thus eliminating unnecessary link and route maintenances. Conventional Bluetooth broadcast consist of discovering all neighbors and then paging them one by one, and sending to each node a route request. Neighbors first enter the piconet, then, after the piconet is created, a route request packet is released to all neighbors. The neighbors, can continue spreading the destination search. The authors proposed two variants. In one variant, route request is released only after the piconet is fully created. In the other variant, route request is released to each neighbor immediately after paging it, without waiting for piconet to be fully created. In the same paper, the authors [LLS] also introduced an extended connectionless broadcast scheme where each master node and its slaves use the same channel for communication. The master node send inquiry messages on the channel while neighbors scan the same channel periodically to catch possible inquiry message. It achieves significantly shortened route discovery delay. The authors [LLS] also proposed to synchronize the piconets along each scatternet route to remove piconet switch overhead and obtain even a better channel utilization. They also presented a route-based scatternet scheduling scheme to enable a fair and an efficient packet transmissions over the scatternet routes. The proposed method in [LLS] provides high network utilization and extremely stable throughput, being especially useful in the transmission of large batches of packets and real time data in wireless environment. Zhen, Park, and Kim [ZPK] proposed an approach called ‘blue-star’, similar to the one in Liu, Lee and Saadawi [LLS] however it lowers the formation delay compared to [LLS], although it needs to perform several consecutive inquiry operations. Other variants of protocols are proposed in [CC1, GKWQ], where nodes concurrently form the scatternet (based on a flooding scheme) and route the data traffic.

Kawamoto, Wong and Leung [KWL] proposed a two-phase scatternet formation protocol to support dynamic topology changes while maintaining a high aggregate throughput. In the first phase, a control scatternet is constructed, which is not degree limited (more precisely, the number of bridge nodes is limited to 6, while pure slave nodes are parked) to support topology changes and route determination. The second phase creates a separate on-demand scatternet whenever a node wants to initiate data communication with another node (a similar approach to the one found in [LLS]). The on-demand scatternet is torn down when the data transmissions are finished. Since all the time slots are dedicated to a single communication session, a high aggregate throughput is achieved at the expense of a slightly higher connection setup delay.

3.3.3. QoS based scatternet formation

Augel and Knoor [AK] presented a survey of available BSF solutions and then proposed to consider QoS criteria for constructing scatternets. They argue that allowing nodes with larger degrees reduces the diameter but has a bad influence on throughput since the piconet capacity has to be shared among more devices. This is correct only if each node in a piconet receives equal amount of time for sending the data. However, some scheduling schemes are based on the actual traffic amount, therefore the time allocated to each slave does not need to be same. Augel and Knorr [AK] proposed that nodes with high degree stop paging and instructs a neighbour with a low degree to start paging instead. Each device may try to influence the topology depending on the QoS requirements. In their article they describe a BSF design guidelines for QoS applications, and they did not present any particular BSF protocol. Threshold based schemes (advocated in [AK]) may fail to construct the scatternet and/or provide the QoS although both may be possible by alternative schemes. One bad link or a bad node on a route does not necessarely fails the QoS criteria on a longer route.

Pabuwal, Jain and Jain [PJJ] proposed to switch between several different BSF algorithms depending on the application requirements. However, no information about the possible criteria on which this switching can be based is given, which is needed to have a flexible control of the topology. Instead of switching between various formations algorithms, it might be better to have one algorithm which controls the topology in an application-oriented manner using some specific related parameters.

Melodia and Cuomo [MC1, MC2, CMA] provided an integrated approach to address scatternet formation, scatternet maintenance, and the quality of service support for small and moderate size personal area networks. They first proposed a self-healing algorithm producing multi-hop scatternets (called SHAPER) which produces tree-shaped scatternets. The initially created tree is converted into a logical tree. The initial physical links in the tree are maintained as much as possible by inserting other links for each network change. When a logical link cannot be supported by a chain of physical links, the overall scatternet is reconfigured. Such reconfiguration guarantees the connectivity of the tree. A procedure that produces a meshed topology (based on a combinatorial optimization formulation with initial centralized solution) by applying a distributed scatternet optimization algorithm (DSOA) on the network built by SHAPER. The main issue in the proposed protocols [MC1, MC2, CMA] is their lack of scalability. Therefore further research is needed for the challenging problem of providing QoS in Bluetooth based multi-hop networks.

3.4. Clustering based BSF

Basagni and Petrioli [BP], [PBC] described a multihop scatternet formation (called BlueStar) scheme based on a clustering scheme [LG], taking into account several Bluetooth issues which do not pertain to clustering. The Clusterhead (master role) decisions are based on node weights (instead of node IDs, as used in [LG]), that express their suitability to become masters. It follows a variant of the clustering method described in [B]. All clusterhead nodes are declared master nodes in a piconet, with all nodes belonging to their clusters as their slaves. In order to assure connectivity, some of the slaves become masters of additional piconets (following, e.g., [AWF]),. However, piconets 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. The topology discovery phase is performed before clustering in order to provide each node with the information about all its neighbors. Each device executes the device discovery protocol for about eight seconds. Then, if the visibility (e.g. unit disk) graph is connected, the resulting network will be connected with high probability. Device discovery is performed according to the procedure in [SBTL] which we outlined above. A performance evaluation of the clustering-based scatternet formation scheme [BP], [PBC] is given in [BBP3].

Basagni et al. [BBP5] described the results of an ns2-based comparative performance evaluation among three major solutions for forming multihop scatternet: [LSW], [PBC], [ZBC]. 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 [PBC] 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.

Ferraguto, Mambrini, Panconesi and Petrioli [FMPP] proposed Blue Pleiades for device discovery and BSF in multi-hop networks. As soon as a node has discovered *c* neighbors, it proceeds to the next phases of the piconet formation and interconnection. Nodes with less than *c* neighbors and upon timeout expiration will exit the device discovery [FMPP]. Their extensive simulations show that *c= 6, 7* are excellent choices that guarantee the connectivity of the topology with high probability. The authors [FMPP] combine their new device discovery protocol with the BlueStar protocol [BP, PBC] to obtain a simple, fast and effective scatternet formation protocol that overcomes the degree limitation problem of BlueStar. This protocol is termed Blue Pleiades [FMPP]. In short, clustering based BSF [BP, PBC] is applied on the topology created after the degree limited neighbor discovery [FMPP].

A greedy centralized multihop algorithm where a hypothetical central entity knows the complete topology has been proposed in [BKNR]. Distributed algorithms have also been proposed in [BKNR], which assume a 2-hop neighborhood information. This is achievable in Bluetooth since the identities of the neighboring nodes are known at the end of the device discovery procedure. The nodes are made to exchange this neighborhood information with each of its neighbors so that they have a 2-hop information and a partial view of the underlying topology. The algorithm [BKNR] applies a variant of a clustering algorithm where the number of nodes in each cluster is limited to seven, in accordance to Bluetooth restriction. A node with a highest degree among all its undecided neighbors will become a master node and will choose up to seven slaves among neighboring nodes, with the priority given to lower degree nodes. However, there are examples where the scatternet is disconnected. This may occur when two clusterheads were originally connected, but they formed clusters and "erased" their link without leaving alternate connection between their piconets. For example, as illustrated in Fig. 2, assume that the graph contains two connected nodes *A* and *B,* each with its own seven more neighbors. Thus, *A* and *B* have degrees eight, and will become masters of two piconets, containing their own seven neighbors as slaves. However, the graph will then be disconnected since the link between *A* and *B* is not part of a scatternet.

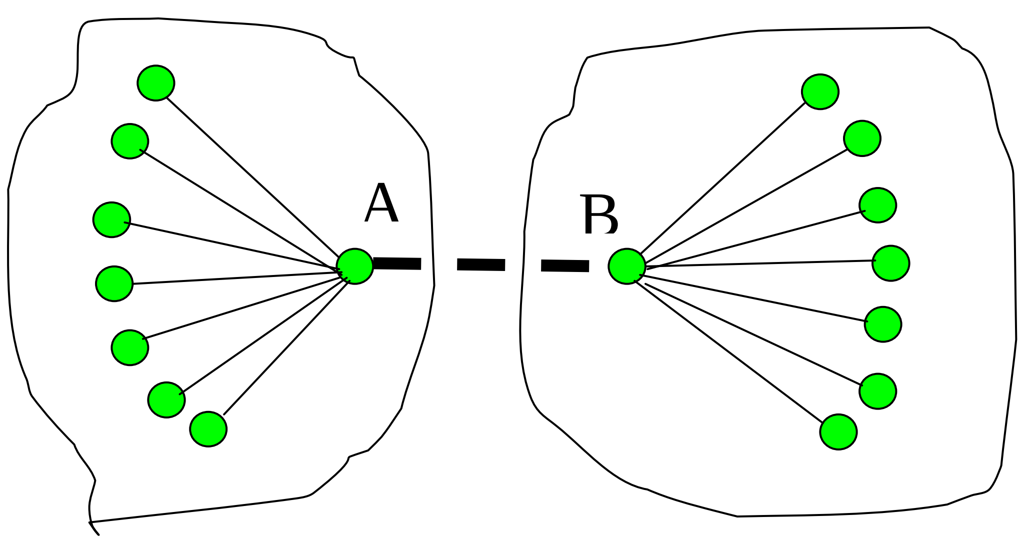

Figure 2. Creation of disconnected scatternets in several algorithms

Wang, Thomas and Haas [WTH], and Guerin, Kim, and Sarkar [GKS] essentially proposed variants of clustering-based scatternet formation schemes, where the clustering process does not use any ID to decide clusterheads, that is, master nodes. Instead, the decisions are made at random. Already existing master nodes have the priority in attracting more slaves, up to the limit. Initial connections are made by nodes entering a scan or an inquiry scan phases at random. Whenever a node is assigned a master or a slave role, or if it was unable to join any piconet or to attract any neighbor as its slave in order to create its own piconet, then bridge piconets are added to connect the scatternet. However, the process does not always lead to a connected structure. The counterexample is the same that applies to [BKNR] (Fig. 2). On a positive side, [WTH] proposes two excellent measures for the performance of scatternets: average shortest-path length and maximum traffic flow.

3.5. Connected out-degree limited scatternets

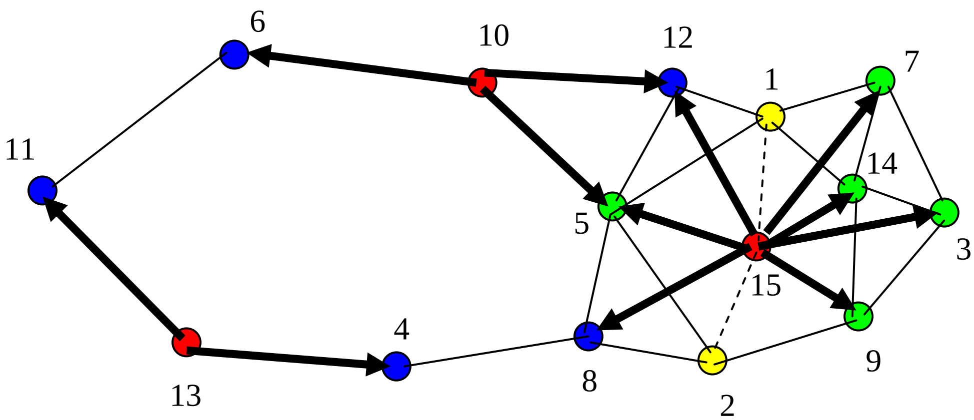

Figure 3. The first iteration of BlueMesh algorithm

Petrioli and Basagni [PB, PBC2] described a scheme (called BlueMesh) that guarantees connectivity and limits the number of slaves in each piconet. Their neat scheme does not require position information, but instead the local information is extended to a two-hop information, with a two round device discovery phase for obtaining necessary information. It is a modified clustering process, where selection of slaves is performed in such a way that, if a master has more than seven neighbors, it chooses up to seven slaves among them so that it can reach all the others via the chosen ones. Such coverage is always possible with up to five slaves [ZBC]. The scatternet formation proceeds in iterations. Nodes that participate in a given iteration perform the modified clustering process. Initially all nodes are undecided. In each iteration, init-nodes (nodes having the largest weight among their immediate undecided neighbors) create piconets, by choosing at most seven neighbors as slaves, and deleting the remaining edges. The iteration stops when all nodes are decided. All created masters, together with the slaves that are not selected for links with slaves from other piconets, withdraw from the next iteration. Simulations by the authors show that the created scatternets have a low average number of roles per node (about two), with an average path length increase between 20% and 80%. The number of iterations grows slowly with the number of nodes (it is about 4.5 for networks with 200 nodes). The method may show weaknesses on some other metrics, especially about the worst-case number of slave roles a node can assume. For instance, in case of dense networks (e.g., complete graph), the second largest node in a neighborhood may end up serving as slave to all the masters in the same neighborhood. Nevertheless, among all methods that do not use position information, the method [PB, PBC2] appears to be currently the best available method for multi-hop networks. An attempt to improve it further is given in the next subsection (and in [S2]).

The BlueMesh algorithm [PB, PBC2] is illustrated in Figures 3 and 4. In the first iteration, clustering scheme will select node 15 as master node, with its seven neighbors as slaves (all but nodes 1 and 2), following as a selection criteria, andwhenever needed, the largest ID,. Node 13 also creates a piconet, having the largest ID among its neighbors. Finally, node 10 also creates a piconet, after its neighbor 12 announces to be 'defeated' by node 15 in the process. Master nodes 15 and 10 select node 12 as gateway, master nodes 15 and 13 select nodes 4 and 8 to connect them, while piconets mastered by 10 and 13 select nodes 11 and 6 for connection. The 'red' nodes (10, 13, 15) are masters of the created piconets, while the 'green' nodes (3, 5, 7, 9, 14) are the slaves that are not needed to connect the piconets in Fig. 3. Therefore, nodes 1, 2, 4, 6, 11, and 12 (yellow and blue nodes in Fig. 1) are the ones selected for the second iteration.

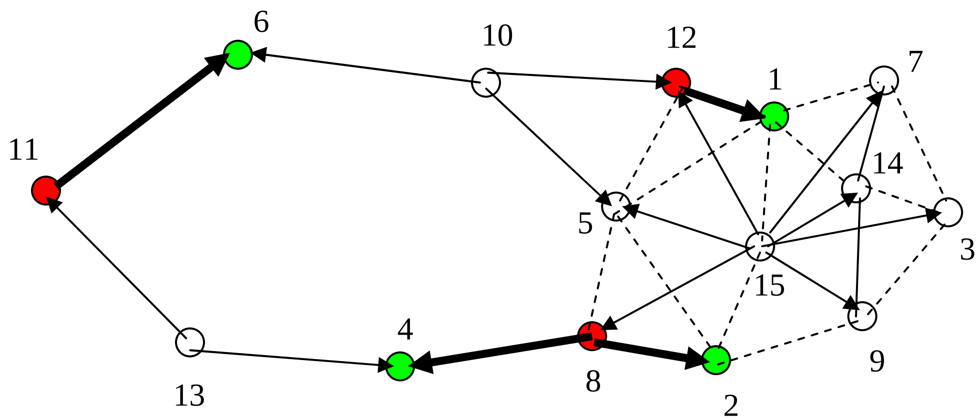

Figure 4. The second iteration of BlueMesh algorithm

In the second iteration (see Figure 4), three more piconets, mastered by 'red' nodes 8, 11, 12, are created, to connect the overall structure.

3.6. Maximal independent set based BSF

An attempt to simplify the BlueMesh procedure was made in [S2]. It essentially interprets the slave selection as the *maximal independent set* problem, and reduces the process to two iterations. In the first iteration, every nodes creates a piconet with itself as a master node. In the second iterations, following a clustering based approach, each node estimates whether or not its piconet is needed for the overall connectivity. If not, it deletes its piconet.

The maximal independent set *MIS(X)* of a set of nodes *X* is a set of nodes *Y* from *X* such that no two nodes from *Y* are connected ('independent set'), and *Y* is not a proper subset of another set with the same property ('maximal'). In the first iteration of MIS based scheme [S2], each node selects the *MIS* of its neighbouring nodes as the set of its slaves. Each node *A* has *key(A)* which can be defined in a variety of ways, known to all its neighbouring nodes. Let *X* be a set of neighbours of a given node *S*. To find *MIS(S)*, the node *S* chooses a node *A* from *X* with maximal *key(A)*. Note that here the algorithm can use the minimal *key(A)* instead, which has impact on the performance of the second iteration. Node *A* is declared a slave of *S*, and is eliminated from *X*, together with all its neighbours. This is repeated until *X* becomes empty. If the number of selected slaves is less than seven then, as in BlueMesh, additional slaves can be selected at random up to the limit.

In the second iteration, the network is not the original unit graph, but the scatternet structure where each node has its own piconet. A clustering based confirmation/elimination scheme is performed. The decisions are made by masters in *S* that have a higher ID than any of the nodes in any of neighboring piconets (this includes the slaves of *S* and the masters of the piconets where *S* is a slave). Such node *S* verifies whether or not the piconet structure would remain locally connected if its piconet is to be destroyed. If it still connected, its piconet is not needed. The decision can be communicated to all piconets where S is participating, which enables other nodes to make their own decision.

For example, in Fig. 3, the first iteration will create all piconets with all the indicated edges except the two dashed ones. In Fig. 5, nodes 15 and 13 first decide to keep their piconets. Node 14 then decides to keep its piconet because of node 1. Node 12 preserves its piconet because of node 10. Node 11 preserves its piconet to connect node 6. Node 10 also preserves the piconet to connect to piconet 11 via 6. Node 9 keeps its piconet because of node 2, while node 8 preserves its piconet in order to remain linked to piconet 13 via node 4. The remaining nodes (drawn in green) do not need to preserve their piconets.

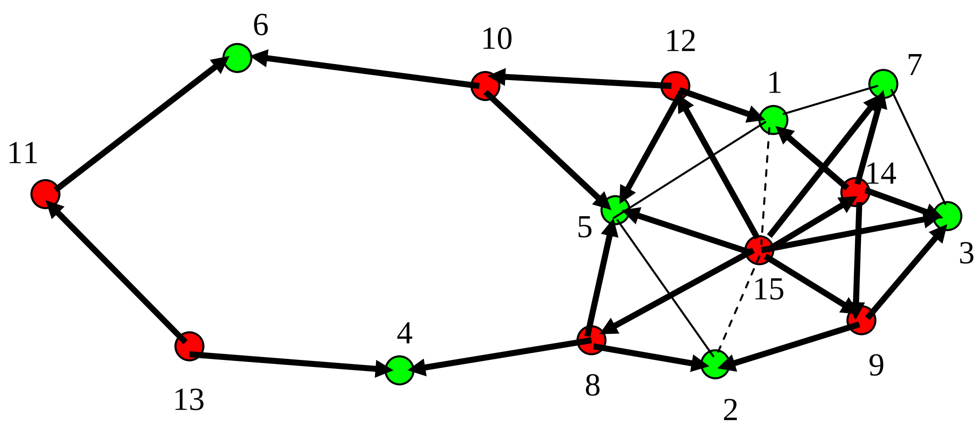

Figure 5. The second iteration of MIS based BSF

While the number of slaves of each master is limited, and the scatternet is connected, the number of slaves for each node is not limited, and in some cases, e.g. complete graph, one node can be selected as slave to all other nodes. This is the same problem shared with the BlueMesh. The only advantage of the new algorithm is to reduce the number of iterations to two and therefore obtain a faster BSF.

3.7. Position based connected and degree limited BSF

Stojmenovic [S] and Li, Stojmenovic and Wang [LSW] proposed (and made available in June 2001) the first BSF schemes that construct degree limited and connected piconets in multihop networks, without parking any node. Moreover, the methods presented in [S, LSW], and described in this section, also provide the limit on the number of slave roles for each node, and (if desirable) planarity which is important for the performance of the routing scheme that guarantees delivery [BMSU]. The BlueMesh scheme [PB, PBC2] achieved the same main objectives (connectivity and limitation of the piconet size), but did limit the number of slave roles and did not construct a planar graph. However, [S, LSW] achieved their objectives by using a stronger assumption, position information, and some geometrical structures.

Position based BSF schemes [S, LSW] require that nodes first discover their *all* neighboring devices before the scatternet is established. This phase (learning the underlying unit disk graph) can be implemented by using either a Bluetooth based discovery [SBTL] or using a single channel medium access such as IEEE 802.11 (another name for WiFi), as discussed above. The protocols then continue by *limiting the degree* of each node, while preserving the connectivity, and by *assigning master and slave* roles to each node. There are several approaches to accomplish this. The basic differences are in the order of these two tasks, in the order of making decisions within each task, and in the way master-slave roles are assigned. Deciding master-slave relations can be done *during* the degree limitation process, or *after* the degree limitation process is finished and as a separate phase. The degree limitation (and/or master-slave decisions) can be done *simultaneously* or *iteratively*. In the *simultaneous approach*, all nodes, independently and at the same time, decide about the links to preserve or the master-slave roles, using a scheme that will assure that the final choices are symmetric (commonly selected links are preserved; master-slave roles decisions are in agreement). Alternatively, nodes can make the degree limitation or the master-slave decisions at different times, and decisions already made by neighbors have impact on decisions to be made by a given node (*iterative approach*).

One approach is, for instance, *after-simultaneous-iterative,* where nodes make degree limitations simultaneously (and, in some cases e.g. when *LMST* or *Yao* structure is applied, gather decisions from neighbors to decide which link remain in the final structure), followed by an iterative procedure for deciding master and slave relations (e.g. clustering based procedure such as BlueStar [BP, PBC]). This variant of the originally proposed procedure [LSW] was proposed by Basagni, Bruno and Petrioli [BBP5]. If the degree limitation step requires a message exchange then it may be faster if the master-slave roles are also assigned at the same time. Such *during-simultaneous-iterative* and *during-iterative-iterative* procedures are the ones proposed in [LSW] and they are also based on clustering. The same article [LSW] also elaborates on *during-iterative-iterative* approach. If the clusterhead decisions are based on node degrees (number of neighbors in the original unit graph) as the primary key in comparing (that is, nodes with more neighbors have more chance to become master nodes) then another round of information exchange (similar to the device discovery round) is needed following the first neighbor discovery. In [S], the *during-simultaneous-simultaneous* and *during-iterative-simultaneous* approaches are described. The major difference is in the master-slave decision process. It was proposed to use *(dominating set membership, node degree, node identifier)* as the key for comparing two nodes, and assign master role to higher key node on any link. This means that nodes that belong to a dominating set (see [SW] for the definition and survey of existing schemes) have priority in being a master node on a link. If this primary key is the same for both nodes, or the key is not used at all, then the node degree can be used. If the node degrees are the same, or not used at all in comparison, then node identifiers can be used for making the role assignment. Note that *after-simultaneous-simultaneous, after-iterative-simultaneous,* and *after-iterative-iterative* BSF protocols can also be considered.

3.7.1. Geometric structures for degree limitation

We will now describe several geometric structures that can be used to achieve the degree limitation in scatternets. The basic solution is to apply a minimum spanning tree (*MST*) or a structure that contains it. The *MST* is a subgraph of a given unit disk graph which contains all the nodes, is connected and whose sum of edge lengths is minimized. The average number of neighbours (the average degree) of each node of a *MST* is ≈1.99, while the maximum number is 6. However, *MST* is not a localized structure, since a its computation requires a global network knowledge at each node. We therefore need to use other structures.

A localized *MST* (*LMST*) based topology control algorithm was proposed by Li, Hou and Sha in [LHS]. Each node *u* first collects positions of its one-hop neighbours *N1(u).* The node *u* then computes the minimum spanning tree *MST(N1(u))* of *N1(u).* The node *u* keeps a directed edge *uv* in *LMST* if and only if the edge *uv* is also an edge in *MST(N1(v)).* If each node already has a 2-hop neighbouring information, then the construction does not involve any message exchange between neighbouring nodes. Otherwise each node contacts the neighbours along its *LMST* link candidates, in order to verify the status at the other nodes. The average number of neighbours (average degree) of nodes is ≈2.04, while the maximum is still limited by 6. *MST* and *LMST* are both planar graphs (a graph is planar is no two edges of it intersect except possibly at common endpoints).

Relative neighbourhood graph (*RNG*) is introduced by Toussaint [T], and can be defined, in the simplest form, as follows. An edge *uv* is included in *RNG* if and only if it is not the longest edge in any triangle *uvw*. Fig. 6 shows an example of an RNG of a UDG (unit disk graph).

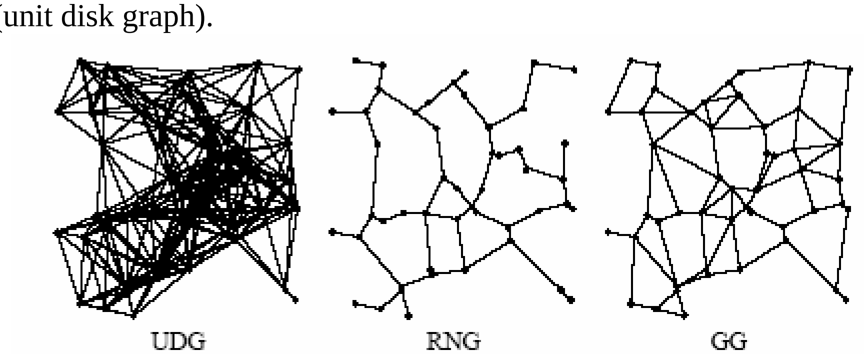

Figure 6. Unit disk graph (UDG), RNG and GG of a set of nodes

Using this definition, some edges may have very large degrees in several particular scenarios. To obtain a degree limited structure, the record *w(AB)=(|AB|, min(id(A),id(B)), max(id(A),id(B)))* can be used instead for edge comparisons, since no two edges have the same record. We refer to this structure in the sequel, assuming a random node placement and a very low chance of any two edges being of the same length. The degree of each node in *LMST* and *RNG* (*LMST* is a subset of *RNG*) is limited to 6 (for nodes located in a plane). The average degree of a node in *RNG* is ≈2.4. Note that the construction of *LMST* and *RNG* does not require that the exact positions of nodes and their neighbours to be known; in fact only the corresponding mutual distances are required. In both cases, each node requires the knowledge of its distances to the neighbours, and the distances between any pair of neighbours.

Gabriel graph (*GG*) is proposed in [GS], and is defined as follows. *GG* contains an edge *uv* if and only if the disk with diameter *uv* contains no other node inside it. This criterion can be tested in two ways. For an edge *uv* to be included in *GG*, each common neighbour *w* of nodes *u* and *v* should be located at a distance of at least $|uv|/2$ from the midpoint of *uv*. Alternatively, one can verify the angles from neighbors to *uv*. If for a common neighbor *w* of *u* and *v*, $\angle uwv > \pi/2$ then *uv* is not in *GG*. It should be observed, as in the case of *LMST* and *RNG,* that the construction of *GG* requires only the knowledge of the location of a node and those of its neighbours. Fig. 6 shows an example of a *GG*. The average degree of a node of GG is ≈3.8. However, there is no worst case limit for the node degree. Therefore, to assure degree limitation, another geometric structure needs to be applied, at least at nodes whose degree exceeds the Bluetooth limit of seven. Note that each node can decide, for each of its links, whether or not it belongs to *RNG* or *GG* without communicating with its neighbours. To construct the *LMST,* it needs to exchange the decisions made with the neighbours, since only links selected by both endpoints are in the final structure.

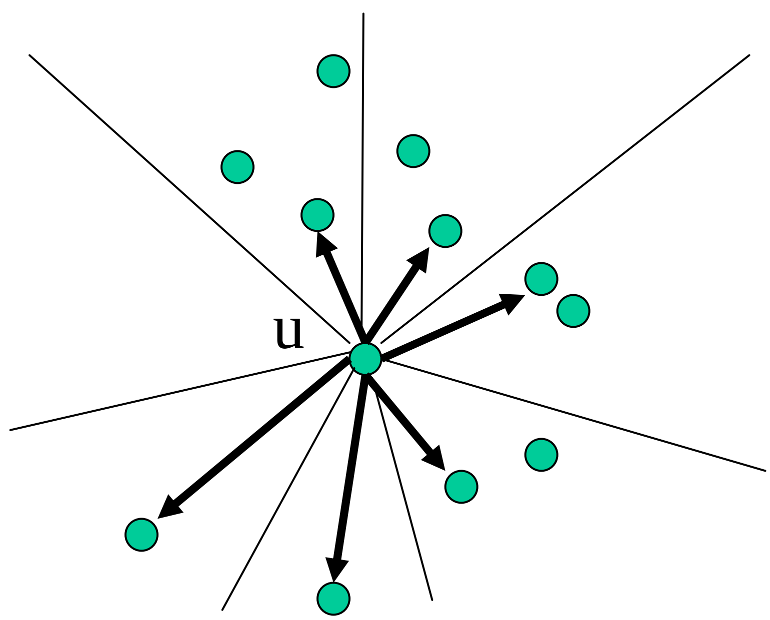

Figure 7. Yao graph degree limitation for *p=7*

The Yao_p graph [Y] is proposed by Yao to construct efficiently *MST* in high dimensions. Any p equally separated rays originated at each node u define p cones. In each cone, u then chooses the closest node v within the transmission range, if there is any, and then selects a directed link uv (see Fig. 7 for an illustration). Links which are not selected by u are deleted. Since Yao_p contains an *MST* as a subgraph (for $p \geq 6$), and that is after deleting all links which are not selected by its both endpoints, the network connectivity is still preserved. Note that Yao_p is not necessarily planar.

3.7.2. Degree limited structures for BSF and routing in scatternets

There are a number of options to obtain degree limited scatternets. One option is to apply Yao_7 to UDG constructed after the device discovery phase (note that, in this option, it is not necessary to complete this phase; it suffices to merely achieve the overall connectivity). The drawback of this option is that the obtained scatternet is not planar. The planar structure may be desirable in order to provide the routing with a guaranteed delivery, since the best existing protocol that achieves that [BMSU] requires planar graph in the recovery mode.

The next option is to use *LMST* or *RNG,* which are planar and guaranteed to be degree limited (therefore it is not necessary then to apply the Yao graph construct). The drawback of using them is that these structures are quite sparse, and therefore the greedy routing (forwarding message to a neighbor that is closest to the destination), will frequently fail, leading to long routes with a protocol that guarantees delivery [BMSU].

Further option is to apply *GG*, followed by Yao_7, applied only on nodes whose degree exceeds 7. The number of such nodes is small, if any, but they may exist. This structure is planar, localized, and the densest known that is defined with so little local knowledge and zero messages (besides device discovery to learn neighbors). Note that [LSW] proposed a partial Delaunay triangulation (*PDT*) as an alternative locally defined structure which is denser, however subsequent measures show that *PDT* is only about 1% denser than *GG,* thus we are not covering it in this chapter.

LMST, RNG and *GG* have average degrees ≈2.04, ≈2.5 and ≈3.8, respectively. They all (and *Yao* structure as well) tend to select short edges for the scatternet structure. Stojmenovic [S1] (a note about it is also made in [LSW]) observed that, to improve the routing performance of a scatternet, some additional edges may be carefully selected. The selection depends on the criterion being applied in measuring routing performance. If the criterion is to minimize the hop count then one can add several ‘long’ edges to the scatternet. It is desirable to spread the added edges in several directions, in order to complement the existing short edges. This can be done by applying the same angular range division used in Yao construct and selecting the long edges in sectors where no short edge exists among the edges of *LMST, RNG* or *GG*. This assures a balanced edge structure in all directions. Addition of randomly selected long edges can also be considered (especially if distances rather than position information were used to define the structure). If power consumption was used as criterion, the additional edges should have a length close to the ideal one, following the discussion made in [SL2].

Conclusions

There is a number of Bluetooth scatternet formation protocols already proposed in the literature. It can be observed that very few of them satisfy most of the desirable characteristics. There are relatively few actual implementations and comparisons. For example, [BBP5, BBMP] compared BlueTree [ZBC], BlueStar [PBC], BlueNet [WTH] and the position based approach [LSW]. They concluded that device discovery is the most time consuming operation. Their final conclusion is that forming scatternets is still a formidable task, because of the device discovery and the extra complexity imposed by the Bluetooth technology on the implementation of the distributed algorithms. Similar conclusions were made in [RVGS]. The reader can find alternative surveys on BSF in [BBP1, PMS, WHC].

We anticipate that more Bluetooth scatternet formation schemes will be developed in the near future, and that some modifications to Bluetooth specifications could be made to find solutions which satisfy a number of desirable properties and make it suitable for commercial applications in the multi-hop scenarios. An interesting and a major open problem in the area is to design of a BSF algorithms that will guarantee connectivity and degree limitation (for both of the master and the slave roles) and without using the position information.

References

[ACNCG] M. Ajmone-Marsan, C.F. Chiasserini, A. Nucci, G. Carello, and L. de Giovanni, Optimizing the Topology of Bluetooth Wireless Personal Area Networks, Proc. INFOCOM, 2002.

[AK] M. Augel, R. Knoor, Bluetooth scatternet formation, State of the art and a new approach, International Conference on Architecture of Computing Systems, Augsburg, Germany, March 23-26, 2004, ARCS 2004, LNCS 2981, 2004, 260-272.

[AWF] K.M. Alzoubi, P.-J. Wan, and O. Frieder, "Message-optimal connected-dominating-set construction for routing in mobile ad hoc networks," Proc. Third ACM Int'l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc'02), 2002.

[BBFMSK] S. Baatz, S. Bieschke, M. Frank, P. Martini, C. Scholz and C. Kuhl, Building efficient Bluetooth scatternet topologies from 1-Factors, *Proc. IASTED Wireless and Optical Communications*, Banff, Canada, July 2002.

[B] S. Basagni, Distributed clustering for ad hoc networks, Proc. Int'l Symp. Parallel Algorithms, Architectures and Networks ISPAN, pp. 310-315, June 1999.

[BBMP] S. Basagni, R. Bruno, G. Mambrini, C. Petrioli, Comparative performance evaluation of scatternet formation protocols for networks of Bluetooth devices, Wireless Networks 10, 197-213, 2004.

[BBP1] S. Basagni, R. Bruno, C. Petrioli, Scatternet formation in Bluetooth networks, in: Mobile Ad Hoc Networking (S. Basagni, M. Conti, S. Giordano, I. Stojmenovic, eds.), Wiley, 2004, 117-137.

[BBP2] S. Basagni, R. Bruno and C. Petrioli, Device discovery in Bluetooth networks: A scatternet perspective, *Proc. IFIP-TC6 Networking Conf., Networking 2002*, Italy, May 2002, LNCS 2345, 1087-1092.

[BBP3] S. Basagni, R. Bruno, and C. Petrioli, Performance evaluation of a new scatternet formation protocol for multi-hop Bluetooth networks, Proc. IEEE Wireless Personal Multimedia Comm. WPMC, Oct. 2002, 208-212.

[BBP5] S. Basagni, R. Bruno and C. Petrioli, A performance comparison of scatternet formation protocols for networks of Bluetooth devices, *Proc. IEEE PerCom*, Texas, Mar. 2003, 341-350.

[BP] S. Basagni and C. Petrioli, A scatternet formation protocol for ad hoc networks of Bluetooth devices, Proc. IEEE Vehicular Technology Conf., May 2002.

[BFNO] L. Barriere, P Fraigniaud, L. Narajanan and J. Opatrny, Dynamic construction of Bluetooth scatternets of fixed degree and low diameter, *Proc. ACM SODA*, 2003.

[BKNR] K. Balaji, S. Kapoor, A.A. Nanavati, and L. Ramachandran, Scatternet formation algorithms in the Bluetooth network, Manuscript, 2001.

[BK] V. Bhatnagar and G. Kesidis, Bluetooth scatternet formation using proximity information of an election protocol, Joint 2nd IEEE Int. Conf. on Networking and IEEe Int. Conf. Wireless LANs and Home Networks, Atlanta, Aug. 2002.

[BMSU] P. Bose, P. Morin, I. Stojmenovic and J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, *ACM/Kluwer Wireless Networks*, 7, 6, November 2001, 609-616.

[BR] P. Bhagwat, S.P. Rao, On the characterization of Bluetooth scatternet topologies, www.winlab.rutgers.edu/~pravin .

[BS] P. Bhagwat and A. Segall, A routing vector (RVM) for routing in Bluetooth scatternets, IEEE Int'l Work. Mobile Multimedia Comm. MoMuC, Nov. 1999, 375-379.

[CABG] C.F. Chiasserini, M. Ajmone-Marsan, E. Baralis, P. Garza, Towards feasible distributed topology formation algorithms for Bluetooth-based WPANs, Hawaii Int. Conf. System sciences HICSS-36, Big Islands, Hawaii, Jan. 2003, 313-322.

[CC-] M.-T. Chou and R.-S. Chang, Blueline: A distributed Bluetooth scatternet formation and routing algorithm, Int. Conf. Parallel and Distributed Computing and Networks PDCN, Innsbruck, Austria, Feb. 2004.

[CC1] C.S. Choi and H.W. Choi, DSR based Bluetooth scatternet, ITC-CSCC, 2002.

[CMA] F. Cuomo, T. Melodia, I.F. Akyildiz, Distributed self-healing and variable topology optimization algorithms for QoS provisioning in scatternets, IEEE J. Selected Areas in Communications, 22, 7, Sept. 2004, 1220-1236; IEEE GLOBECOM Nov. 2003, 236-240.

[D] A. S. Daptardar, Meshes and cubes: Distributed scatternet formations for Bluetooth personal area networks, Master of Science, Washington State Univ., School of EECS, 2004.

[DW] Y. Dong and J. Wu, Three Bluetree formations for constructing efficient scatternets in Bluetooth, *Proc. of the 7th Joint Conference on Information Sciences*, Sept. 2003, 385-388.

[ETJ-] P. Engelstad, Do van Thanh, T. E. Jønvik: Formation of scatternets with heterogenous Bluetooth devices, Int. Conf. on Third Generation Wireless and Beyond (3Gwireless2003), San Francisco, USA, May 27-30, 2003.

[FC] C.C. Foo, K.C. Chua, Bluerings – Bluetooth scatternets with ring structures, IASTED Wireless and optical Communications, 2002.

[FMPP] F. Ferraguto, G. Mambrini, A. Panconesi, C. Petrioli, Blue Pleiades, a new solution for device discovery and scatternet formation in multihop Bluetooth networks, WINET, to appear.

[GKWQ] J. Ghosh, V. Kumar, X. Wang, C. Qiao, BTSpin - Single Phase Distributed Bluetooth Scatternet Formation, TR 2004-06, Dept. Comp. Sci. & Eng., Univ. of Buffalo, December 13, 2003.

[GKS] R. Guerin, E. Kim, and S. Sarkar, Bluetooth technology key challenges and initial research, Proc. SCS Comm. Networks and Distributed Systems Modeling and Simulation CNDS, pp. 157-163, 2002.

[GS] K.R. Gabriel and R.R. Sokal, A new statistical approach to geographic variation analysis, *Systematic Zoology*, Vol.18, 1969, pp. 259-278.

[GSV] R. Guerin, S. Sarkar, and E. Vergetis, Forming connected topologies in Bluetooth ad hoc networks." 2002.

[H] J.C. Haartsen, The Bluetooth radio system, IEEE Personal Comm., vol. 7, pp. 28-36, Feb. 2000.

[HB] J. Hightower and G. Borriello, Location systems for ubiquitous computing, *IEEE Computer*, Vol.34, No.8, 2001, pp. 57-66.

[HCSKS] L. Huang, H. Chen, T.V.L.N. Sivakumar, T. Kashima, K. Sezaki, Impact of Topology on Bluetooth, International Conference of Embedded and Ubiquiotus Computing (EUC2004) Aizu, Japan, Springer LNCS Aug, 2004; Int. Workshop on Wireless Ad-hoc Networks IWWAN, Oulu, Finland, June 2004.

[HW] L.E. Hodge, R.M. Whitaker, What are characteristics of optimal Bluetooth scatternets, Mobiquitous, Boston, Aug. 2004.

[HYBH] T.C. Huang, C.S. Yang, S.W. Bai and C.C. Huang, Hierarchical grown bluetrees – An effective topology for Bluetooth scatternets, Int. Symp. Par. & Distr. Proc. & Applications ISPA, AIZU-Wakamatsu, Japan, July 2003, LNCS.

[JH] Yuh-Jzer Joung, Geng-Dian Huang, A Simple and Fast Algorithm for Bluetooth Network Formation, manuscript, 2004.

[K-] M. Kazantzidis, Locally optimal Bluetooth Scatternet formation, UCLA Technical Report #01033. Los Angeles, CA. September 2001.

[KSG] R. Kapoor, M.Y.M. Sanadidi, M. Gerla, An analysis of Bluetooth scatternet topologies, IEEE ICC, 2003.

[KWL] Y. Kawamoto, V.W.S. Wong, V.C.M. Leung, A two-phase scatternet formation protocol for Bluetooth wireless personal area networks, IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, Louisiana, pp. 1453-1458, March 2003.

[LG] C.R. Lin and M. Gerla, Adaptive clustering for mobile wireless networks, IEEE J. Selected Areas in Comm., vol. 15, no. 7, pp. 1265-1275, 1997.

[LHS] N. Li, J. C. Hou, and L. Sha, Design and Analysis of an MST-Based Topology Control Algorithm, Proc. *INFOCOM 2003*, San Francisco, USA, 2003.

[LLS] Y. Liu, M.J. Lee, T.N. Saadawi, A Bluetooth scatternet route structure for multi-hop ad hoc networks, IEEE J. Selected Areas in Communications, 21, 2, Feb. 2003, 229-239.

[LMS] C. Law, A.K. Mehta and K.Y. Siu, A new Bluetooth scatternet formation protocol, *Proc. ACM MobiHoc01*, October 2001, 183-192; IEEE GLOBECOM Nov. 2001; Mobile Networks and Applications, 8, 485-498, 2003.

[LSW] X.-Y. Li, I. Stojmenovic, Y. Wang, Partial Delaunay triangulation and degree limited localized Bluetooth scatternet formation, Proc. AD-HOC NetwOrks and Wireless (ADHOC-NOW), Toronto, September 20-21, 2002, 17-32; IEEE Trans. Parallel and Distributed Systems, 15, 4, April 2004.

[LTC] T.Y. Lin, Y.C. Tseng, K.M. Chang, A new BlueRing scatternet topology for Bluetooth with its formation, routing, and maintenance protocols, HICSS Jan. 2003; Wireless Communications and Mobile Computing, 3, 2003, 517-537.

[MeZ] V. Mehta and M. El Zarki, A Fixed Sensor Networks for Civil Infrastructure Monitoring, First Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), Sardegna, Italy, Sep. 2002; Wireless Networks, July 2004, 401-412.

[MC1] T. Melodia, F. Cuomo, Ad hoc networking with Bluetooth: key metrics and distributed protocols for scatternet formation, Ad Hoc Networks, 2, 2, 2004, 109-202; IEEE GLOBECOM 2002, Vol. 1, 941-945; IEEE ICC June 2004.

[MC2] T. Melodia, F. Cuomo, Locally optimal scatternet topologies for Bluetooth ad hoc networks, WONS, Madona di Campiglio, Jan. 2004, LNCS 2928, 116-129.

[MM] V.B. Misic and J. Misic, Bluetooth scatternet with a master-slave bridge: A queuing theoretical analysis, Proc. IEEE GLOBECOM, Nov. 2002.

[MRTVJ] G. Miklós, A. Rácz, Z. Turányi, A. Valkó, and P. Johansson, Performance aspects of Bluetooth scatternet formation, 1st ACM Int. Symp. MOBIHOC, Boston, Massachusetts, USA, November 2000.

[MTZ] D. Miorandi, A. Trainito, A. Zanella, On efficient topologies for Bluetooth scatternets, 8th IFIP TC6 PWC, Sept. 2003, LNCS 2775, 726-740.

[MZ] D. Miorandi and A. Zanella, On the optimal topology of Bluetooth piconets: Roles swapping algorithms, *Proc. Mediterranean Conf. on Ad Hoc Networks*, Sardinia, Italy, Sept. 2002.

[PB] C. Petrioli and S. Basagni, Degree-constrained multihop scatternet formation for Bluetooth networks, *Proc. IEEE GLOBECOM 2002*, Taipei, Taiwan, November 2002.

[PBC2] C. Petrioli, S. Basagni, I. Chlamtac, BlueMesh : Degree-constrained multi-hop scatternet formation for Bluetooth networks, Mobile Networks and Applications, 9, 33-47, 2004.

[PBC] C. Petrioli, S. Basagni, and I. Chlamtac, Configuring BlueStars: Multihop scatternet formation for Bluetooth networks, IEEE Trans. Computers, 52, 6, 779-790, 2003.

[PC] B.J. Prabhu and A. Chockalingam, A routing protocol and energy efficient techniques in Bluetooth scatternets, *Proc. IEEE ICC* 2002.

[PJJ] N. Pabuwal, N. Jain, B.N. Jain, An architectural framework to deploy scatternet based applications over Bluetooth, IEEE Int. Conf. on Communications, 2003.

[PK] C. Pamuk and E. Karasan, SF-Devil: Distributed Bluetooth scatternet formation algorithm based on device and link characteristics, IEEE Int. Symp. Computers and Comm. ISCC, Turkey, July 2003, 646-651; Medhoc, Bodrum, Turkey, 2004.

[PMS] K. E. Persson, D. Manivannan and M. Singhal, Bluetooth scatternets: criteria, models and classification, Ad Hoc Networks, 2004, to appear.

[PRT] E. Pagani, G.P. Rossi, S. Tebaldi, An on-demand Bluetooth scatternet formation algorithm, WONS, Madona di Campiglio, Jan. 2004, LNCS 2928, 130-143.

[RKSA] L. Ramachandran, M. Kapoor, A. Sarkar, A. Aggarwal, Clustering algorithms for ad hoc wireless networks, ACM DIALM Workshop, 2000, 54-63.

[RVGS] J. Rank, E. Vergetis, R. Guerin, S. Sarkar, On the challenges of transforming Bluetooth into an ad hoc networking technology, IEEE MASS, Fort Lauderdale, USA, October 2004.

[SCH] T.V.L.N Sivakumar, H. Chen, L. Huang, Adaptive Network Formation Protocol for Bluetooth Scatternet, The First IEEE and IFIP Int. Conf. on Wireless and Optical Communications Networks (WOCN 2004), Muscat, Oman, June, 2004.

[Si] Siegemund, F.: Kontextbasierte Bluetooth-Scatternetz-Formierung in ubiquit¨aren Systemen. First German Workshop on Mobile Ad hoc Networks, 2002.

[SLWW] W.Z. Song, X.-Y. Li, Yu Wang and W. Z. Wang, dBBlue: Low diameter and self-routing Bluetooth scatternet, J. Parallel & Distributed Computing, to appear.

[SA] H. Sreenivas and H. Ali, An Evolutionary Bluetooth Scatternet Formation Protocol, Proceedings of the 37th Hawaii Int. Conf.on System Sciences, Jan. 2004.

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

[S1] I. Stojmenovic, Routing in Bluetooth with geometric structures enhanced with long links, in preparation.

[S2] I. Stojmenovic, Degree limited Bluetooth scatternet formation based on maximal independent sets, in preparation.

[SL2] Ivan Stojmenovic and Xu Lin, Power aware localized routing in wireless networks, IEEE Transactions on Parallel and Distributed Systems, Vol. 12, No. 11, November 2001, 1122-1133.

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

[SCL] M.T. Sun, C.K. Chang and T.H. Lai, A self-routing topology for Bluetooth scatternets, *Proc. ISPAN*, Philipini, May 2002, 13-18.

[SW] Stojmenovic I. and J. Wu, Broadcasting and activity scheduling in ad hoc networks, in: Mobile Ad Hoc Networking (S. Basagni, M. Conti, S. Giordano and I. Stojmenovic, eds.), IEEE/Wiley, 2004, 205-229.

[T] G. Toussaint, The relative neighborhood graph of a finite planar set, *Pattern Recognition*, Vol.12, No.4, 1980, pp. 261-268.

[TMGB] G. Tan, A. Miu, J. Guttag and H. Balakrishnan, An efficient scatternet formation algorithm for dynamic environments, *First Annual Student Oxygen Workshop, Gloucester*, MA, July 2001; MIT Technical Report, MIT-LCS-TR-826 (2001); IASTED Communications and Computer Networks (2002).

[VVL] S.G. Gonzalez-Valenzuela, S.T. Vuong, V.C.M. Leung, BlueScouts: A scatternet formation protocol based on mobile agents, 4^{th} Workshop on Applications and Services in Wireless Networks, Boston, Aug. 2004.

[W-] J.P.F. Willekens, Ad hoc routing in Bluetooth, LNCS 2213, 2001, 130-144.

[WHC] R.M. Whitaker, L. Hodge, I. Chlamtac, Bluetooth scatternet formation: a survey, Ad Hoc Networks, 2004, to appear.

[WSL] Yu Wang, Ivan Stojmenovic and Xiang-Yang Li, Bluetooth scatternet formation for single-hop ad hoc networks based on virtual positions, IEEE Symposium on Computers and Communications, Alexandria, Egypt, June 2004; Journal of Internet Technology, to appear.

[WTH] Z. Wang, R.J. Thomas, and Z. Haas, Bluenet, a new scatternet formation scheme, Proc. Hawaii Int'l Conf. System Sciences, 2002, 779-787.

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

[YKKM] J. Yun, J. Kim, Y.S. Kim, J. Ma, A three-phase ad hoc network formation protocol for Bluetooth systems, Proc. 5th Int. Symp. Wireless Personal Multimedia Communications WPMC, Hawaii, Oct. 2002.

[ZBC] G.V. Zaruba, S. Basagni and I. Chlamtac, Bluetrees - scatternet formation to enable Bluetooth based ad hoc networks, *Proc. IEEE ICC*, 2001, 273-277.

[ZHS] H. Zhang, J. Hou, L. Sha, Design and analysis of a Bluetooth loop scatternet formation algorithm, IEEE ICC May 2003; WINET, to appear.

[ZPK] B. Zhen, J. Park, and Y. Kim, Scatternet Formation of Bluetooth Ad Hoc Networks, 36th Hawaii Int. Conference on System Sciences, 2003, 312-319.

[ZWL-] C. Zhang, V. Wong and V.C.M. Leung, TPSF+: A new two-phase scatternet formation algorithm for Bluetooth ad hoc networks, IEEE Globecom'04, Dallas, TX, Nov. 2004.