Locally Optimal Scatternet Topologies for Bluetooth Ad Hoc Networks

Tommaso Melodia and Francesca Cuomo

INFOCOM Department, University of Rome “La Sapienza”
Via Eudossiana 18 00184, Rome, ITALY
{tommaso,cuomo}@net.infocom.uniroma1.it

Abstract. *Bluetooth* is a promising technology for personal/local area wireless communications. A Bluetooth *scatternet* is composed of overlapping *piconets*, each with a low number of devices sharing the same radio channel. This paper discusses the *scatternet formation* issue by analyzing topological characteristics of the scatternet formed. A matrix-based representation of the network topology is used to define metrics that are applied to evaluate the key cost parameters and the scatternet performance. Numerical examples are presented and discussed, highlighting the impact of metric selection on scatternet performance. Then, a distributed algorithm for *scatternet topology optimization* 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.

1 Introduction

Bluetooth[1] is a promising technology for ad hoc networking that could impact several wireless communication fields providing WPAN (*Wireless Personal Area Networks*) extensions of public radio networks (e.g., GPRS, UMTS, Internet) or of local area ones (e.g. 802.11 WLANs, Home RF) [1][2]. The Bluetooth system supports a 1 Mbit/s gross rate in a so-called *piconet*, where up to 8 devices can simultaneously be inter-connected. The radius of a piconet (*Transmission Range*, TR) is about 10 meters for Class 3 devices.

One of the key issues associated with the BT technology is the possibility of dynamically setting-up and tearing down piconets. Different piconets can coexist by sharing the spectrum with different frequency hopping sequences, and interconnect in a scatternet. When all nodes are in radio visibility, scenario which we will refer to as *single hop*, the formation of overlapping piconets allows more than 8 nodes to contemporary communicate and may enhance the system capacity. In a *multi-hop* scenario, where nodes are not all in radio vicinity, a scatternet is mandatory to develop a connected platform for ad-hoc networking.

[1] This work was partially funded by MIUR (Italian Ministry of Research) in the framework of the VICOM (Virtual Immersive COMmunications) project.

R. Battiti, M. Conti, and R. Lo Cigno (Eds.): WONS 2004, LNCS 2928, pp. 116–129, 2004.

This paper addresses the scatternet formation issue by considering topological properties that affect the performance of the system. Most works in literature aim at forming a connected scatternet while performance related topological issues typically remain un-addressed. To this aim we introduce a matrix based scatternet representation that is used to define metrics and to evaluate the relevant performance. We then propose a distributed algorithm that performs topology optimization by relying on the previously introduced metrics. We conclude by describing a two-phases scatternet formation algorithm based on the optimization algorithm. To the best of our knowledge, this is the first scatternet formation algorithm explicitly aimed at optimizing the topology of the network.

The paper is organized as follows. Section 2 recalls the main aspects related to the piconet and scatternet models. Section 3 briefly summarizes the state of the art in scatternet formation, while in Section 4 a framework for scatternet analysis, based on a matrix representation is presented, together with simple metrics to evaluate scatternet performance. Section 5 presents the Distributed Scatternet Optimization Algorithm (DSOA) while Section 6 describes a two-phase scatternet formation algorithm based on DSOA. Section 7 concludes the paper.

2 Bluetooth Basics

Bluetooth exploits an 83.5 MHz band, divided into 79 equally spaced 1 MHz channels [1]. The multiple access technique is the FHSS-TDD (Frequency Hopping Spread Spectrum - Time Division Duplexing). Two Bluetooth units exchange information by means of a master-slave relationship. Master and slave roles are dynamic: the device that starts the communication acts as master, the other one as slave. After connection establishment, master and slave exchange data by hopping at a frequency of 1600 hops/second on the 79 available channels. Different hopping sequences are associated to different masters.

A master can connect with up to 7 slaves within a *piconet*. Devices belonging to the same piconet share a 1 Mbit/s radio channel and use the same frequency hopping sequence. Only communications between master and slaves are permitted. Time is slotted and the master, by means of a polling mechanism, centrally regulates the medium access. Thanks to the FHSS, which is robust against interference, multiple piconets can coexist in the same area. Considerable performance degradation only occurs for a high number of co-located piconets (in the order of 50) [3].

A *scatternet* is defined as an interconnection of overlapping piconets. Each device can join more than one piconet, and participate to communications in different piconets on a time-division basis. Devices that belong to more that one piconet are called *gateways* or *BridGing units* (BG).

Since there are many topological alternatives to form a scatternet out of the same group of devices, the way a scatternet is formed considerably affects its performance.

3 Related Work

Scatternet formation in Bluetooth has recently received a significant attention in the scientific literature. Existing works can be classified as single-hop [4][5][6][7] and multi-hop solutions [8][9][10][11][12].

Paper [4] addresses the Bluetooth scatternet formation with a distributed logic that selects a leader node which subsequently assigns roles to the other nodes in the system. In [5] a distributed formation protocol is defined, with the goal of reducing formation time and message complexity. In [5] and [6], the resulting scatternet has a number of piconets close to the theoretical minimum. The works in [7], [8] and [9] form tree shaped scatternets. In [7], Tan et al. present the TSF (Tree Scatternet Formation) protocol, which assures connectivity only in single-hop scenarios. Zaruba et al. propose a protocol which operates also in a multi-hop environment [8] but is based on time-outs that could affect the formation time. SHAPER [9] forms tree-shaped scatternets, works in a multi-hop setting, shows very limited formation time and assures self-healing properties of the network, i.e. nodes can enter and leave the network at any time without causing long term loss of connectivity.

A second class of multi-hop proposals is based on clustering schemes. These algorithms principally aim at forming connected scatternets. In [10] and [11] the *BlueStars* and *BlueMesh* protocols are described respectively. Also [12] defines a protocol that limits the number of slaves per master to 7 by applying the *Yao* degree reduction technique. The proposed algorithm assumes that each node knows its geographical position and that of each neighbor.

Recently, the work in [13] proposed a new on-demand route discovery and construction approach which, however, requires substantial modifications to the Bluetooth standard to guarantee acceptable route-setup delay.

Some other works discuss the optimization of the scatternet topology. This issue is faced in [14] and [15] by means of centralized approaches. In [14] the aim is minimizing the load of the most congested node in the network, while [15] discusses the impact of different metrics on the scatternet topology. In [16], an analytical model of a scatternet based on queuing theory is introduced, aimed at determining the number of non-gateway and gateway slaves to guarantee acceptable delay characteristics.

4 The Scatternet Formation Issue

Before addressing the issue of scatternet formation, we introduce a suitable scatternet representation.

4.1 Scatternet Representation

Let us consider a scenario with N devices. The scenario can be modelled as an undirected graph $G(V, E)$, where V is the set of nodes and an edge e_{ij}, between any two nodes v_i and v_j, belongs to the set E iif $distance(v_i, v_j) < TR$, i.e., if v_i and v_j are within each other's transmission range. $G(V, E)$ can be represented by an NxN adjacency matrix $A = [a_{ij}]$, whose element a_{ij} equals 1 iff device j is in the TR of device i (i.e., j can directly receive the transmission of i).

Besides the *adjacency graph* $G(V, E)$, we model the scatternet with a *bipartite graph* $G_B(V_M, V_S, L)$, where $|V_M| = M$ is the number of masters, $|V_S| = S$ is the number of slaves, and L is the set of links (with $N = M + S, V_M \cap V_S = \{\emptyset\}, V_M \cup V_S = V$). A link may exist between two nodes only if they belong to the two different sets V_M and V_S. Obviously, for any feasible scatternet, we have $L \subseteq E$. This model is valid under the hypothesis that a master in a piconet does not assume the role of slave in another piconet; in other words, by adopting this model, the BGs are slaves in all the piconets they belong to. We rely on this hypothesis to slightly simplify the scatternet representation, the complexity in the description of the metrics and to reduce the space of possible topologies. Moreover, intuitively, the use of master/slave BGs can lead to losses in the system efficiency. If the BG is also a master, no communications can occur in the piconet where it plays the role of master when it communicates as slave. However, to the best of our knowledge, this claim has never been proved to be true. Future work will thus extend the results presented in this paper to non-bipartite graphs.

The bipartite graph G_B can be represented by a rectangular MxS binary matrix $\mathbf{B}$. In $\mathbf{B}$, each row is associated with one master and each column with one slave. Element b_{ij} in the matrix equals 1 iff slave j belongs to master i's piconet. Moreover, a *path* between a pair of nodes (h, k) can be represented by another MxS matrix $\mathbf{P}^{h,k}(\mathbf{B})$, whose element $p_{ij}^{h,k}$ equals 1 iff the link between master i and slave j is part of the path between node h and node k $(1 \leq i, j, h, k \leq N)$. To finish with, we will say that an MxS rectangular matrix $\mathbf{B}$ represents a "Bluetooth-compliant" scatternet with M masters and S slaves if it represents a fully connected network (i.e., the matrix does not have a block structure, notwithstanding permutations of the rows), and no more than 7 slaves belong to each piconet (the sum of the elements of each row is less than 7).

4.2 Metrics for Scatternet Performance Evaluation

In [15], we introduced some metrics for scatternet evaluation. These metrics can either be dependent on or independent of the traffic loading the scatternet. For the convenience of the reader, we recall the Traffic Independent (TI) metrics which will be considered in the following.

A first traffic independent metric is the overall capacity of the scatternet. Evaluating such a capacity is not an easy task, since it is related to the capacity of the composing piconets which in turn depends on the intra-piconet and inter-piconet scheduling policies. To the best of our knowledge, no such evaluation is available in literature. In the following, we introduce a simple model to estimate the capacity of a scatternet and we exploit this evaluation for scatternet formation. In the model we assume that:

- a master may offer the same amount of capacity to each of its slaves by equally partitioning the piconet capacity;
- a BG slave spends the same time in any piconet it belongs to.

These assumptions are tied to intra and inter piconet scheduling; here, for the sake of simplicity, we assume policies that equally divide resources; however the model can be straightforwardly extended to whatever scheduling policy.

The scatternet capacity will be evaluated by normalizing its value to the overall capacity of a piconet (i.e., 1 Mbit/s). Let us define two MxS matrices, $\mathbf{O_{TI}}(\mathbf{B}) = [o_{ij}]$, and $\mathbf{R_{TI}}(\mathbf{B}) = [r_{ij}]$ with $o_{ij} = b_{ij}/s_i$ and $r_{ij} = b_{ij}/m_j$, where s_i denotes the number of slaves connected to master i and m_j denotes the number of masters connected to slave j (for $j = 1,..,S$ and $i = 1,..,M$):

$$m_j = \sum_{i=1}^{M} b_{ij}, j = 1,..,S \qquad s_i = \sum_{j=1}^{S} b_{ij}, i = 1,..,M \tag{1}$$

The matrix $\mathbf{O_{TI}}(\mathbf{B})$ represents the portions of capacity a master may offer to each of its slaves. The $\mathbf{R_{TI}}(\mathbf{B})$ matrix represents the portions of capacity a slave may "spend" in the piconet it is connected to. The overall capacity of the scatternet is given by the sum of the capacities of all links. The capacity c_{ij} of link (i, j) is the minimum between the capacity o_{ij} and the capacity r_{ij}. Let us define the matrix $\mathbf{C_{TI}}(\mathbf{B})$, whose elements represent the normalized link capacity, as:

$$\mathbf{C_{TI}}(\mathbf{B}) = [c_{ij}] = [\min(o_{ij}, r_{ij})] \tag{2}$$

The associated metric is the *normalized capacity* $c_{TI}(\mathbf{B})$ of a scatternet defined as:

$$c_{TI}(\mathbf{B}) = \sum_{i=1}^{M} \sum_{j=1}^{S} \min(o_{ij}, r_{ij}) \tag{3}$$

As shown in [15], path lengths have a considerable impact on scatternet performance. As a consequence we introduce two metrics that do take into account path lengths. Let us denote, for a scatternet represented by a matrix $\mathbf{B}$, the length of the path between device h and device k (expressed in number of hops) as:

$$q^{h,k}(\mathbf{B}) = \sum_{i=1}^{M} \sum_{j=1}^{S} p_{ij}^{h,k} \tag{4}$$

We can now introduce the *average path length*, which is the path length averaged over all possible source-destination couples, and is given by:

$$q_{TI}(\mathbf{B}) = \sum_{h=1}^{N} \sum_{k=1, k \neq h}^{N} \frac{q^{h,k}(\mathbf{B})}{N \cdot (N-1)} \tag{5}$$

Obviously, we want $q_{TI}(\mathbf{B})$ to be minimized.

Given the capacity of a scatternet $c_{TI}(\mathbf{B})$ and the relevant average path length $q_{TI}(\mathbf{B})$, the capacity available, on average, for the generic source-destination couple among the nodes in $\mathbf{B}$ is given by:

$$a_{TI}(\mathbf{B}) = \frac{c_{TI}(\mathbf{B})}{q_{TI}(\mathbf{B}) \cdot N \cdot (N-1)} \tag{6}$$

This last metric, which we will refer to as *average path capacity*, will be considered in all the experiments reported in the following. As we showed in [15], scatternets with high values of this metric show a good compromise between capacity and path length.

5 A Distributed Algorithm for Topology Optimization

In this section we describe a Distributed Scatternet Optimization Algorithm (DSOA), that aims at optimizing the topology to obtain a performance (in terms of the chosen metric) as close as possible to the optimum. Note that the selection of the optimized topology is decoupled from the establishment of the links that compose it, as will become clearer in Section 6, where we will describe a two-phases distributed scatternet formation algorithm based on DSOA.

5.1 Distributed Scatternet Optimization Algorithm (DSOA)

We consider the adjacency graph $G(V, E)$. First, we aim at obtaining an ordered set of the nodes in V. The first procedure orders the nodes in the graph according to a simple property: a node k must be in transmission range of at least one node in the set $1..k-1$.

Procedure 1 ORDER_NODES

```
Input: $G(V, E)$
Output: ordered set of the nodes in $V$, $W = \{w_k\}$, $k = 1, 2, .., N$, $N = |V|$
begin
$w_1$=random selection of a node $v$ from $V$
$W = w_1$
for k = 2 : $N$ do
    $w_k$=random selection of a node $v$ from $V$ such that:
    1. $v \notin W$
    2. $\exists u \in W$ such that $distance(u, v) \leq TR$
    $W = W \cup w_k$
end for
end
```

Since with DSOA the nodes sequentially select how to connect, each node must be in TR of at least another node already entered. The following proofs that it is always possible to obtain such an ordering of the nodes, i.e. that this procedure always ends.

Theorem 1 *Given a connected graph $G(V, E)$, the procedure* ORDER_NODES *always terminates, and $|W| = N$.*

Proof: Suppose that at some step k of the procedure, $k < N$, we have $W = \{w_1, w_2, .., w_{k-1}\}$ and no couple (v, w) with $v \in (V \setminus W), w \in W$ exists such that

$distance(v, w) < TR$. Therefore, since $W \subseteq V$, there exist two disconnected components W and $V \setminus W$ of $G(V, E)$.

At the end of this procedure, then, node k is in transmission range of at least one of the nodes $1, 2, \ldots k-1$. The second procedure is the core of the algorithm. Here we let e_{ij} be the link between the nodes w_i and w_j of a scatternet ($1 \leq i, j \leq N$). This part of the algorithm is also dependent on the selected metric M. At each step k, node w_k "enters" in the scatternet in the best possible way, according to M.

Procedure 2 SCATTERNET_OPTIMIZATION_ALGORITHM

```
Input: $W$, $G(V,E)$, $M$
Output: Locally Optimal Scatternet $\mathbf{B}^*$
begin
$V_M = \emptyset$
$V_S = \emptyset$
$V_M = V_M \cup w_1$
$V_S = V_S \cup w_2$
$\mathbf{B^2} = [1]$
for k = 3 : $N$ do
  case 1: consider $w_k$ in $V_M$
  * derive all Bluetooth-compliant matrices $\mathbf{B^k}$ with $|V_M|+1$ rows and $|V_S|$ columns
  calculate values of $M(\mathbf{B^k})$
  case 2: consider $w_k$ in $V_S$
  * derive all Bluetooth-compliant matrices $\mathbf{B^k}$ with $|V_M|$ rows and $|V_s|+1$ columns
  calculate values of $M(\mathbf{B^k})$
  select the $\mathbf{B^k}$ with optimal $M(\mathbf{B^k})$
  if optimum in case 1 then
    $V_M = V_M \cup w_k$
  else
    if optimum in case 2 then
      $V_S = V_S \cup w_k$
    else
      RECONFIGURE($\mathbf{B^{k-1}}$,$V_M$,$V_S$)
    end if
  end if
end for
$\mathbf{B}^* = \mathbf{B^N}$
end
```

The RECONFIGURE procedure is executed in the unlikely case when w_k is only in transmission range of master nodes that have already 7 slaves in their piconet. For the sake of simplicity, details of this procedure are only given in the following proof of correctness. In this case, one of the 7 slaves is forced to become master of one of the other slaves. This is shown to be always possible. The following proves the correctness of SOA, i.e. it is always possible for a node to enter the network respecting the Bluetooth properties.

Proof of correctness. Node w_2 is in transmission range of w_1, thus the two nodes can connect. Each node w_k, with $k > 2$ can always establish a new piconet, thus connecting as a master, whenever a node $v \in \{w_1, w_2, .., w_{k-1}\}$ exists s.t. $v \in V_S$ and $distance(w_k, v) \leq TR$, i.e. one of the slave nodes already in the network is in transmission range of w_k. If no slaves are in TR of w_k, whenever a node $v \in V_M$ exists, with $distance(w_k, v) \leq TR$, and $slaves(v) \leq 7$, w_k can be a slave of v. Otherwise, at least one node $w_i \in V_M$ must exist, with $distance(w_k, v) \leq TR$, and $slaves(v) = 7$, with $i \leq k$. The Reconfigure procedure can always be executed in this way. If at step i node w_i selected more than 1 slave, it can disconnect from the slave that causes the minimum decrease/increase in the metric value. The topology is still connected, and w_k can select w_i as its slave. If, otherwise, w_i selected only one slave at step i, this cannot be disconnected, since this could cause loss of connectivity for the network. Thus, one of the other 6 slaves must be disconnected. However, it was proven in [8] that in a piconet with at least 5 slaves, at least 2 of them are in TR of each other. Thus, at least one of the slaves can become master and select another slave. The network can therefore be reconfigured by forcing the 7th slave that connected to w_i to become master of another slave of w_i, to minimize reconfigurations. If it is not in TR of any other slave of w_i, we can try with the 6th, and so on. At least one of the six slaves must be able to become master and select one of the other 5 as its slave.

The local optimization in SOA (steps with mark *) can be performed by means of state space enumeration, as in the simulations results we show, or, e.g., by means of randomized local search algorithms.

The distributed version of the SOA (Distributed SOA, DSOA) straightforwardly follows. At each step k, a new node w_k receives information on the topology selected up to that step ($\mathbf{B^{k-1}}$ matrix) and selects the role (master or slave) it will assume and the links it will establish, with the aim of maximizing the global scatternet metric. If the node becomes a master it will select a subset of the slaves in its TR already in the scatternet; if it becomes a slave it will select a subset of the masters in its TR, already in the scatternet. Order_Nodes is needed to guarantee that, when node k enters, it can connect to at least one of the previously entered nodes. DSOA can be classified as a *greedy* algorithm, since it tries to achieve the optimal solution by selecting at each step the *locally optimal* solution, i.e. the solution that maximizes the metric of the overall scatternet, given local knowledge and sequential decisions. Greedy algorithms do not always yield the global optimal solution. As will be shown in the next subsection, however, the results obtained with DSOA are close to the optimum.

5.2 Examples and Numerical Results (DSOA)

In this section we show some results obtained with DSOA, by using average path capacity as a metric. As previously discussed, we believe that average path capacity is a good metric since it takes into account both capacity and average path length of the scatternet. Figure 1 shows a comparison between the optimal $a_{TI}(\mathbf{B})$ and the one obtained with the DSOA. The dotted curve shows the

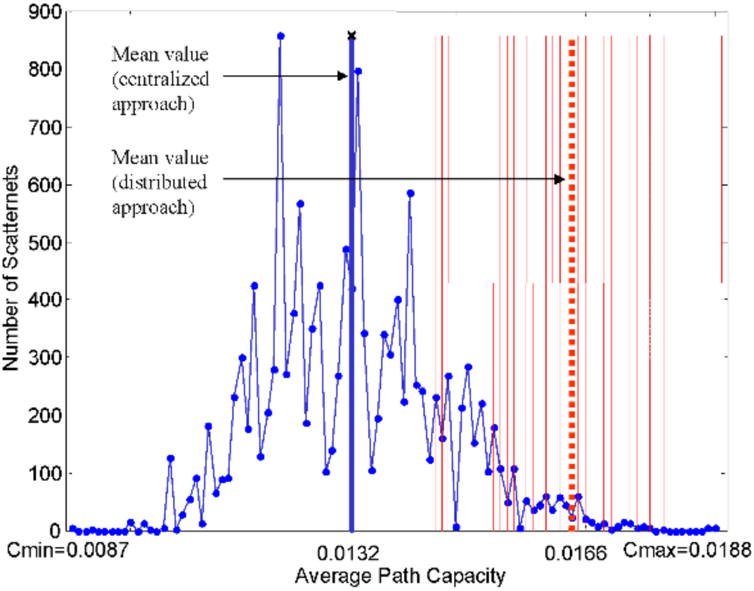

Fig. 1. Results obtained with DSOA compared with the distribution of the metric values

histogram of the average path capacity of all possible Bluetooth-compliant scatternets feasible in this scenario. The histogram of the average path capacity of all the feasible scatternets in a scenario constituted by 10 nodes distributed in an area of 25x25 metersis represented. As will be shown later, a similar distribution holds in general. It is easy to see that, already with 10 nodes, the number of different feasible topologies is very high. The values of $a_{TI}(\mathbf{B})$ are distributed in a range starting from $a_{TI,min}(\mathbf{B}) = 0.0087$ ($\approx$ 8 kbit/s for every possible node pair) to $a_{TI,max}(\mathbf{B}) = 0.0188$ ($\approx$ 19 kbit/s per pair); the mean value of $a_{TI}(\mathbf{B})$ is also shown (equal to 0.0132). Note that the mean value is quite distant from the maximum value, which corresponds to the value associated with the optimal scatternet. Moreover, a few scatternets have a high value of $a_{TI}(\mathbf{B})$ and are thus contained in the right tail of the histogram. This is an interesting result because it suggests that topology optimization is a fundamental issue for Bluetooth scatternets: in fact, this distribution of the metric values means that it is highly unlikely to obtain a high performance scatternet by randomly selecting a topology. We need to deploy protocols that not only search for a connected scatternet but also explicitly aim at maximizing its performance. As regards the DSOA, the vertical lines in Figure 1 correspond to the values of $a_{TI}(\mathbf{B})$ for 100 different scatternets formed by using 100 different randomly chosen sequential orders. The lines are concentrated in the right part of the figure (i.e., the scatternets formed have a value of $a_{TI}(\mathbf{B})$ greater than the overall mean value of all possible scatternets). The mean value of $a_{TI}(\mathbf{B})$ of these 100 DSOA scatternets is equal to 0.0166. The un-normalized values of the average capacity per path

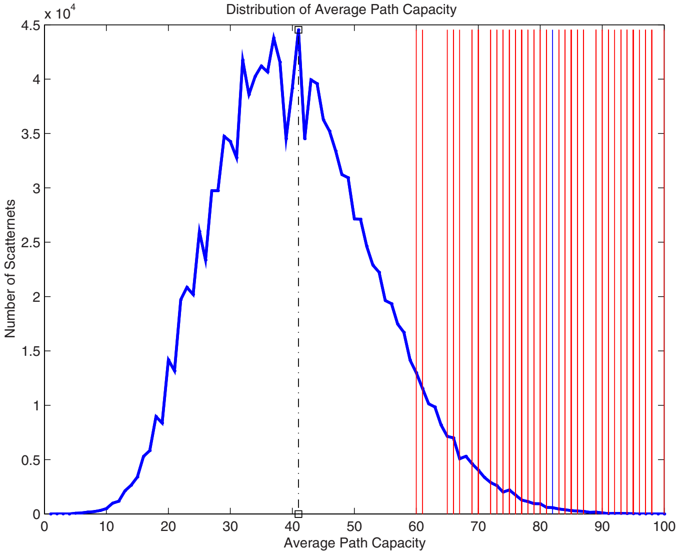

Fig. 2. Distribution of Average Path Capacity for 15 nodes

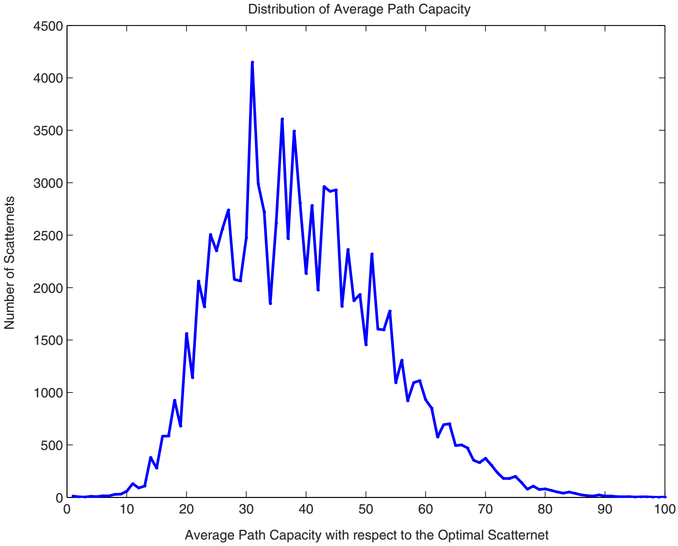

Fig. 3. Distribution of average path capacity on different scenarios

obtained with DSOA is about 17 kbit/s, while the maximum possible value is 19 kbit/s; this confirms the good behavior of the DSOA.

Figure 2 shows a similar distribution in a scenario with 15 nodes in a multi-hop context. In Figure 3 a distribution averaged over 100 different scenarios, with varying number of nodes, is shown; Figure 4 reports the distribution of the values obtained with DSOA in the same scenarios. The probability of obtaining a value of the metric between the optimal and 70% of the optimal by randomly selecting a topology is very low; by using DSOA this probability is close to 1. For a higher number of nodes the state-space enumeration approach, which has been useful in obtaining the distribution of the metric values, becomes unfeasible. The conclusion we can draw from the above figures is that scatternets formed with DSOA have a structure quite similar to the optimal ones, obtained with the centralized approach. Correspondingly, the value of the metric obtained with DSOA is close (sometimes equal) to the one obtained with the centralized approach. The same behavior has been observed in numerous experiments, carried out with different metrics and number of nodes.

6 A Two-Phases Scatternet Formation Algorithm

The actual Distributed Scatternet Formation Protocol is divided in two phases:

1. Tree Scatternet Formation (SHAPER);
2. DSOA and new Connections Establishment.

To implement DSOA we need a mechanism to distribute the "right" to enter in the network to every node k at step k, and to convey the topology selected by the previous $k-1$ nodes ($\mathbf{B^{k-1}}$ matrix). The distributed implementation in Bluetooth however is not simple since the system lacks a shared broadcast medium that would allow signaling among nodes. A good solution which guarantees: i) the required ordering of the nodes; ii) synchronization of the decisions; iii) a shared communication medium, is to form a tree-shaped "provisional" scatternet. A tree-shaped scatternet can asynchronously be formed in a distributed fashion. In [9], we proposed a new protocol for tree scatternets (SHAPER), which works in an asynchronous and totally distributed fashion, thus allowing the self-organized formation of a tree shaped scatternet in a multi-hop context. We showed that a tree scatternet can be formed in a few seconds time, and that less time is required when nodes are denser.

After the tree has formed, a simple recursive visit procedure can be executed on it, which allows implementing the DSOA topology optimization process. It is easy to see that a sequential visit of all nodes in the tree, from the root down to the leaves, guarantees the order provided by Order_Nodes. We let *parent*(v) be the parent of v in the tree and *children*(v) be the set of children nodes for v. Step k of the distributed procedure is executed on a node when it receives an execute_enter($\mathbf{B^{k-1}}$,k) message from its parent. $\mathbf{B^{k-1}}$ is the matrix representing the topology selected by the previously visited nodes. The root node resulting from SHAPER starts the distributed execution of such procedure at the expiration of a timeout.

When a given node v starts the Enter procedure, it executes the DSOA, i.e. it decides how to enter in the network. Then, the node randomly picks up one of its children nodes, and sends the execute_enter($\mathbf{B^k}, k+1$) message to it. This causes the execution of the Enter procedure on the child. After sending the execute_enter command, v waits for an answer message (branch_entered)

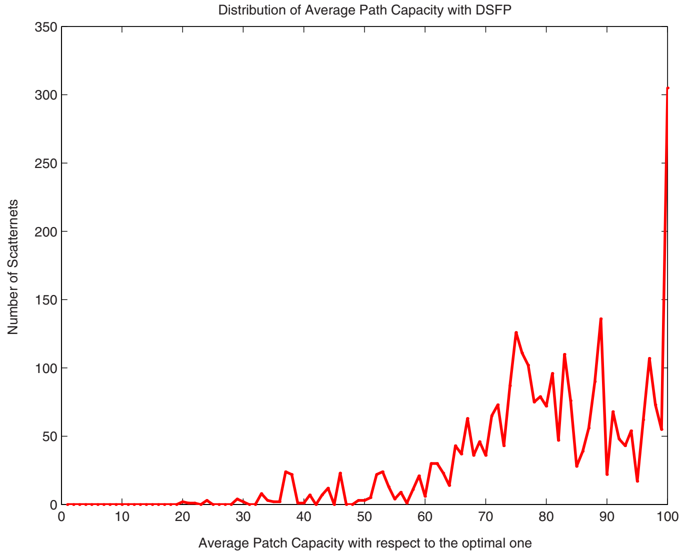

Fig. 4. Distribution of average path capacity for DSOA scatternets

from the child. This contains information about the topology selected by the whole *branch* which goes down from v to the *leaf* nodes. After the answer from the child is received, v selects another child and does the same. When v receives the answer from its last child, it informs its parent of the topology selected by itself and by all of its descendants with the BRANCH_ENTERED message. When the *root* node receives the answer from its last son, all nodes have taken their decision.

The last step concerns the actual connection establishment. The root node broadcasts the matrix representing the final scatternet structure. The matrix is recursively broadcasted at every level of the tree. Once a node has broadcasted the matrix down to its children in the tree, it enters the *topology reconfiguration*

Procedure 3 ENTER

```
begin
$\mathbf{B^k}$=DSOA($\mathbf{B^{k-1}}$)
for each $v \in children$ do
    send(EXECUTE_ENTER($\mathbf{B^k}, k+1$),$v$)
    wait_answer()
    [$\mathbf{B^{k+c}}, c$]=answer($v$)
    $k = k + c$
end for
send(BRANCH_ENTERED($\mathbf{B^k}$,$k$),$parent$)
end
```

phase. During this phase the node can start establishing the connections that will compose the optimized scatternet. Every link that is not already part of the tree topology has to be established. Redundant links have to be torn down. Every node alternates between a *communication* and a *formation* state. During the latter the node tries to establish the new links, while during the former user data is transmitted so as to guarantee the continuity of service during the reconfiguration phase. If a node has a master role in the optimized scatternet, it *pages* its first slave. When the connection is established, it continues with the other ones. If the node has a slave role, it will *page scan* for incoming connections. Priority is given to previously entered masters so as to avoid deadlocks. Every node starts tearing down the old links only when the new ones have been established, so as to preserve connectivity. Since all nodes know the overall topology, the routing task is also simplified. Route discovery algorithms have to be implemented only when mobility has to be dealt with or for other particular situations.

The most time consuming phase of the algorithm is the formation of the tree, which, as said before, becomes necessary because Bluetooth lacks a shared broadcast medium. However, in [9] we showed that the tree can be formed in a few seconds. During the tree formation phase data exchange among nodes can start, so users don't have to wait for the overall structure to be set up. Data exchange can continue on the provisional tree scatternet during the optimization process. Work is in progress to add self-healing functionalities to the algorithm (nodes can enter and exit the network which is re-optimized periodically) and to simulate the integration of SHAPER and DSOA.

7 Conclusions

In this paper, 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. A distributed algorithm for Scatternet Topology Optimization, DSOA, was described. The performance of DSOA was evaluated and shown to be encouraging: the distributed approach gives results very similar to a centralized one. The integration with the SHAPER Scatternet Formation Algorithm and other implementation concerns have been discussed. Ongoing activities include the full design of a distributed scatternet formation algorithm which implements DSOA and deals with mobility and failures of nodes, as well as a simulative evaluation of the time needed to set-up a scatternet and its performance in presence of different traffic patterns.

References

1. J. Haartsen, “The Bluetooth Radio System”, *IEEE Personal Communications*, Vol. 7, n. 1, pp. 28–36, February 2000.
2. P. Johansson, R. Kapoor, M. Gerla, M. Kazantzidis,“Bluetooth an Enabler of Personal Area Networking”, *IEEE Network*, Special Issue on Personal Area Networks, pp. 28–37, September/October 2001.

3. S. Zurbes, "Considerations on Link and System Throughput of Bluetooth Networks", Proc. of the *11th IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, Vol. 2, pp. 1315–1319, 2000.
4. T. Salonidis, P. Bhagwat, L. Tassiulas, R. La Maire, "Distributed topology construction of Bluetooth personal area networks", Proc. of the *IEEE Infocom 2001*, pp. 1577–1586, April 2001.
5. C. Law, A. Mehta, K-Y Siu, "Performance of a new Bluetooth scatternet formation protocol", Proc. of the *Mobihoc 2001*, pp. 183–192, 2001.
6. H. Zhang, J. C. Hou, L. Sha, "A Bluetooth Loop Scatternet Formation Algorithm" Proc. of the *IEEE International Conference on Communications (ICC 2003)*, pp. 1174–1180, May 2003.
7. G. Tan, A. Miu, J. Guttag, H. Balakrishnan, "An Efficient Scatternet Formation Algorithm for Dynamic Environments", in *IASTED Communications and Computer Networks (CCN)*, Cambridge, November 2002.
8. G. Zaruba, S. Basagni, I. Chlamtac, "Bluetrees – Scatternet formation to enable Bluetooth-based personal area networks", Proc. of the *IEEE International Conference on Communications (ICC 2001)*, pp. 273–277, 2001.
9. F. Cuomo, G. Di Bacco, T. Melodia, "SHAPER: a Self Healing Algorithm Producing multihop bluetooth scattERnets", Proc. of the *IEEE Globecom 2003*, San Francisco, December 2003.
10. C. Petrioli, S. Basagni, I. Chlamtac "Configuring BlueStars: multihop scatternet formation for Bluetooth networks", *IEEE Transactions on Computers*, Vol. 52, Issue 6, pp. 779–790, June 2003.
11. C. Petrioli, S. Basagni, I. Chlamtac, "BlueMesh: Degree-constrained multihop scatternet formation for Bluetooth networks", *ACM/Kluwer Journal on Special Topics in Mobile Networking and Applications* (MONET), Special Issue on Advances in Research of Wireless Personal Area Networking and Bluetooth Enabled Networks, 2002.
12. I. Stojmenovic, "Dominating set based scatternet formation with localized maintenance", Proc. of the *Workshop on Advances in Parallel and Distributed Computational Models*, April 2002.
13. Y. Liu, M. J. Lee, T. N. Saadawi, "A Bluetooth Scatternet-route Structure for Multihop Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 21, n. 2, pp. 229–239, February 2003.
14. M. Ajmone Marsan, C. F. Chiasserini, A. Nucci, G. Carello, L. De Giovanni, "Optimizing the Topology of Bluetooth Wireless Personal Area Networks", *IEEE INFOCOM 2002*, Vol.2, pp. 572–579, June 2002.
15. F. Cuomo, T. Melodia, "A General Methodology and Key Metrics for Scatternet Formation in Bluetooth", Proc. of the *IEEE Globecom 2002*, Taipei, November 2002.
16. R. Kapoor, M. Y. M. Sanadidi, M. Gerla, "An Analysis of Bluetooth Scatternet Topologies," Proc. of the *IEEE International Conference on Communications (ICC 2003)*, pp. 266–270, 2003.