Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®
Ad Hoc Networks 2 (2004) 185–202
Ad Hoc Networks
www.elsevier.com/locate/adhoc

Ad hoc networking with Bluetooth: key metrics and distributed protocols for scatternet formation

Tommaso Melodia *, Francesca Cuomo

INFOCOM Department, University of Rome "La Sapienza", Via Eudossiana 18, 00184 Rome, Italy

Received 13 June 2003; received in revised form 29 July 2003; accepted 31 July 2003

Abstract

Bluetooth is a promising technology for personal/local area wireless communications. A Bluetooth *scatternet* is composed of simple overlapping *piconets*, each with a low number of devices sharing the same radio channel. A scatternet may have different topological configurations, depending on the number of composing piconets, the role of the devices involved and the configuration of the links. 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.

Keywords: Ad hoc networks; Bluetooth; Scatternet formation; Topology optimization; Distributed protocols

1. Introduction

Bluetooth (BT) 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 BT system is described in the Bluetooth Specifications 1.1 [3] and supports a 1 Mbit/s gross rate in a so-called *piconet*, where up to 8 devices can simultaneously be interconnected. The radius of a piconet (*transmission range—TR*) is about 10 m for Class 3 devices. A BT based standard has been released by the IEEE 802.15, which also addresses coexistence with the 802.11 wireless LAN technology, in the un-licensed 2.4 GHz ISM (Industrial, Scientific and Medical) band [4].

One of the key issues associated with the BT technology is the possibility of dynamically setting-up and tearing down piconets. Devices (named also nodes in the following) can join and leave piconets. Different piconets can coexist by sharing the spectrum with different frequency hopping sequences, and interconnect in a scatternet. When all nodes

* Corresponding author.
E-mail addresses: tommaso@net.infocom.uniroma1.it (T. Melodia), cuomo@infocom.uniroma1.it (F. Cuomo).

1570-8705/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S1570-8705(03)00054-4

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 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.

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 network topology.

The paper is organized as follows. Section 3 briefly summarizes the state of the art in scatternet formation, while in Section 4 we present a framework for scatternet analysis, based on a matrix representation of the scatternet. Section 5 presents some metrics that can be used to evaluate a scatternet; related numerical results are shown in Section 6. Section 7 presents the Distributed Scatternet Optimization Algorithm (DSOA) while Section 8 describes a two-phase scatternet formation algorithm based on DSOA. Section 9 concludes the paper.

2. From piconets to scatternets

BT exploits an 83.5 MHz band, divided into 79 equally spaced 1 MHz channels [1,3]. The multiple access technique is the FHSS–TDD (frequency hopping spread spectrum–time division duplexing). Two BT 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 access to the medium. Thanks to the FHSS, which is robust against interference, multiple piconets can co-exist in the same area. Considerable performance degradation only occurs for a high number of co-located piconets (in the order of 50) [5].

To overcome the limits imposed by the low number of devices that can simultaneously communicate (up to 8) and by the channel capacity (less than 1 Mbit/s) in a piconet, the BT specifications introduced the concept of "scatternet", defined as an interconnection of overlapping piconets. Each device can join more than one piconet, and participates 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).

In Fig. 1 we show an example of scatternet, composed of 8 devices organized in 3 piconets. Devices number 1, 5 and 6 are masters; devices 4 and 7 are slaves in two different piconets. Thus, they act as BGs between them, i.e., they can forward traffic from and to devices belonging to the two different piconets.

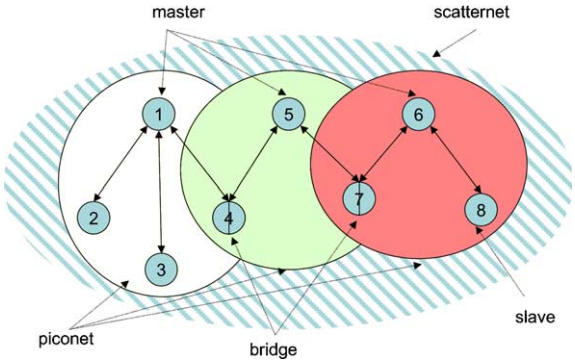

Fig. 1. An example of scatternet made up of 3 piconets.

It is easy to see that 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 works

Scatternet formation in BT has recently received a significant attention in the scientific literature. Existing works can be classified as single-hop [6–9] and multi-hop solutions [10–15].

Paper [6] addresses the BT scatternet formation with a distributed logic that selects a leader node which subsequently assigns roles to the other nodes in the system. The scheme works for a number of nodes $\leqslant 36$. In [7] a distributed formation protocol is defined, with the goal of reducing formation time and message complexity. In both [7,8], the resulting scatternet has a number of piconets close to the theoretical minimum. The works in [9–11] form tree shaped scatternets. The tree structure is shown to be simple to realize and efficient for packet scheduling and routing. The work [9], by Tan et al., presents the TSF (Tree Scatternet Formation) protocol. The topology of the scatternet is a collection of one or more rooted spanning trees, each autonomously attempting to merge and converge to a topology with a smaller number of trees. TSF assures connectivity only in single-hop scenarios since trees can merge only if their root nodes are in transmission range of each other. Zaruba et al. propose a protocol which operates also in a multi-hop environment [10]. This latter protocol is based on a process that is initiated by a unique node (named *blueroot*) and repeated recursively till the "leaves" of the tree are reached. In order to operate in a distributed way and to avoid deadlocks the algorithm is based on time-outs that could affect the formation time. SHAPER [11] also forms tree-shaped scatternets, but is fully distributed, works in a multi-hop setting, has 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 loss of connectivity. The *Bluenet* protocol in [12] forms a scatternet which has reasonably good connectivity.

A second class of multi-hop proposals is concerned with algorithms based on clustering schemes. These algorithms principally aim at forming connected scatternets. In [13,14] the *BlueStars* and *BlueMesh* protocols are described respectively. The *BlueStars* protocol has three phases: *device discovery*, *partitioning* of the network into Bluetooth piconets and *interconnection* of the piconets into a connected scatternet. It is executed at each node with no prior knowledge of the network topology, thus being fully distributed. The selection of the Bluetooth masters is driven by the suitability of a node to be the "best fit" for serving as a master. Finally, the generated scatternet is a connected mesh with multiple paths between any pair of nodes, which guarantees robustness. Simulation results are provided which evaluate the impact of the Bluetooth device discovery phase on the performance of the protocol. Also the protocol in BlueMesh forms scatternets without requiring the BT devices to be all in each other's transmission range. BlueMesh scatternet topologies are meshes with multiple paths between any pair of nodes. BlueMesh piconets are made up of no more than 7 slaves. Simulation results in networks with over 200 nodes show that BlueMesh is effective in quickly generating a connected scatternet in which each node, on average, does not assume more than 2.4 roles. Moreover, the route length between any two nodes in the network is comparable to that of the shortest paths between the nodes. Also [15] defines a protocol that limits the number of slaves per master to 7 by applying the *Yao* degree reduction technique, assuming that each node knows its geographical position and that of each neighbor.

Recently, the work in [16] 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 [17,20] by means of centralized approaches. In [17] the aim is minimizing the load of the most congested node in the network while [20] discusses the impact of different metrics on the scatternet topology. A distributed approach based on simple

heuristics is presented in [21]. In [18], 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.

In this framework the objectives of our contribution are

- provide a framework for scatternet topology analysis based on matrices which turns out to be a very simple and effective design tool;
- identify metrics that can be used to form and to evaluate scatternets; we emphasize differences between traffic dependent metrics and traffic independent ones and we show selected numerical results;
- to present the building blocks for the implementation of a distributed algorithm that optimizes the scatternet topology.

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 modeled 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 iff $\mathrm{dist}(v_i,v_j) < TR$, i.e., if v_i and v_j are within each other's transmission range. $G(V,E)$ can be represented by a $N \times N$ *adjacency matrix* $\mathbf{A} = [a_{ij}]$, 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* $G(V,E)$, in accordance to, we use a *bipartite* graph $G_B(V_M,V_S,L)$, to model the scatternet, 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 $N = M + S$, $V_M \cap V_S = \{0\}$, $V_M \cap V_S = V$). 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 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 $M \times S$ binary matrix $\mathbf{B}$ [19]. 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. The scatternet of Fig. 1 may be represented by the following matrix $\mathbf{B}$ (Eq. 1a).

In addition, a *path* between a pair of nodes (h, k) can be represented by another $M \times S$ 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 ($1 \leqslant i,j,h,k \leqslant N$). As an example, and referring again to Fig. 1, the path between nodes 2 and 8 can be represented by the matrix $\mathbf{P}^{2,8}(\mathbf{B})$ of Eq. (1b)

$$\mathbf{B}=\begin{array}{c} \\ 1 \\ 5 \\ 6 \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} 2 & 3 & 4 & 7 & 8 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}_{(a)} \qquad \mathbf{P}^{2,8}(\mathbf{B})=\begin{array}{c} \\ 1 \\ 5 \\ 6 \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} 2 & 3 & 4 & 7 & 8 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}_{(b)} . \tag{1}$$

Given a set V of N nodes, and an adjacency matrix **A,** we can build the $M \times S$ matrix $\mathbf{B} = [b_{ij}]$ by associating its rows to a V_M non-empty subset of M nodes in V, and the columns to a V_S non-empty subset of S nodes in V (with $N = M + S$, $V_M \cap V_S = \{0\}$ and $V_M \cup V_S = V$), and by selecting a subset of links in E. The resulting matrix $\mathbf{B}$ represents a "BT-compliant" scatternet with M masters and S slaves if the following properties apply:

1. Each master is connected at least to a slave, $\sum_{j=1}^{S} b_{ij} \geqslant 1, \quad i = 1, \ldots, M$.
2. No more than seven slaves belong to a piconet, $\sum_{j=1}^{S} b_{ij} \leqslant 7, \quad i = 1, \ldots, M$.
3. Each slave is connected at least to a master, $\sum_{i=1}^{M} b_{ij} \geqslant 1, \quad j = 1, \ldots, S$.
4. The resulting network is connected, the matrix **B** does not have a block structure, row permutations notwithstanding.

4.2. A framework for scatternet topology analysis

By exploiting the above matrix representations, we are interested in

(a) characterize the space of solutions; i.e., all the **B** matrices compliant with rules 1–4;
(b) define metrics to evaluate the scatternet performance;
(c) single out the optimal scatternet ($\mathbf{B}^*$) with respect to a selected metric;
(d) analyze the topological properties of the extracted solutions.

We stress that our objective is to enucleate scatternet characteristics related to specific metrics that could be adopted in the formation process. We do not aim at proposing sophisticated algorithms that elaborate the matrix **A** to derive the optimal scatternet, also because, due to the complexity of the problem, they probably shall operate in a centralized manner while scatternet formation should be solved by adopting distributed operations performed by all network nodes. To work out points (a) and (c) we rely on space state enumeration that pays the complexity of examining and listing all potential scatternets but on the other hand allows us to completely characterize the space of possible solutions. In the following we briefly describe our approach to go along steps a–d, while further details can be found in [20].

We randomly generate communication scenarios taking as inputs the number of devices, N, and the dimensions of the area where the nodes are located; the scenario is represented by the **A** matrix.

We then identify and enumerate all the "BT-compliant" scatternets that may be obtained from the scenario; if we let M be the number of masters in the scatternet, with $M_{\min} \leqslant M \leqslant M_{\max}$ the number of possible choices of roles for the network nodes is equal to $\sum_{M=M_{\min}}^{M_{\max}} \binom{N}{M}$, since there are $\binom{N}{M}$ possible ways of selecting M masters among N nodes. Each choice implies a set L' of possible links (with $L' \subseteq E$) and we consider every subset $L \subseteq L'$ which gives rise to a scatternet that respects properties 1–4. All these scatternets constitute our **B** matrices. We consider $M_{\min} = \lceil N/(7+1) \rceil$ and $M_{\max} = \lfloor N/2 \rfloor$. A number of masters greater than half the nodes introduces inefficiencies (e.g., interference) without bringing benefits to the scatternet.

The **B** matrices are evaluated by applying suitable metrics described in Section 5. For a given metric, the output of the overall process is the identification of the optimal **B** (indicated with $\mathbf{B}^*$), which represents the scatternet with the optimal topology, and a distribution of the metric values.

5. Metrics for scatternet evaluation

Metrics for scatternet evaluation can either be dependent on or independent of the traffic loading the scatternet. In the traffic independent (TI) case, the scatternet is formed without a priori knowledge of traffic patterns among involved devices. The scenario is described only by means of the adjacency matrix **A**, without associating to possible pairs of devices a description of the related exchanged traffic. On the other hand, it may be useful to form a scatternet by taking into account traffic patterns, if such information is available. In that case, the traffic patterns can conveniently be described by a traffic matrix, **T**. In the following we refer to this case as traffic dependent (TD).

In the following, we introduce several metrics; each of them has pros and cons.

5.1. TI metrics: scatternet with maximum capacity

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 interpiconet 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 $M \times S$ matrices, $\mathbf{O}_{\mathrm{TI}}(\mathbf{B}) = [o_{ij}]$, and $\mathbf{R}_{\mathrm{TI}}(\mathbf{B}) = [r_{ij}]$ 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 $j = 1, \ldots, S$ and $i = 1, \ldots, M$):

$$m_j = \sum_{i=1}^{M} b_{ij} \quad \text{for } j = 1, \ldots, S,$$

$$s_i = \sum_{j=1}^{S} b_{ij} \quad \text{for } i = 1, \ldots, M. \tag{2}$$

The matrix $\mathbf{O}_{\mathrm{TI}}(\mathbf{B})$ represents the portions of capacity a master may offer to each of its slaves. The $\mathbf{R}_{\mathrm{TI}}(\mathbf{B})$ 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 $\mathbf{C}_{\mathrm{TI}}(\mathbf{B})$, whose elements represent the normalized link capacity, as

$$\mathbf{C}_{\mathrm{TI}}(\mathbf{B}) = [c_{ij}] = [\min(o_{ij}, r_{ij})]. \tag{3}$$

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

$$c_{\mathrm{TI}}(\mathbf{B}) = \sum_{i=1}^{M} \sum_{j=1}^{S} \min(o_{ij}, r_{ij}). \tag{4}$$

As an example, let us consider the scatternet of Fig. 1. The matrices $\mathbf{O}_{\mathrm{TI}}(\mathbf{B})$ and $\mathbf{R}_{\mathrm{TI}}(\mathbf{B})$ are

$$\mathrm{O}_{\mathrm{TI}}(\mathrm{B}) = \begin{array}{c} \\ 1 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{ccccc} 2 & 3 & 4 & 7 & 8 \end{array} \\ \begin{pmatrix} 1/3 & 1/3 & 1/3 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 \end{pmatrix} \end{array}$$

$$\mathrm{R}_{\mathrm{TI}}(\mathrm{B}) = \begin{array}{c} \\ 1 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{ccccc} 2 & 3 & 4 & 7 & 8 \end{array} \\ \begin{pmatrix} 1 & 1 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 1 \end{pmatrix} \end{array} \tag{5}$$

and the related matrix $\mathbf{C}_{\mathrm{TI}}(\mathbf{B})$ is

$$\mathrm{C}_{\mathrm{TI}}(\mathrm{B}) = \begin{array}{c} \\ 1 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{ccccc} 2 & 3 & 4 & 7 & 8 \end{array} \\ \begin{pmatrix} 1/3 & 1/3 & 1/3 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 \end{pmatrix} \end{array} \tag{6}$$

The resulting normalized capacity is $c_{\mathrm{TI}}(\mathbf{B}) = 3$ ($= 3$ Mbit/s).

Eq. (4) is valid in the ideal case where both interference from co-located piconets and switching overhead, caused by BGs that change piconet, are neglected. In order to include these two effects, Eq. (4) can be rewritten as

$$\bar{c}_{\mathrm{TI}}(\mathbf{B}) = (c_{\mathrm{TI}}(\mathbf{B}) - \Delta(\mathbf{B})) \cdot I(\mathbf{B}), \tag{7}$$

where $\Delta(\mathbf{B})$ represents a loss of capacity due to the switching overhead and $I(\mathbf{B})$ is a decreasing factor that accounts for interference from co-located piconets.

5.2. TD metrics: scatternet with maximum residual capacity or minimum average load

We consider two traffic dependent metrics: (i) the so-called *residual capacity* (i.e., the capacity that remains available in a scatternet, after that all pre-defined traffic patterns are accommodated); (ii) the *nodes' average load*.

The evaluation of the above metrics is TD, and as such, is dependent on the adopted routing strategy too. As an example, given a traffic pattern, for instance a data flow between device h and device k (with $1 \leqslant h, k \leqslant N$), the capacity that such flow requires from the overall scatternet depends on the number of hops that make up the path

between device h and device k. In our analysis, we assume without loss of generality, that a shortest path routing algorithm is adopted.

To evaluate the metrics, we start by describing the traffic relationships with an $N \times N$ traffic matrix $\mathbf{T} = [t_{hk}]$, whose element t_{hk} represents the capacity, normalized with respect to the piconet's capacity, required by the (h,k) relationship, $(1 \leqslant h,k \leqslant N)$. We assume that the t_{hk} (normalized traffic rates) are fixed for each source–destination couple. We also denote by R the number of traffic relationships expressed by this matrix.

It is easy to see that the capacity required on each link by the traffic relationship between node h and node k is given by $t_{hk}\mathbf{P}^{h,k}(\mathbf{B})$.

The matrix of the overall normalized capacity required for each link to transport the traffic patterns expressed by $\mathbf{T}$ is given by

$$\mathbf{C}_{\mathrm{TD}}(\mathbf{B}) = \sum_{h=1}^{N}\sum_{k=1,k\neq h}^{N} t_{hk}\cdot \mathbf{P}^{h,k}(\mathbf{B}) = \left[d_{ij} = \sum_{h=1}^{N}\sum_{k=1,k\neq h}^{N} t_{hk}\cdot p_{ij}^{h,k}\right]. \tag{8}$$

It is to be noted that the traffic relationships defined by the matrix $\mathbf{T}$ can effectively be supported by the scatternet represented by $\mathbf{B}$ if the following conditions, that assure the steady state, are verified:

masters are not over-loaded

$$\Rightarrow \sum_{j=1}^{S} d_{ij} \leqslant 1 \quad \forall i \in M, \tag{9}$$

slaves are not over-loaded

$$\Rightarrow \sum_{i=1}^{M} d_{ij} \leqslant 1 \quad \forall j \in S. \tag{10}$$

The total capacity required by the traffic relationships of matrix $\mathbf{T}$ is

$$c_{\mathrm{TD}}(\mathbf{B}) = \sum_{i=1}^{M}\sum_{j=1}^{S} d_{ij}. \tag{11}$$

Based on the above definitions, we can finally introduce two TD metrics. The first measures the capacity that remains available in a scatternet, after all traffic patterns are accommodated. Recalling that the capacity of each link is assigned according to Eq. (3), the *residual capacity* matrix, is given by

$$\mathbf{C}_{\mathrm{res,TD}}(\mathbf{B}) = \mathbf{C}_{\mathrm{TI}}(\mathbf{B}) - \mathbf{C}_{\mathrm{TD}}(\mathbf{B}) = \lfloor f_{ij} \rfloor. \tag{12}$$

The information contained in this matrix can be summarized in a single parameter, the *residual capacity*, $c_{\mathrm{res,TD}}(\mathbf{B})$, given by

$$c_{\mathrm{res,TD}}(\mathbf{B}) = \sum_{i=1}^{M}\sum_{j=1}^{S} f_{ij}. \tag{13}$$

Also in this case considerations about decreasing factors due to interference and switching overhead can be applied.

According to this metric, a scatternet is optimal, when the value of $c_{\mathrm{res,TD}}(\mathbf{B})$ is maximized.

Alternatively, we can adopt as metric the *nodes' average normalized load.*

In accordance to Eq. (8) the normalized load on a slave j is $l_j^S = \sum_{i=1}^{M} d_{ij}$, while the normalized load on a master is $l_i^M = \sum_{j=1}^{S} d_{ij}$.

The normalized load averaged over the N nodes, which we denote as $l(\mathbf{B})$, is

$$l(\mathbf{B}) = \frac{\sum_{j=1}^{S} l_j^S + \sum_{i=1}^{M} l_i^M}{N} = \frac{2\cdot\sum_{i=1}^{M}\sum_{j=1}^{S} d_{ij}}{N} = \frac{2\cdot c_{\mathrm{TD}}(\mathbf{B})}{N}. \tag{14}$$

With this metric, the target is the minimization of $l(\mathbf{B})$; we point out that the minimization of the average load goes in the direction of a minimization of the average energy consumption of the scatternet.

5.3. Metrics associated to the path length

As will be shown in Section 6 path lengths have a considerable impact on scatternet performance. As a consequence, in this section, we define three metrics that do take into account path lengths. The first two are not dependent on the traffic loading the scatternet and are defined as follows.

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

$$q^{h,k}(\mathbf{B}) = \sum_{i=1}^{M}\sum_{j=1}^{S} p_{ij}^{h,k}. \tag{15}$$

The first metric that we introduce is the *average path length*, which is the path length averaged over all possible source–destination couples, and is given by

$$q_{\mathrm{TI}}(\mathbf{B}) = \sum_{h=1}^{N} \sum_{k=1,k\neq h}^{N} \frac{q^{h,k}(\mathbf{B})}{N\cdot(N-1)}, \tag{16}$$

the optimization target of this metric is the minimization of $q_{\mathrm{TI}}(\mathbf{B})$.

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

$$a_{\mathrm{TI}}(\mathbf{B}) = \frac{c_{\mathrm{TI}}(\mathbf{B})}{q_{\mathrm{TI}}(\mathbf{B})\cdot N\cdot(N-1)}. \tag{17}$$

We will call this last metric *average path capacity*.

The last metric we introduce depends on traffic and thus it depends on the matrix $\mathbf{T}$, which defines R traffic relationships. The metric is the *path length averaged over the R traffic relationships*, instead that over all possible $N\cdot(N-1)$ relationships, as done in Eq. (15); it is given by

$$q_{\mathrm{TD}}(\mathbf{B}) = \sum_{(h,k)\in T, h\neq k} \frac{q^{h,k}(\mathbf{B})}{R}. \tag{18}$$

The associated target consists in minimizing this path length; this metric goes in the direction of minimizing the transfer delay related to the number of hops.

Table 1 summarizes the metrics introduced and briefly reports the significance of the metric themselves.

Our analysis does not aim at identifying all the possible metrics, however by following this methodology many other metrics can be identified, as metrics related to the behavior of specific nodes/links in the network (i.e., minimize the energy consumption of the bottleneck node, maximize the minimal residual capacity among the links).

The six metrics in Table 1 have been introduced in order to measure some significant performance parameters of the overall scatternet. In particular, while within TI metrics $c_{\mathrm{TI}}(\mathbf{B})$ simply measures the potential of the scatternet in terms of capacity, the other two are introduced to evaluate the scatternet by considering a generic source–destination couple obtained by averaging over all possible ones and by assuming uniform traffic patterns.

As for the TD metrics, the residual capacity has been introduced to evaluate how the scatternet is able, in terms of capacity, to sustain adjunctive flows. The other two metrics, that are related to the path lengths, are in favor of scatternets that efficiently support the given traffic matrix $\mathbf{T}$. If traffic patterns are balanced (i.e., all the traffic relationship require the same capacity) the optimization of the two metrics $q_{\mathrm{TD}}(\mathbf{B})$ and $l_{\mathrm{TD}}(\mathbf{B})$ gives rise to the same optimal scatternet $\mathbf{B}^*$.

6. Numerical results

In this Section, we present numerical results relevant to the metrics defined above obtained in accordance to Section 4.2. Each of the following figures represents the area containing the scatter-

Table 1
A summary of the defined metrics

Traffic independent	Normalized capacity	$c_{\mathrm{TI}}(\mathbf{B})$	Capacity of the overall scatternet normalized with respect to the capacity of a piconet
	Average path length	$q_{\mathrm{TI}}(\mathbf{B})$	Average number of hops of shortest paths between nodes
	Average path capacity	$a_{\mathrm{TI}}(\mathbf{B})$	Capacity available for the generic source–destination couple
Traffic dependent	Residual capacity	$c_{\mathrm{res,TD}}(\mathbf{B})$	Residual capacity of the scatternet once traffic in $\mathbf{T}$ has been allocated
	Average path length	$q_{\mathrm{TD}}(\mathbf{B})$	Average number of hops of shortest paths between nodes for patterns in $\mathbf{T}$
	Average normalized load	$l_{\mathrm{TD}}(\mathbf{B})$	Average load on nodes in the scatternet to support patterns in $\mathbf{T}$

net; x and y axes are measured in meters. The TR is assumed to be equal to 10 m. Nodes are numbered according to the order they are generated in the communication scenario. Figures report nodes, their roles (master, slave or bridge) and radio links connecting them. Since number of possible scatternets quickly becomes overwhelming with respects to the number of nodes, in this section we analyze networks with a limited number of devices.

6.1. Traffic independent metrics

This subsection shows examples of scatternets resulting from TI optimization. The scenario consists of 12 devices not all the nodes are in radio visibility. The scatternet in Fig. 2 is obtained by selecting the one with maximum *normalized capacity*. The interference effect in Eq. (7) is evaluated by relying on the model presented in [22]. The switching overhead, which depends on the adopted interpiconet scheduling policy [2], has been considered here with simple assumptions on the switching frequency.

It can be immediately noticed that the scatternet presents a linear structure, i.e., every node is connected with two other nodes only. The value assumed by $c_{\mathrm{TI}}(\mathbf{B})$, evaluated as in Eq. (4) is 5.5. When switching overhead is considered $c_{\mathrm{TI}}(\mathbf{B})$ decreases to 5.2727 and by considering also the interference effect it becomes 4.7151. Although this scatternet is the one with maximum *normalized capacity*, it has characteristics that make its topology undesirable: it presents large values of the average path length that could lead to high transfer delays, and, since for Class 3 devices no power control mechanisms have been defined, increasing the number of hops per traffic relationship does not bring any benefit in terms of power consumption and consumes capacity as function of the involved links. As an example, a single 500 kbit/s bi-directional flow between node 10 and node 12 in Fig. 2 would use all scatternet capacity: nodes along the path would spend half their time receiving traffic from one of the two directions and the remaining time relaying traffic in the opposite direction.

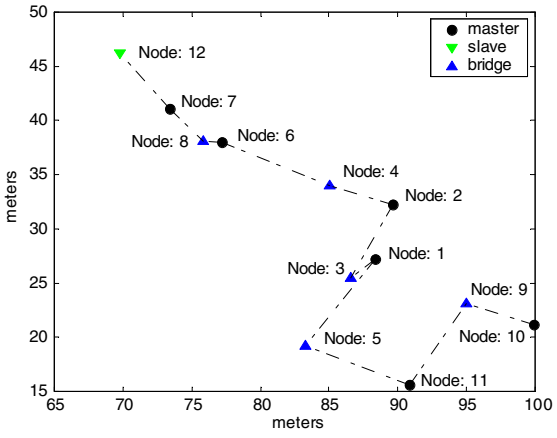

Fig. 2. Scatternet with maximum normalized capacity.

The peculiar structure produced by this metric is due to the following reasons:

- The metric tends to favor scatternets formed by a large number of piconets, since each new piconet increases the overall capacity with its contribution.
- The interference effect is not significant since the number of co-located piconets is low (<50).
- When the switching overhead effect is taken into account, a BG loses capacity as a function of the number of piconets it is connected to. Thus, high performance, in terms of capacity, is attained when a bridge node is connected to only two piconets.

These considerations explain why path lengths have to be taken into account. However, minimizing the path lengths without considering capacity at the same time, could lead to undesirable scatternet topologies too since, if the nodes are distributed in a small area, the resulting scatternet presents a fully meshed topology where every slave is connected to every master. In this case, the resulting capacity is low because of the high number of BGs connected to a high number of piconets. Fig. 3 refers to the metric that minimizes the average path length in the same scenario of Fig. 2. In this case the nodes in the lower part of the figure (which are in radio visibility of each other) are all connected.

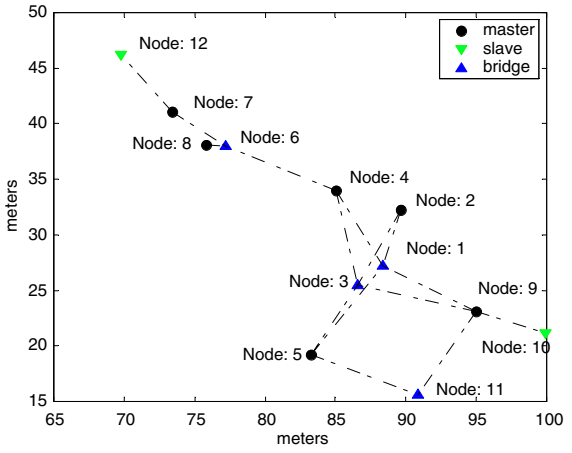

Fig. 3. Scatternet with minimum average path length.

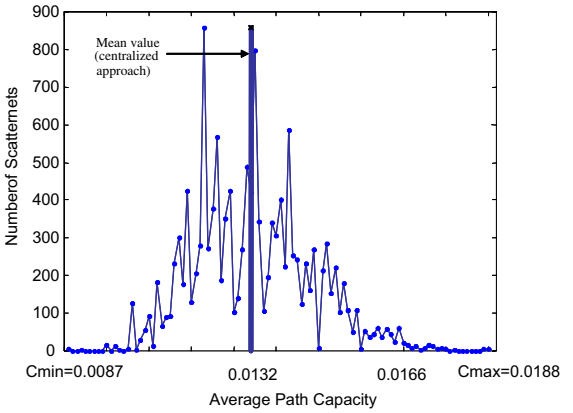

Fig. 5. Distribution of the values of average path capacity.

Let us now look at Fig. 4, which shows a scatternet obtained with a metric that maximizes the *average path capacity*. This metric seems to be the most suited to maximize network performance, since both capacity and path length are taken into account. The scatternet of Fig. 4 presents a capacity, $c_{\mathrm{TI}}(\mathbf{B})$ equal to 4.667 (4.5271 taking into account the switching overhead and 4.0483 taking into account also the interference effect). The overall capacity is smaller than the value obtained by maximizing the *normalized capacity*, but, while in that case the average path length was 4.33, and the resulting capacity available for a generic source–destination pair was 0.0082, in the case of Fig. 4 the average path length is 2.81 and the resulting capacity per source–destination pair is 0.0109.

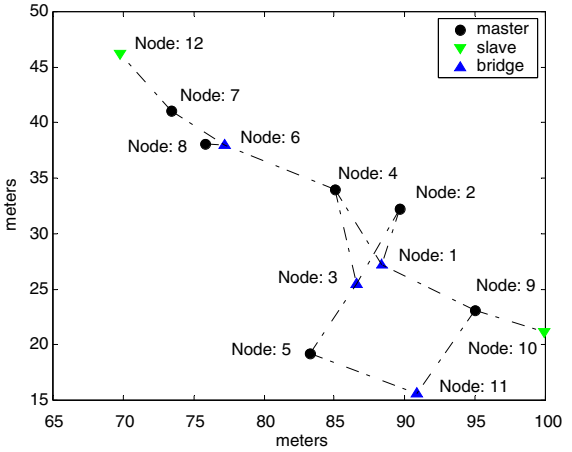

Fig. 4. Scatternet with maximum average path capacity.

As a result, we select as suitable TI metric the average path capacity. In order to better analyze the space of feasible scatternets Fig. 5 shows the histogram of the *average path capacity* of *all* possible "BT-compliant" scatternets obtained in a scenario constituted by 10 nodes distributed in an area of 25×25 m but, as will be shown later, a similar distribution holds in general. As indicated in Section 4.2 the number of different feasible topology is huge. The values of $a_{\mathrm{TI}}(\mathbf{B})$ are distributed in a range starting form $a_{\mathrm{TI,min}}(\mathbf{B}) = 0.0087$ ($\approx$8 kbit/s for every possible node pair) to $a_{\mathrm{TI,max}}(\mathbf{B}) = 0.0188$ ($\approx$19 kbit/s per pair); the mean value of $a_{\mathrm{TI}}(\mathbf{B})$ is also shown (equal to 0.0132). Note that the mean value is quite distant from the maximum value, which corresponds to the one associated with the optimal scatternet. Moreover, a few scatternets have a high value of $a_{\mathrm{TI}}(\mathbf{B})$ and are thus contained in the right tail of the histogram. This is an interesting result because it indicates that topology optimization is going to be 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

Table 2
Traffic relationships

Traffic relationship	Required normalized capacity	Traffic relationship	Required normalized capacity
$1 \leftrightarrow 2$	0.08	$5 \leftrightarrow 6$	0.125
$1 \leftrightarrow 5$	0.1	$6 \leftrightarrow 2$	0.125
$3 \leftrightarrow 4$	0.1	$7 \leftrightarrow 8$	0.05
$3 \leftrightarrow 9$	0.1	$10 \leftrightarrow 7$	0.1
$4 \leftrightarrow 9$	0.1	$10 \leftrightarrow 8$	0.05

search for a connected scatternet but also explicitly aim at maximizing its performance.

6.2. Traffic dependent metrics

In this Section, we show results derived by applying the TD metrics. The scenario is composed of 10 nodes; the relevant traffic relationships are shown in Table 2. In Fig. 6 we depict the scatternet with maximum *residual capacity*. It can be noticed that this metric suffers from the same drawbacks of the metric it is derived from (i.e., *normalized capacity*, see Fig. 2): the resulting scatternet is likely to present a linear structure.

In Fig. 7 we show the scatternet presenting the minimum *average path length* per traffic relationship, see Eq. (18). In this case the scatternet presents a more connected structure, with respect to that of Fig. 6.

Finally, in Fig. 8 we minimize the *average normalized load*. The resulting normalized loads are

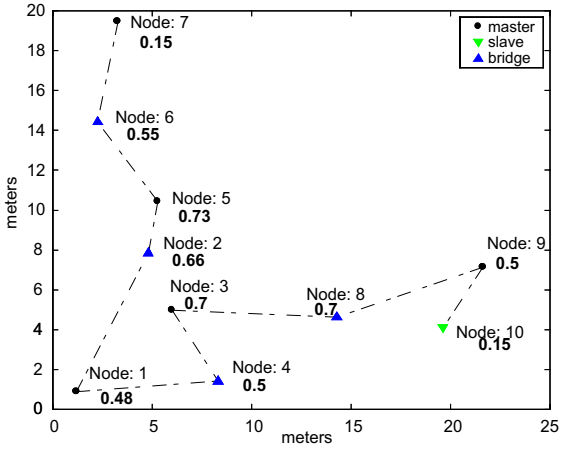

Fig. 6. Scatternet with maximum residual capacity.

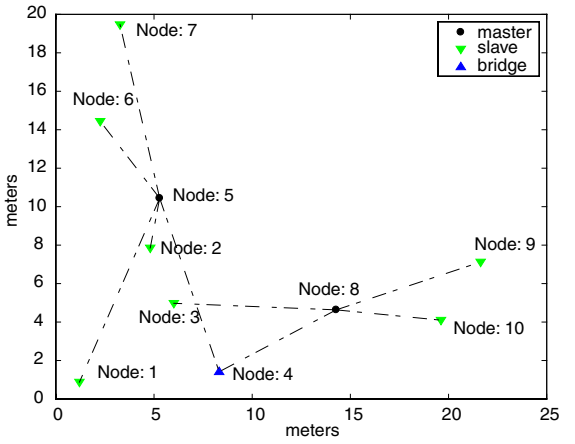

Fig. 7. Scatternet with minimum average path length per traffic relationship.

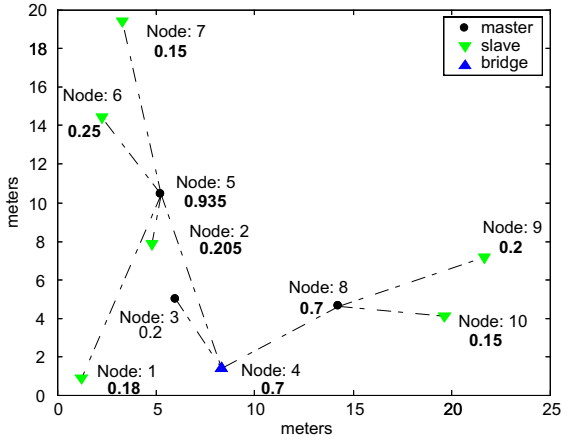

Fig. 8. Scatternet with minimum average load.

reported in the figure, next to each node. The scatternet average load is 0.3058. It can be noticed that nodes that are located in specific positions

have to take care of all the forwarding load (i.e., traffic handled on behalf of other nodes). Nevertheless, since reducing load means reducing energy consumption, this scatternet consumes 70% of the energy required by the scatternet of Fig. 6 (the average load for the latter scatternet is 0.4267).

7. 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 8, where we will describe a two-phases distributed scatternet formation algorithm based on DSOA.

7.1. 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 $1 \ldots k-1$.

```
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 = 1 : N
    w_k = random selection of a node v
    from V such that:
      (1) v ∉ W
      (2) ∃ u ∈ W such that distance
          (u,v) ⩽ TR
    W = W ∪ {w_k}
  endfor
end
```

Since with DSOA the nodes sequentially select how to connect, each node must be in TR of at least another already entered node. 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 $W = \{w_1, w_2, \ldots, w_{k-1}\}$ and no couple (v,w) with $v \in V \setminus W$, $w \in W$ exists such that $\mathrm{dist}(v,w) < TR$. Therefore, since $W \subseteq V$, there exist two disconnected components, namely 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 $(i,j \in 1,\ldots,N)$. This part of the algorithm is 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.

```
SCATTERNET_OPTIMIZATION_ALGORITHM
(SOA)
Input: W, G(V,E), M
Output: locally optimal scatternet
B*
begin
V_M = Ø
V_S = Ø
V_M = V_M ∪ w_1
V_S = V_S ∪ w_2
B^2 = [1]
  for k = 3 : N
    case 1) consider w_k in V_M
      * derive all BT-compliant ma-
      trices B^k with |V_M| + 1 rows and
      |V_S| columns
      calculate values of M(B^k)
    case 2) consider w_i in V_S
      * derive all BT-compliant ma-
      trices B^k with |V_M| rows and
```

```
      |V_S| + 1 columns
      calculate values of M(B^k)
    select the B^k with optimal M(B^k)
    if optimum in case 1)
    then
      V_M = V_M ∪ w_k
    else
      if optimum in case 2)
        V_S = V_S ∪ w_k
      else
        RECONFIGURE(B^{k-1}, V_M, V_S)
      endif
    endif
  endfor
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 $v \in \{w_1, w_2, \ldots, w_{k-1}\}$ exists s.t. $v \in V_S$ and distance $(w_k, v) \leqslant \mathrm{TR}$, i.e. one of the slave nodes already in the network is in transmission range of w_k. If no slaves are in transmission range of w_k, whenever a node $v \in V_M$ exists, with distance $(w_k, v) \leqslant \mathrm{TR}$, and slaves $(v) <= 7$, w_k can be a slave of v. Otherwise, at least one node $w_i \in V_M$ must exist, with distance $(w_k, v) \leqslant \mathrm{TR}$, and slaves $(v) = 7$, with $i \leqslant k$. 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 has to be disconnected. However, it was proven in [10] 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 seventh 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 sixth, 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, the results obtained with DSOA are close to the optimum.

7.2. Examples and numerical results

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.

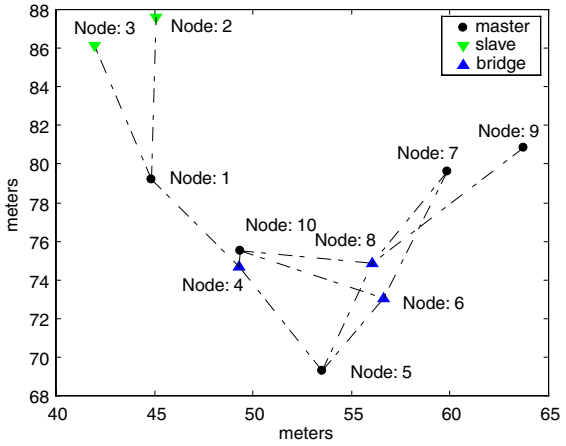

Fig. 9. Scatternet formed with DSOA.

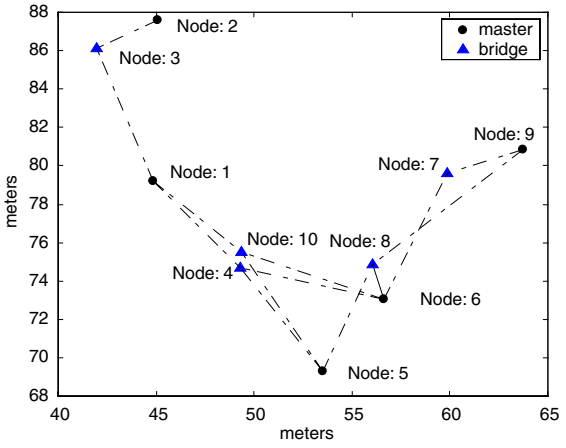

Fig. 10. Optimal scatternet.

An example is shown in Figs. 9 and 10, where 10 nodes are distributed in an area of 25×25 m (the same scenario of Fig. 5). The first figure reports the scatternet formed with the DSOA (with an $a_{\mathrm{TI}}(\mathbf{B}) = 0.0145$). Fig. 10 depicts the optimal scatternet (that presents an $a_{\mathrm{TI}}(\mathbf{B}) = 0.0188$). Fig. 11 shows a comparison between the optimal $a_{\mathrm{TI}}(\mathbf{B})$ and the one obtained with the DSOA. The dotted curve shows the histogram of the *average path capacity* of *all* possible "BT-compliant" scatternets feasible in this scenario. The values of $a_{\mathrm{TI}}(\mathbf{B})$ are distributed in a range starting form $a_{\mathrm{TI,min}}(\mathbf{B}) = 0.0087$ (≈8 kbit/s for every possible node pair) to $a_{\mathrm{TI,max}}(\mathbf{B}) = 0.0188$ (≈19 kbit/s per pair); the mean

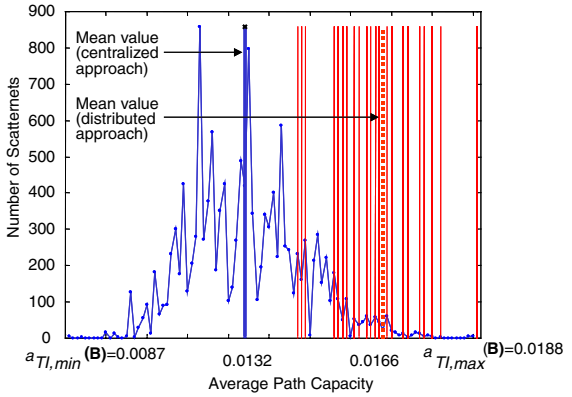

Fig. 11. Comparison between results obtained with DSOA and results derived from scatternet space enumeration.

value of $a_{\mathrm{TI}}(\mathbf{B})$ is also shown (equal to 0.0132). Note that the mean value is quite distant from the maximum value, which corresponds to the one associated with the optimal scatternet. Moreover, a few scatternets have a high value of $a_{\mathrm{TI}}(\mathbf{B})$ and thus are contained in the right tail of the histogram. This is an interesting results because it suggests that topology optimization is a fundamental issue for Bluetooth scatternets: in fact, this distribution for 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 explicitly aim at maximizing performance.

As regards the DSOA, the vertical lines in Fig. 11 correspond to the values of $a_{\mathrm{TI}}(\mathbf{B})$ 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 $a_{\mathrm{TI}}(\mathbf{B})$ greater than the overall mean value of all possible scatternets). The mean value of $a_{\mathrm{TI}}(\mathbf{B})$ of these 100 DSOA scatternets is equal to 0.0166. The unnormalized values of the average capacity per path 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.

Fig. 12 shows a similar distribution in a scenario with 15 nodes in a multi-hop context. In Fig. 13 a distribution mediated on 100 different scenarios, with varying number of nodes is shown, while Fig. 14 reports the distribution of the values obtained

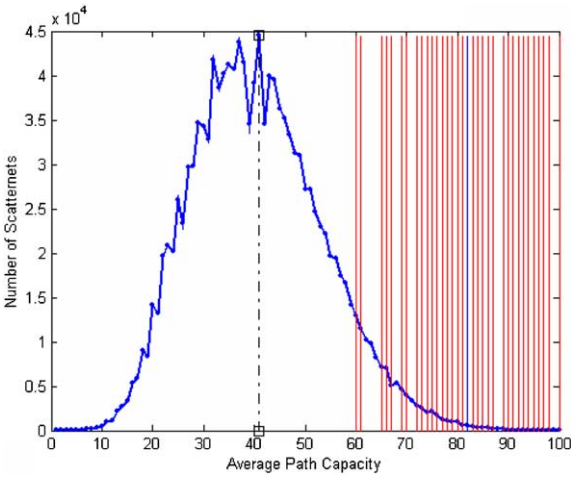

Fig. 12. Distribution of average path capacity for 15 nodes.

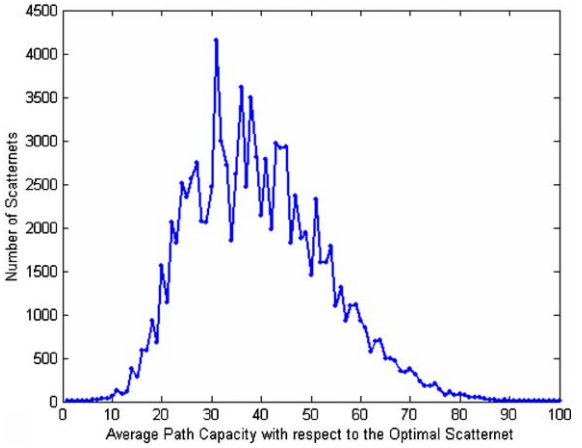

Fig. 13. Distribution of average path capacity on different scenarios.

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.

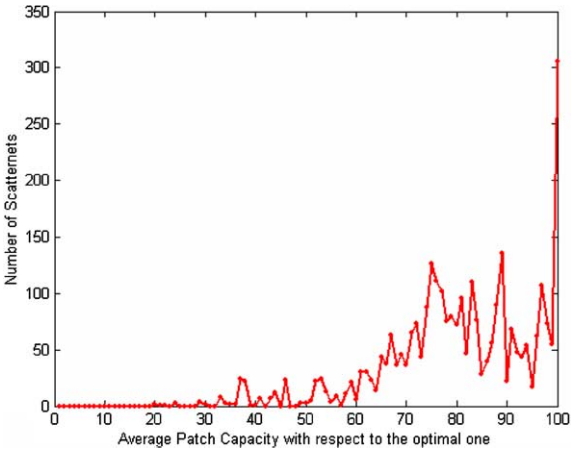

Fig. 14. Distribution of average path capacity for DSOA scatternets.

8. 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 ($\mathbf{B}^{\mathbf{k-1}}$ 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 [11], 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.

```
PROCEDURE ENTER (B^{k-1}, k) B^k = DSOA(B^{k-1})
foreach v ∈ children
   send (execute_enter (B^k, k+1), v)
   wait_answer()
   [B^{k+c}, c] = answer(v)
   k = k + c
endforeach
send (branch_entered (B^k, k), parent)
```

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 it the `execute_enter`($\mathbf{B}^k$, $k+1$) message. 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`) 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 descendents with the `branch_entered` message. When the *root* node receives the answer from its last son, all nodes have taken their decision. Fig. 15 shows how a simple tree, composed of 13 nodes, can be visited in 25 steps to execute DSOA. The numbers inside the circles represent the order in DSOA, i.e., the order in which nodes enter the network. The numbers outside the circles represent the "path" followed by the **B** matrix in the visit procedure.

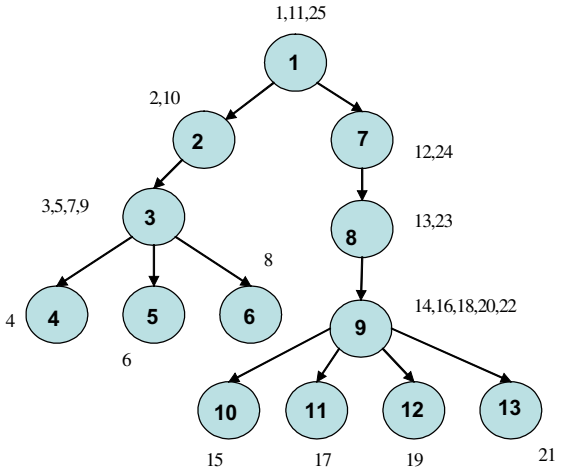

Fig. 15. Visit procedure on the tree.

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 broadcast the matrix down to its children in the tree, it enters the *reconfiguration* phase. During reconfiguration it 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* state 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 dead-

locks. 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 in 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, we showed in [11] 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 with the Blueware [23] simulator.

9. Conclusions

In this paper, we discussed the scatternet formation issue in Bluetooth, by setting a framework for scatternet analysis based on a representation in a matrix form, which allows developing and applying different metrics. We identified several metrics both in a traffic independent and in a traffic dependent context, and we showed the relevant numerical results. The analysis of these results allows selecting the most suitable metric for a given scenario.

A distributed algorithm for scatternet topology optimization, DSOA, was then described. The performance of DSOA has been evaluated and is 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.C. Haartsen, The Bluetooth radio system, IEEE Personal Communications 7 (1) (2000) 28–36.
[2] P. Johansson, M. Kazantzidis, R. Kapoor, M. Gerla, Bluetooth—an enabler for personal area networking, IEEE Network 15 (5) (2001) 28–37.
[3] Bluetooth SIG Specification of the Bluetooth System, Version 1.1—Core, February 2001.
[4] IEEE Std 802.15.1™-2002, Available from <http://www.ieee802.org/15/pub/TG1.html>.
[5] S. Zurbes, Considerations on Link and System Throughput of Bluetooth Networks, 11th IEEE International Symposium on Personal Indoor and Mobile Radio Communications, vol. 2, 2000, pp. 1315–1319.
[6] T. Salonidis, P. Bhagwat, L. Tassiulas, R. LaMaire. Distributed Topology Construction of Bluetooth Personal Area Networks, IEEE INFOCOM'01, 2001, pp. 1577–1586.
[7] C. Law, A. Mehta, K. Siu, Performance of a new Bluetooth formation protocol, ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), October 2001.
[8] H. Zhang, J.C. Hou, L. Sha, A Bluetooth loop scatternet formation algorithm, Proceedings of the IEEE ICC 2003, Anchorage, 2003, pp. 1174–1180.
[9] G. Tan, A. Miu, J. Guttag, H. Balakrishnan, An efficient scatternet formation algorithm for dynamic environments, IASTED Communications and Computer Networks (CCN), Cambridge, MA, November 2002.
[10] G.V. Zaruba, S. Basagni, I. Chlamtac, Bluetrees-scatternet formation to enable Bluetooth-based ad hoc networks, IEEE International Conference on Communications (ICC'01), vol. 1, 2001, pp. 273–277.
[11] F. Cuomo, G. Di Bacco, T. Melodia, SHAPER: A self-healing algorithm producing multi-hop Bluetooth scattERnets, Proceedings of the IEEE Globecom 2003, San Francisco, USA, in press.
[12] Z. Wang, R. Thomas, Z. Haas, bluenet—a new scatternet formation scheme, Proceedings of the Hawaii International Conference on System Science (HICSS-35), 2002.
[13] C. Petrioli, S. Basagni, I. Chlamtac, Configuring bluestars: multihop scatternet formation for Bluetooth networks, IEEE Transactions on Computers (Special Issue on Wireless Internet) 52 (6) (2003) 779–790.
[14] C. Petrioli, S. Basagni, I. Chlamtac, BlueMesh: degree-constrained multihop scatternet formation for Bluetooth networks, Mobile Networks and Applications (Special Issue on Advances in Research of Wireless Personal Area Networking and Bluetooth Enabled Networks), 9 (1) (2004) 33–47.
[15] I. Stojmenovic, Dominating set based scatternet formation with localized maintenance, Proceedings of the Workshop on Advances in Parallel and Distributed Computational Models, April 2002.

[16] 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 21 (2) (2003) 229–239.
[17] M. Ajmone Marsan, C.F. Chiasserini, A. Nucci, G. Carrello, L. De Giovanni, Optimizing the topology of Bluetooth Wireless Personal Area Networks, IEEE INFOCOM'02.
[18] R. Kapoor, M.Y.M. Sanadidi, M. Gerla, An analysis of Bluetooth scatternet topologies, Proceedings of the ICC 2003, 2003.
[19] P. Bhagwat, S.P. Rao, On the characterization of Bluetooth scatternet topologies, Available from <http://win-www.rutgers.edu/~pravin/bluetooth/>.
[20] F. Cuomo, T. Melodia, A general methodology and key metrics for scatternet formation in Bluetooth, IEEE Globecom 2002, November 2002.
[21] C.F. Chiasserini, M. Ajmone Marsan, E. Baralis, P. Garza, Towards feasible distributed topology formation algorithms for Bluetooth-based WPANs, Hawaii International Conference on System Science (HICSS-36), Big Island, Hawaii, 6 January 2003.
[22] E.-H. Amre, Interference between Bluetooth networks-upper bound on the packet error rate, IEEE Communications Letters 5 (2001) 245–247.
[23] G. Tan, Blueware: Bluetooth simulator for ns, MIT Laboratory for Computer Science, Cambridge, October 2002.

Tommaso Melodia received his "laurea" degree in Telecommunications Engineering from the University of Rome "La Sapienza" in 2001. He is currently a Ph.D. student in Information and Communication Engineering at the same University. From February to August 2003 he was a Visiting Researcher at the Broadband and Wireless Networking Laboratory at the Georgia Institute of Technology. His main research interests are related to Computer Networks, Wireless ad hoc Networks, Wireless Sensor Networks, Personal and Mobile Communications.

Francesca Cuomo received her "Laurea" degree in Electrical Engineering in 1993, magna cum laude, from the University of Rome "La Sapienza", Italy. She earned the Ph.D. degree in Information and Communication Engineering in 1998, also from the University of Rome "La Sapienza". Since 1996 she is a "Researcher" (Assistant Professor) at the INFOCOM Department of this University. She teaches courses in Telecommunication Networks.

Her main research interests focus on: Modeling and Control of broadband integrated networks, Signaling and Intelligent Networks, Architectures and protocol for fixed an mobile wireless networks, Mobile and Personal Communications, Quality of Service guarantees and real time service support in the Internet and in the radio access, Reconfigurable radio systems and Wireless ad hoc networks. She participated in: (I) the European ACTS INSIGNIA project dedicated to the definition of an Integrated IN and B-ISDN network (1995–1998); (II) RAMON project, funded by the Italian Public Education Ministry, focused on the definition of a reconfigurable access module for mobile computing applications (2000–2002); (III) National project "5% Multimedialità" CNR-MURST. She is now participating to the European IST WHYLESS.COM project focusing on adoption of the Ultra Wide Band radio technology for the definition of an Open Mobile Access Network (2000–2003). In this project she is leader of the WP4 (Network Resource Manager). As for current national projects: (I) she is involved in FIRB project VIRTUAL IMMERSIVE COMMUNICATIONS (VICOM) where she is responsible of the research activities on the BAN and VAN networks; (II) she is responsible of the research unit at the University of Rome "La Sapienza" in the EURO project funded by the Italian Public Education Ministry.

In 1995 she joined Coritel, a research institute on telecommunications, and she has been responsible for two years of the SWAP project in the Radio Access area.