Capacity Assignment in Bluetooth Scatternets – Analysis and Algorithms

Gil Zussman and Adrian Segall

Department of Electrical Engineering
Technion – Israel Institute of Technology, Haifa 32000, Israel
{gilz@tx, segall@ee}.technion.ac.il
http://www.comnet.technion.ac.il/segall

Abstract. Bluetooth enables portable electronic devices to communicate wirelessly via short-range ad-hoc networks. Initially Bluetooth will be used as a replacement for point-to-(multi)point cables. However, in due course, there will be a need for forming multihop ad-hoc networks over Bluetooth, referred to as scatternets. This paper investigates the capacity assignment problem in Bluetooth scatternets. The problem arises primarily from the special characteristics of the network and its solution requires new protocols. We formulate it as a problem of minimizing a convex function over a polytope contained in the matching polytope. Then, we develop an optimal algorithm which is similar to the well-known flow deviation algorithm and that calls for solving a maximum-weight matching problem at each iteration. Finally, a heuristic algorithm with a relatively low complexity is developed.

Keywords: Bluetooth, Scatternet, Capacity assignment, Capacity allocation, Scheduling, Personal Area Networks (PAN)

1 Introduction

Recently, much attention has been given to the research and development of Personal Area Networks (PAN). These networks are comprised of personal devices, such as cellular phones, PDAs and laptops, in close proximity to each other. Bluetooth is an emerging PAN technology which enables portable devices to connect and communicate wirelessly via short-range ad-hoc networks [5],[6],[11]. Since its announcement in 1998, the Bluetooth technology has attracted a vast amount of research. However, the issue of capacity assignment in Bluetooth networks has been rarely investigated. Moreover, most of the research regarding network protocols has been done via simulation. In this paper we formulate an analytical model for the analysis of the capacity assignment problem and propose optimal and heuristic algorithms for its solution.

Bluetooth utilizes a short-range radio link. Since the radio link is based on frequency-hop spread spectrum, multiple channels (frequency hopping sequences) can co-exist in the same wide band without interfering with each other. Two or more units sharing the same channel form a *piconet*, where one unit acts as a *master* controlling the communication in the piconet and the others act as *slaves*.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 411-422, 2002.

Bluetooth channels use a frequency-hop/time-division-duplex (FH/TDD) scheme. The channel is divided into 625-μsec intervals called *slots*. The master-to-slave transmission starts in even-numbered slots, while the slave-to-master transmission starts in odd-numbered slots. Masters and slaves are allowed to send 1,3 or 5 slots *packets* which are transmitted in consecutive slots. A slave is allowed to start transmission in a given slot if the master has addressed it in the preceding slot. Information can only be exchanged between a master and a slave, i.e. there is no direct communication between slaves. Although packets can carry synchronous information (voice link) or asynchronous information (data link), in this paper we concentrate on networks in which only data links are used.

Multiple piconets in the same area form a *scatternet*. Since Bluetooth uses packet-based communications over slotted links, it is possible to interconnect different piconets in the same scatternet. Hence, a unit can participate in different piconets, on a time-sharing basis, and even change its role when moving from one piconet to another. We will refer to such a unit as a *bridge*. For example, a bridge can be a master in one piconet and a slave in another piconet. However, a unit cannot be a master in more than one piconet.

Initially Bluetooth piconets will be used as a replacement for point-to-(multi)point cables. However, in due course, there will be a need for multihop ad-hoc networks (scatternets). Due to the special characteristics of such networks, many theoretical and practical questions regarding the scatternet performance are raised. Nevertheless, only a few aspects of the scatternet performance have been studied. Two issues that received relatively much attention are: research regarding scatternet topology and development of efficient scatternet formation protocols (e.g. [4],[13]).

Much attention has also been given to scheduling algorithms for piconets and scatternets. In the Bluetooth specifications [5], the capacity allocation by the master to each link in its piconet is left open. The master schedules the traffic within a *piconet* by means of polling and determines how bandwidth capacity is to be distributed among the slaves. Numerous heuristic scheduling algorithms for piconets have been proposed and evaluated via simulation (e.g. [7],[8]). In [11] an overall architecture for handling scheduling in a *scatternet* has been presented and a family of inter-piconet scheduling algorithms (algorithms for masters and bridges) has been introduced. Inter-piconet scheduling algorithms have also been proposed in [1] and [16].

Although scatternet formation as well as piconet and scatternet scheduling have been studied, the issue of *capacity assignment* in Bluetooth scatternets has not been investigated. Moreover, Baatz et al. [1] who made an attempt to deal with it have indicated that it is a complex issue.[1] Capacity assignment in communication networks focuses on finding the best possible set of link capacities that satisfies the traffic requirements while minimizing some performance measure (such as average delay). We envision that in the future, capacity assignment protocols will start operating once the scatternet is formed and will determine link capacities that will be dynamically allocated by scheduling protocols. Thus, *capacity assignment protocols* are the

---

[1] In [1] the term *piconet presence schedule* is used to refer to a notion similar to *capacity assignment*.

missing link between scatternet formation and scatternet scheduling protocols. A correct use of such protocols will improve the utilization of the scatternet bandwidth. We also anticipate that the optimal solution of the *capacity assignment problem* will improve the evaluation of heuristic scatternet scheduling algorithms.

Most models of capacity assignment in communication networks deal mainly with static networks in which a cost is associated with each level of link capacity (see [3] for a review of models). The following discussion shows that there is a need to study the capacity assignment problem in Bluetooth scatternets in a different manner:

- In contrast with a wired and static network, in an ad-hoc network, there is no central authority responsible for network optimization and there is no monetary cost associated with each level of link capacity.
- The nature of the network allows frequent changes in the topology and requires frequent changes in the capacities assigned to every link.
- There are constraints imposed by the tight master-slave coupling and by the time-division-duplex (TDD) scheme.
- Unlike other ad-hoc networks technologies in which all nodes within direct communication from each other share a common channel, in Bluetooth only a subgroup of nodes (piconet) shares a common channel and capacity has to be allocated to each link.

A scatternet capacity assignment protocol has to determine the capacities that each master should allocate in its own piconet, such that the network performance will be optimized. Currently, our major interest is in algorithms for quasi-static capacity assignment that will minimize the average delay in the scatternet. The analysis is based on a static model with stationary flows and unchanging topology. To the best of our knowledge, the work presented in this paper is the first attempt to analytically study the capacity assignment problem in Bluetooth scatternets.

In this paper we focus on formulating the problem and developing centralized algorithms. The development of the distributed protocols is subject of further research.

In the sequel we formulate the scatternet capacity assignment problem as a minimization of a convex function over a polytope contained in the polytope of the well-known *matching problem* [14, p. 608] and show that different formulations apply to bipartite and nonbipartite scatternets. The methodology used by Gerla et al. [9],[15] is used in order to develop an *optimal scatternet capacity assignment algorithm* which is similar to the *flow deviation algorithm* [3, p. 458]. The main difference between the algorithms is that at each iteration there is a need to solve a *maximum-weight matching problem* instead of a *shortest path problem*. Finally, we introduce a *heuristic algorithm* whose complexity is much lower than the complexity of the optimal algorithm and whose performance is often close to that of the optimal algorithm. Due to space constraints, numerical results are not presented in this paper and the proofs are omitted. Yet, numerical examples and the proofs can be found in [18].

This paper is organized as follows. In Section 2, we present the model and in Section 3 we formulate the scatternet capacity assignment problem for bipartite and nonbipartite scatternets. An algorithm for obtaining the optimal solution of the problem is presented in Section 4. In Section 5, we develop a heuristic algorithm for bipartite scatternets and in Section 6 we summarize the main results.

2 Model and Preliminaries

Consider the connected undirected scatternet graph $G = (N,L)$. N will denote the collection of *nodes* $\{1,2,\ldots,n\}$. Each of the nodes could be a master, a slave, or a bridge. The *bi-directional link* connecting nodes i and j will be denoted by (i,j) and the collection of bi-directional links will be denoted by L. For each node i, denote by $Z(i)$ the collection of its neighbors. We denote by $L(U)$ $(U \subseteq N)$ the collection of links connecting nodes in U.

Usually, capacity assignment protocols deal with the allocation of capacity to directional links. However, due to the tight coupling of the uplink and downlink in Bluetooth piconets[2], we concentrate on the total bi-directional link capacity. Hence, we assume that the average packet delay on a link is a function of the total link flow and the total link capacity. An equivalent assumption is that the uplink and the downlink flows are equal (symmetrical flows).

Let F_{ij} be the average bi-directional flow on link (i,j) and let C_{ij} be the capacity of link (i,j) (the units of F and C are bits/second). We assume that at every link the average bi-directional flow is positive ($F_{ij} > 0 \;\; \forall (i,j) \in L$). We define f_{ij} as the ratio between F_{ij} and the maximal possible flow on a Bluetooth link when using a given type of packets[3]. We also define c_{ij} as the ratio between C_{ij} and the maximal possible capacity of a link. It is obvious that $0 < f_{ij} \leq 1$ and that $0 \leq c_{ij} \leq 1$. In the sequel, f_{ij} will be referred to as the *flow on link* (i,j) and c_{ij} will be referred to as the *capacity of link* (i,j). Accordingly, $\bar{c}$ will denote the vector of the link capacities and will be referred to as the *capacity vector*.

The objective of the capacity assignment algorithms, described in this paper, is to minimize the average delay in the scatternet. We define D_{ij} as the total delay per unit time of all traffic passing through link (i,j), namely:

Definition 1. *D_{ij} is the average delay per unit of the traffic multiplied by the amount of traffic per unit time transmitted over link (i,j).*

We assume that D_{ij} is a function of the link capacity c_{ij} only. We should point out that the optimal algorithm requires no explicit knowledge of the function $D_{ij}(c_{ij})$. We shall need to assume only the following reasonable properties of the function $D_{ij}(\cdot)$.

Definition 2. *$D_{ij}(\cdot)$ is defined such that all the following holds*:
1. *D_{ij} is a nonnegative continuous decreasing function of c_{ij} with continuous first and second derivatives.*
2. *D_{ij} is convex.*
3. $\lim_{c_{ij} \to f_{ij}} D_{ij}(c_{ij}) = \infty$
4. $D_{ij}'(c_{ij}) < 0$ *for all c_{ij} where D_{ij}' is the derivative of D_{ij}.*

---

2 A slave is allowed to start transmission only after a master addressed it in the preceding slot.

3 For example, currently the maximal flow on a symmetrical link, when using five slots unprotected data packets (DH5), is 867.8 Kbits/second.

Using Definition 1, we shall now define the total delay in the network.

Definition 3. *The* total delay in the network per unit time *is denoted by* D_T *and is given by*:

$$D_T = \sum_{(i,j)\in L} D_{ij}(c_{ij})$$

Since the total traffic in the network is independent of the capacity assignment procedure, we can minimize the average delay in the network by minimizing D_T. A capacity vector that achieves the minimal average delay will be denoted by $\overline{c}^*$.

A *capacity assignment algorithm* has to determine what portion of the slots should be allocated to each master-slave link. On the other hand, a *scheduling algorithm* has to determine which master-slave links should use any given slot pair. Hence, we define a scheduling algorithm as follows.

Definition 4. *A* Scheduling Algorithm *determines how each slot pair is allocated. It does not allow transmission on two adjacent links in the same slot pair.*

The Bluetooth specifications [5] do not require that different masters' clocks will be synchronized. Since the clocks are not synchronized a guard time is needed in the process of moving a bridge from one piconet to another. Yet, in order to formulate a simple analytical model we assume that the *guard times are negligible*. This assumption allows us to consider a scheduling algorithm for the whole scatternet.

3 Formulation of the Problem

Scatternet graphs can be *bipartite graphs* or *nonbipartite graphs* [4] (a graph is called bipartite if there is a partition of the nodes into two disjoint sets *S* and *T* such that each edge joins a node in *S* to a node in *T* [14, p. 50]). Any scatternet graph in which no master is allowed to be a bridge is necessarily bipartite. For example, the scatternet graph described in Fig. 1-A is bipartite. Even if a master is allowed to be a bridge, the scatternet may be bipartite (e.g. Fig. 1-B). Obviously, if a master is allowed to be a bridge, the scatternet graph may be nonbipartite (e.g. Fig 1-C).

In this section, we shall formulate the *capacity assignment problem* for bipartite and nonbipartite scatternets. We will show that the formulation for nonbipartite scatternets is more complex than the formulation for bipartite scatternets.

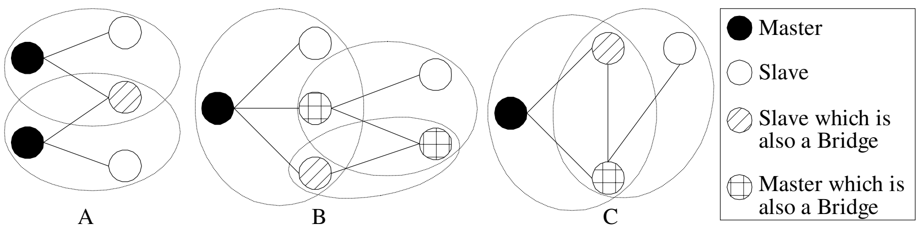

Fig. 1. Scatternet graphs – A bipartite scatternet in which no master is also a bridge (A), a bipartite scatternet in which a master is also a bridge (B), and a nonbipartite scatternet (C)

3.1 Bipartite Scatternets

When a bipartite scatternet graph is given, the nodes can be partitioned into two sets S and T such that no two nodes in S or in T are adjacent. Accordingly, the problem of *scatternet capacity assignment in bipartite graphs* (SCAB) is formulated as follows.

Problem SCAB
Given: Topology of a bipartite graph and flows (f_{ij}).
Objective: Find capacities (c_{ij}) such that the average packet delay is minimized:

$$\min D_T = \min \sum_{(i,j)\in L} D_{ij}(c_{ij}) \tag{1}$$

Subject to:

$$c_{ij} > f_{ij} \quad \forall (i,j) \in L \tag{2}$$

$$\sum_{j\in Z(i)} c_{ij} \le 1 \quad \forall i \in S \tag{3}$$

$$\sum_{j\in Z(i)} c_{ij} \le 1 \quad \forall i \in T \tag{4}$$

The first set of constraints (2) is obvious. Constraints (3) and (4) result from the TDD scheme and reflect the fact that the total capacity of the links connected to a node cannot exceed the maximal capacity of a link. Due to the assumption that the *guard times are negligible*, in (3) and (4) we neglect the time needed in the process of moving a bridge from one piconet to another. Notice that it is easy to see that the polytope defined by (2) - (4) is contained in the *bipartite matching* polytope [14].

3.2 Nonbipartite Scatternets

We shall now show that a formulation similar to the formulation of Problem SCAB is not valid for nonbipartite scatternets. A simple example of a nonbipartite scatternet, given in [1], is illustrated in Fig. 2-A. Constraint (2) and the constraint:

$$\sum_{j\in Z(i)} c_{ij} \le 1 \quad \forall i \in N \tag{5}$$

are not sufficient in order for the capacity vector to be feasible in this example. The capacities described in Fig. 2-A satisfy (2) and (5) but are not feasible because in any scheduling algorithm no two neighboring links can be used simultaneously. If links (1,2) and (1,3) are in use for distinct halves of the available time slots, there are no free slots in which link (2,3) can be in use. Thus, if c_{12} = 0.5 and c_{13} = 0.5, there is no feasible way to assign any capacity to link (2,3), i.e., there is no scheduling algorithm that can allocate the capacities described in the figure.

Baatz et al. [1] suggest that a methodology for finding a *feasible* (not necessarily efficient) capacity assignment[4] will be based on minimum coloring of a graph. They do not develop this methodology and indicate that: "*the example gives an idea of how*

[4] Baatz et al. [1] refer to *piconet presence schedule* instead of *capacity assignment.* A piconet presence schedule determines in which parts of its' time a node is present in each piconet. It is very similar to link capacity assignment as it is described in this paper.

complex the determination of piconet presence schedules may get". We propose a formulation of the problem that is based on the formulations of Problem SCAB and the *matching problem* [14], and that allows obtaining an optimal capacity allocation.

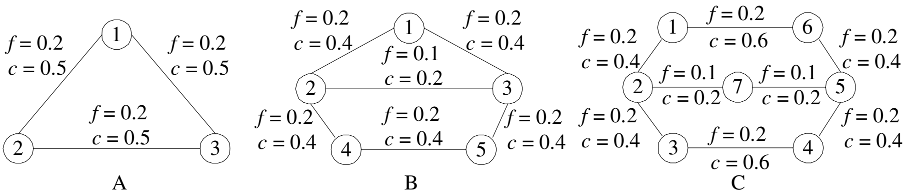

Fig. 2. Examples of scatternets with capacity vectors which are not feasible

It is now obvious that the formulation of the capacity assignment problem for non-bipartite scatternets requires additional constraints to the constraints described in Problem SCAB. For example, one could conclude that the capacity of the links composing the cycle described in Fig. 2-A should not exceed 1. Moreover, one could further conclude that the total capacity of links composing any odd cycle should not exceed: (|links|-1)/2. Namely:

$$\sum_{(i,j)\in C} c_{ij} \le \left(|C|-1\right)/2 \quad \forall C \subseteq L,\ C \text{ odd cycle} \tag{6}$$

However, in the examples given in Fig. 2-B and Fig. 2-C, although the capacities satisfy (6), they cannot be scheduled in any way. Thus, in the following theorem we define a new set of constraints such that the capacity of links connecting nodes in any odd set of nodes U will not exceed (|U|-1)/2.[5] These constraints and the proof of the theorem are based on the properties of the matching problem [10],[14].

Theorem 1. *The capacity vector must satisfy (2),(5), and the following constraints:*

$$\sum_{(i,j)\in L(U)} c_{ij} \le \left(|U|-1\right)/2 \quad \forall U \subseteq N,\ |U|\, odd,\ |U| \ge 3 \tag{7}$$

The proof appears in [18].

The *scatternet capacity assignment* problem (SCA) can now be formulated as follows (for bipartite graphs it reduces to Problem SCAB).

Problem SCA

Given: Topology and flows (f_{ij}).
Objective: Find capacities (c_{ij}) such that the average packet delay is minimized: (1)
Subject to: (2),(5) and (7)

The constraints (2),(5) and (7) form a *convex set* which is included in the matching polytope corresponding to the scatternet graph (for bipartite scatternets these constraints reduce to constraints (2) - (4) described in Problem SCAB.). This set consists of all the *feasible capacity vectors* ($\bar{c}$). Up to now we have not shown that a

---

[5] We note that a similar observation has been recently independently made by Tassiulas and Sarkar [17] who have considered the problem of max-min fair scheduling in scatternets.

feasible capacity vector has a corresponding scheduling algorithm. Namely, that it is possible to determine which links are used in each slot pair such that no two adjacent links are active at the same slot pair and the capacity used by each link is as defined by the capacity vector ($\overline{c}$). This result is shown by the following proposition. We note that the proof of the proposition and the transformation of a capacity vector to a scheduling algorithm are based on the fact that the vertices of the matching polytope are composed of (0,1) variables and on an algorithm described in [10].

Proposition 1. *If a capacity vector $\overline{c}$ satisfies (2),(5) and (7), there is a corresponding scheduling algorithm.*
The proof appears in [18].

4 Optimal Algorithm for Problems SCA and SCAB

In this section a *centralized scatternet capacity assignment algorithm* for finding an optimal solution of Problem SCA, defined in Section 3.2, is introduced.[6] The algorithm is based on the conditional gradient method also known as the Frank-Wolfe method [2, p. 215], which was used for the development of the flow deviation algorithm [3, p. 458]. Gerla et al. [9],[15] have used the Frank-Wolfe method in order to develop bandwidth allocation algorithms for ATM networks. Following their approach, we shall now describe the optimality conditions and the algorithm.

Since the objective of Problem SCA is to minimize a convex function (D_T) over a convex set ((2),(5) and (7)), any local minimum is a global minimum. Thus, necessary and sufficient conditions for the capacity vector $\overline{c}^*$ to be a global minimum are formulated as follows (the following proposition is derived from a well-known theorem [2, p. 194] and, therefore, its proof is omitted).

Proposition 2. *The capacity vector $\overline{c}^*$ minimizes the average delay for Problem SCA, if and only if:*
- *$\overline{c}^*$ satisfies constraints (2),(5) and (7) of Problem SCA.*
- *There are no feasible directions of descent at $\overline{c}^*$; i.e. there does not exist $\overline{c}$ such that*[7]:

$$\nabla D_T(\overline{c}^*)(\overline{c}-\overline{c}^*)<0 \tag{8}$$

$$\sum_{j\in Z(i)} c_{ij} \le 1 \quad \forall i \in N \tag{9}$$

$$\sum_{(i,j)\in L(U)} c_{ij} \le (|U|-1)/2 \quad \forall U \subseteq N,\ |U|\,odd,\ |U|\ge 3 \tag{10}$$

Proposition 2 suggests a steepest descent algorithm in which we can find a feasible direction of descent $\overline{c}$ at any feasible point $\overline{c}^K$ by solving the problem:

$$\min \nabla D_T(\overline{c}^K)\overline{c} \tag{11}$$

subject to - (9),(10) and:

---

[6] The algorithm for the solution of Problem SCAB is similar (the changes are outlined below).
[7] $\nabla D_T(\overline{c}^*)$ is the gradient of D_T with respect to $\overline{c}$ evaluated at $\overline{c}^*$.

$$c_{ij} \geq 0 \quad \forall (i,j) \in L \tag{12}$$

Since the constraint set (10) may include exponentially many constraints, this problem cannot be easily solved using a linear programming algorithm. Yet, since $D_{ij}'(c_{ij}) < 0$ for all c_{ij} (according to Definition 2.4), the formulation of the problem conforms to the formulation of the *maximum-weight matching* problem [14, p. 610], which has a polynomial-time algorithm ($O(n^3)$):

$$\max\left[-\nabla D_T(\overline{c}^K)\overline{c}\right] \tag{13}$$

subject to:

$$\sum_{j\in Z(i)} c_{ij} \leq 1 \quad \forall i \in N \tag{14}$$

$$c_{ij} \in \{0,1\} \quad \forall (i,j) \in L \tag{15}$$

This result and Proposition 2 are the basis for the optimal algorithm, described in Fig 3. The input to the algorithm is the topology, the flows (f_{ij}), a feasible initial solution ($\overline{c}^0$), and the tolerance (t). The output is the optimal capacity vector: $\overline{c}^*$.

1	Set $K = 0$
2	Find the vector $\overline{c}^\#$ - the optimal solution of (13) - (15) (i.e. solve a *maximum-weight matching* problem)
3	Find the value α^* that minimizes $D_T(\alpha\overline{c}^K + (1-\alpha)\overline{c}^\#)$ (α^* may be obtained by any line search method [2, p. 723])
4	Set $\overline{c}^{K+1} = \alpha^*\overline{c}^K + (1-\alpha^*)\overline{c}^\#$
5	If $\nabla D_T(\overline{c}^K)(\overline{c}^K - \overline{c}^\#) \leq t$ then stop
6	Else set $K = K+1$ and go to 2

Fig. 3. An algorithm for obtaining the optimal solution to Problem SCA

We emphasize that unlike the flow deviation algorithm, in which at each iteration a feasible direction is found by solving a shortest path problem, in the capacity assignment algorithm there is a need to solve a maximum-weight matching problem at each iteration. In case the algorithm is applied to Problem SCAB, there is a need to solve a *bipartite* maximum-weight matching problem.

5 Heuristic Algorithm for Problem SCAB

When considering *bipartite scatternets* (Problem SCAB), the initial solution for the optimal algorithm can be obtained using a low complexity *heuristic centralized scatternet capacity assignment* algorithm, presented in this section. In our experiments (see [18]), the results of the heuristic algorithm are very close to the optimal results.

The algorithm is based on the assumption that the delay function conforms to *Kleinrock's independence approximation* [12], described in the following definition.

Definition 5. (Kleinrock's independence approximation) *When neglecting the propagation and processing delay, $D_{ij}(c_{ij})$ is given by*:

$$D_{ij}(c_{ij}) = \begin{cases} f_{ij} / \left(c_{ij} - f_{ij}\right) & c_{ij} > f_{ij} \\ \infty & c_{ij} \le f_{ij} \end{cases}$$

The algorithm assigns capacity to links connected to bridges and to masters which have at least two slaves. Accordingly, we define N' as follows:

Definition 6. N' *is a subgroup of N consisting of bridges and masters which have at least two slaves. Namely*: $N' = \{\ i \mid i \in N \cap \left| j \in Z(i) \right| > 1\ \}$.

We also define the *slack capacity* of a node as follows:

Definition 7. *The* slack capacity of node *i is the maximal capacity which can be added to links connected to the node. It is denoted by s_i and is given by*:

$$s_i = 1 - \sum_{j \in Z(i)} c_{ij}$$

Initially all the link capacities are equal to the flows on the links ($c_{ij} = f_{ij} \quad \forall (i,j) \in L$). The algorithm selects a node from the nodes in N' and allocates the slack capacity to some of the links connected to it. Then, it selects another node, allocates capacity and so on. Once a node (k) is selected, *the slack capacity of this node is allocated to its links whose capacities have not yet been assigned.* The slack capacity is assigned to these links according to the square root assignment [12, p. 20]:

$$c_{kj} = f_{kj} + \frac{s_k \sqrt{f_{kj}}}{\sum\limits_{m:\, m \in Z(k), c_{km} = f_{km}} \sqrt{f_{km}}} \quad \forall j : j \in Z(k), c_{kj} = f_{kj} \tag{16}$$

There are various ways to define the process of *node selection*. For example, nodes can be selected according to their slack capacity or their average slack capacity. However, some of the possible selection methodologies require taking special measures in order to ensure that the obtained capacity vector is feasible (satisfies constraints (2) - (4) of Problem SCAB). We propose a simple selection methodology that guarantees a feasible capacity vector.

It can be shown that after capacity is assigned to a subgroup of the links connected to a node (i) (links whose capacities have not been assigned before), the delay derivatives ($D_{ij}'(c_{ij})$) of all these links will be equal. Accordingly, we define the *delay derivative of a node* as follows:

Definition 8. *The* delay derivative of node *i is proportional to the absolute values of the delay derivatives of the links connected to node i, whose capacities have not yet been assigned. Its value is computed as if node i has been selected as the node whose capacity has to be assigned and the capacities of these links have been assigned according to (16). It is denoted by d_i and is given by*:

$$d_i = \frac{\sum_{m:\, m\in Z(i), c_{im}=f_{im}} \sqrt{f_{im}}}{s_i} \quad (17)$$

Node k, whose link capacities are going to be assigned, is selected from the nodes in N' which are connected to links whose capacities have not yet been allocated. The delay derivatives (d_i's) of all these nodes are computed and the node with the largest derivative is selected. Thus, *the capacities of links with high absolute value of delay derivative, whose delay is more sensitive to the level of capacity, are assigned first.*

The algorithm, which is based on the above methodology, is described in Fig 4. The input is the topology and the flows (f_{ij}), and the output is a capacity vector: $\bar{c}$. It can be seen that the complexity of the algorithm is $O(n^2)$, which is about the complexity of an iteration in the optimal algorithm. Moreover, the following proposition shows that the capacity vector obtained by the algorithm is always feasible.

1 Set $c_{ij}=f_{ij}\ \forall(i,j)\in L$

2 Set $k = \underset{\substack{i\in N' \\ \exists m\in Z(i) \text{ such that } c_{im}=f_{im}}}{\arg\max} \dfrac{\sum_{m:\, m\in Z(i), c_{im}=f_{im}} \sqrt{f_{im}}}{1-\sum_{m\in Z(i)} c_{im}}$

3 Set $c_{kj} = f_{kj} + \dfrac{s_k\sqrt{f_{kj}}}{\sum_{m:\, m\in Z(k), c_{km}=f_{km}} \sqrt{f_{km}}} \quad \forall j : j\in Z(k), c_{kj} = f_{kj}$

4 If there exists $(i,j)\in L$ such that $c_{ij}=f_{ij}$ then go to 2

5 Else stop

Fig. 4. An algorithm for obtaining a heuristic solution to Problem SCAB

Proposition 3. *The heuristic algorithm results in an allocation {* $\bar{c}$ *} that satisfies constraints (2) - (4) of Problem SCAB.*
The proof appears in [18].

6 Conclusions and Future Study

This paper presents an analytical study of the capacity assignment problem in Bluetooth scatternets. The problem has been formulated for bipartite and nonbipartite scatternets, using the properties of the matching polytope. Then, we have introduced a centralized algorithm for obtaining its optimal solution. A heuristic algorithm for the solution of the problem in bipartite scatternets, which has a relatively low complexity, has also been described. Several numerical examples can be found in [18].

The work presented here is the first approach towards an analysis of the scatternet performance. Hence, there are still many open problems to deal with. For example, *distributed protocols* are required for actual Bluetooth scatternets and, therefore, future study will focus on developing optimal and heuristic distributed protocols.

Moreover, in this paper we have made a few assumptions regarding the properties of the *delay function*. An analytical model for the computation of bounds on the delay is required in order to evaluate these assumptions. In addition, it might enable developing efficient piconet scheduling algorithms.

Finally, we note that a major future research direction is the development of capacity assignment protocols that will be able to deal with various quality of service requirements and to interact with scatternet formation, scheduling, and routing protocols.

References

1. Baatz, S., Frank, M., Kühl, C., Martini, P., Scholz, C.: Adaptive Scatternet Support for Bluetooth using Sniff Mode. Proc. IEEE LCN'01 (Nov. 2001)
2. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific, Massachusetts (1999)
3. Bertsekas, D.P., Gallager, R.: Data Networks. Prentice Hall, New Jersey (1992)
4. Bhagwat, P., Rao, S.P.: On the Characterization of Bluetooth Scatternet Topologies. Submitted for publication (Feb. 2002)
5. Bluetooth Special Interest Group: Specification of the Bluetooth System - Version 1.1. (Feb. 2001)
6. Bray, J., Sturman, C.: Bluetooth connect without cables. Prentice Hall (2001)
7. Bruno, R., Conti, M., Gregori, E.: Wireless Access to Internet via Bluetooth: Performance Evaluation of the EDC Scheduling Algorithm. Proc. ACM WMI'01 (July 2001)
8. Das, A., Ghose, A., Razdan, A., Saran, H., Shorey, R.: Enhancing Performance of Asynchronous Data Traffic over the Bluetooth Wireless Ad-hoc Network. Proc. IEEE INFOCOM'01 (Apr. 2001)
9. Gerla, M., Monteiro, J.A.S., Pazos-Rangel, R.A.: Topology Design and Bandwidth Allocation in ATM Nets. IEEE JSAC **7** (Oct. 1989) 1253-1262
10. Hajek, B., Sasaki, G.: Link Scheduling in Polynomial Time. IEEE Trans. on Information Theory **34** (Sep. 1988) 910-917
11. Johansson, P., Kazantzidis, M., Kapoor, R., Gerla, M.: Bluetooth: An Enabler for Personal Area Networking. IEEE Network **15** (Sep./Oct. 2001) 28-37
12. Kleinrock, L.: Communication Nets: Stochastic Message Flow and Delay. McGraw-Hill, New York (1964)
13. Law, C., Mehta, A.M., Siu, K.Y.: Performance of a New Bluetooth Scatternet Formation Protocol. Proc. ACM MOBIHOC'01 (Oct. 2001)
14. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization. John Wiley and Sons (1988)
15. Pazos-Rangel, R.A., Gerla, M.: Express Pipe Networks. Proc. Global Telecommunications Conf. (1982) B2.3.1-5
16. Racz, A., Miklos, G., Kubinszky, F., Valko, A.: A Pseudo Random Coordinated Scheduling Algorithm for Bluetooth Scatternets. Proc. ACM MOBIHOC'01 (Oct. 2001)
17. Tassiulas, L., Sarkar, S.: Maxmin Fair Scheduling in Wireless Networks. Proc. IEEE INFOCOM'02 (to appear)
18. Zussman, G., Segall, A.: Capacity Assignment in Bluetooth Scatternets - Analysis and Algorithms. CCIT Report 355, Technion - Department of Electrical Engineering, (Oct. 2001) Available at URL: http://www.comnet.technion.ac.il/segall/Reports.html