

Capacity Assignment in Bluetooth Scatternets – Analysis and Algorithms

Gil Zussman¹ 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 Corresponding author - phone: 972-4-829-4749, fax: 972-4-832-3041

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],[7],[12]. Since its announcement in late spring 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*.

Bluetooth channels use a frequency-hop/time-division-duplex (FH/TDD) scheme. The channel is divided into 625- μ sec 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 (see for example [4],[14] and references therein).

Much attention has also been given to scheduling algorithms for piconets and scatternets. In the Bluetooth specifications [6], 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 polling/scheduling algorithms for piconets have been proposed and evaluated via simulation (see for example [8],[9],[11] and references therein). In [12] 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 [17].

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 issue¹. 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, scatternet 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 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 and algorithms). For example, in the problem presented by Bertsekas and Gallager [3, p. 439], the objective is to select link capacities so as to minimize the total cost, subject to the constraint that the average delay per packet should not exceed a given level.

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, there is no cost associated with each link and no budget constraint.
- 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

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

knowledge, the work presented in this paper is the first attempt to analytically analyze 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 show that the scatternet capacity assignment problem is more complex than it seems at first glance and that different formulations apply to bipartite and nonbipartite scatternets. We prove that the problem can be formulated as a minimization of a convex function over a polytope contained in the polytope of the well-known *matching problem* [15, p. 608] and use this formulation in order to identify a few properties of the problem. The methodology used by Gerla et al. [10] and Pazos-Rangel and Gerla [16] is used in order to develop an *optimal scatternet capacity assignment algorithm* which is similar to the well-known *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, the proofs of the theorems and propositions are omitted. These proofs and several numerical examples 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 obtaining the solution in a bipartite scatternet. Section 6 summarizes the main results and discusses possible extensions.

2 Model and Preliminaries

Consider the connected undirected scatternet graph $G = (N, L)$. N will denote the collection of *nodes* $\{1, 2, \dots, n\}$. Each of the nodes could be a master, a slave, or a bridge¹. 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 piconets², 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).

¹ A bridge participates in different piconets, on a time-sharing basis.

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

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 ($F_{ij} > 0 \quad \forall (i,j) \in L$). 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 packets¹. 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. $\lim_{c_{ij} \rightarrow f_{ij}} D_{ij}(c_{ij}) = \infty$
4. $D_{ij}'(c_{ij}) < 0$ for all c_{ij} where D_{ij}' is the derivative of D_{ij} .

Using Definition 1, we 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 .

As we have mentioned, slot allocation is dynamic and it is managed by the masters. Accordingly, 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 [6] do not require that different masters' clocks will be synchronized. On the contrary, since the clocks are not synchronized a guard time

¹ For example, currently the maximal flow on a symmetrical link, when using five slots unprotected data packets (DH5), is 867.8 Kbits/second.

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 the following.

Assumption 1. *The guard times are negligible.*

This assumption allows us to consider a scheduling algorithm for the whole scatternet (which master-slave links are active in each slot pair).

3 Formulation of the Problem

Scatternet graphs can be *bipartite graphs* or *nonbipartite graphs* [4] (see Fig. 1). For example, all the scatternets in which no master is allowed to be a bridge are bipartite. In this section, we shall show that the formulation of the *capacity assignment problem* for nonbipartite scatternets is more complex than the formulation for bipartite scatternets. We will also identify a few properties of the capacity vector.

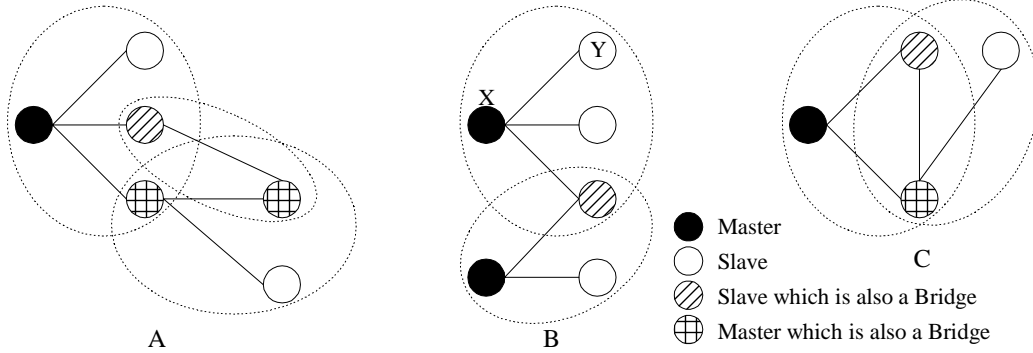


Fig. 1. Scatternet graphs – A bipartite scatternet (A), a bipartite scatternet in which no master is also a bridge (B), and a nonbipartite scatternet (C)

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:

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

$$\text{Subject to: } c_{ij} > f_{ij} \quad \forall (i, j) \in L \quad (2)$$

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

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

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 Assumption 1, in (3) and (4) we neglect the guard 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 [15].

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. In this example, constraint (2) and the constraint:

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

are not sufficient in order for the capacity vector to be feasible. 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.

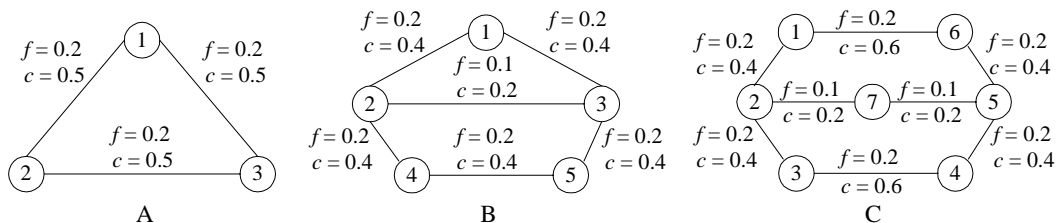


Fig. 2. Examples of scatternets with capacity vectors which are not feasible

Baatz et al. [1] suggest that a methodology for finding a *feasible* (not necessarily efficient) capacity assignment¹ 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 complex the determination of piconet presence schedules may get*”. In this paper, we propose a formulation of the problem that is based on the formulations of Problem SCAB and of the *matching problem* [15]. This formulation allows obtaining an optimal capacity allocation.

It is now obvious that the formulation of the capacity assignment problem for nonbipartite 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 fur-

¹ 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.

then conclude that the total capacity of links composing any odd cycle should not exceed: $(|links|-1)/2$. Namely:

$$\sum_{(i,j) \in C} c_{ij} \leq \frac{|C|-1}{2} \quad \forall C \subseteq L, C \text{ odd cycle} \quad (6)$$

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 $(|U|-1)/2$. These constraints and the proof of the theorem are based on the properties of the matching problem.

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

$$\sum_{(i,j) \in L(U)} c_{ij} \leq \frac{|U|-1}{2} \quad \forall U \subseteq N, |U| \text{ odd}, |U| \geq 3 \quad (7)$$

The proof appears in [18].

Notice that the polytope defined by (2),(5) and (7) is included in the matching polytope corresponding to the scatternet graph. Notice also that for bipartite scatternets the constraints described in Theorem 1 reduce to constraints (2) - (4) described in Problem SCAB.

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)

3.3 Properties of the Capacity Vector

Each of the constraint sets (2) - (4), and (2),(5),(7) forms a *convex set*. These sets consist of all the *feasible capacity vectors* (\bar{c}) for the corresponding problem (SCAB and SCA). A capacity vector that achieves the minimal delay will be denoted by \bar{c}^* .

Up to now we have not shown that a 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 (\bar{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.

Proposition 1. *If a capacity vector \bar{c} satisfies (2),(5) and (7), there is a corresponding scheduling algorithm.*

The proof appears in [18].

Finally, it is easy to see that due to Definition 2.4, in an optimal allocation, some of the nodes must utilize their full capacity (i.e. for such nodes (3),(4) or (5) is satisfied

with equality). Such nodes are connected to at least one node which has a single neighbor. On the other hand, in a scatternet consisting of more than two nodes, nodes that have a single neighbor cannot utilize their full capacity. For example, in an optimal allocation, master X in Fig. 1 must utilize its full capacity and slave Y cannot utilize its full capacity. These two properties are useful when trying to solve simple capacity assignment problems in scatternets composed of a few nodes and a few links. They are also useful in order to construct optimal capacity assignment algorithms.

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 introduced¹. 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]. Therefore, we refer to the algorithm as the *scatternet capacity deviation* (SCD) algorithm.

Gerla et al. [10] and Pazos-Rangel and Gerla [16] 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 \bar{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 \bar{c}^* minimizes the average delay for Problem SCA, if and only if:*

- \bar{c}^* satisfies constraints (2),(5) and (7) of Problem SCA.
- There are no feasible directions of descent at \bar{c}^* ; i.e. there does not exist \bar{c} such that²:

$$\nabla D_T(\bar{c}^*)(\bar{c} - \bar{c}^*) < 0 \quad (8)$$

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

$$\sum_{(i,j) \in L(U)} c_{ij} \leq \frac{|U|-1}{2} \quad \forall U \subseteq N, |U| \text{ odd}, |U| \geq 3 \quad (10)$$

Notice that equation (8) means that D_T cannot be reduced by moving to \bar{c} , whereas (9) and (10) restrict \bar{c} to the feasible region. Notice also that due to the property described in Definition 2.3 ($D_T \rightarrow \infty$ when $c_i \rightarrow f_i$), the first set of constraints of Problem SCA (2) is included in the objective function as a penalty function and is not needed

¹ The algorithm for the solution of Problem SCAB is similar (the changes are outlined below).

² $\nabla D_T(\bar{c}^*)$ is the gradient of D_T with respect to \bar{c} evaluated at \bar{c}^* .

in order to restrict \bar{c} to the feasible region. It is easy to see that a similar optimality condition holds for Problem SCAB (explicitly, (9) and (10) should be replaced with (3) and (4)).

Proposition 1 suggests a steepest descent algorithm in which we can find a feasible direction of descent \bar{c} at any feasible point \bar{c}^K by solving the following problem:

$$\min \nabla D_T(\bar{c}^K) \bar{c} \quad (11)$$

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

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

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 [15, p. 610], which has a polynomial-time algorithm ($O(n^3)$):

$$\max [-\nabla D_T(\bar{c}^K) \bar{c}] \quad (13)$$

subject to:

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

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

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

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

Fig. 3. Algorithm SCD for obtaining an 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 Algorithm SCD there is a need to solve a maximum-weight matching problem at each iteration. In case Algorithm SCD is applied to Problem SCAB, the constraints (9),(10) and (12) reduce to the constraints of the *bipartite* maximum-weight matching problem.

5 Heuristic Algorithm for Problem SCAB

When considering *bipartite scatternets* (Problem SCAB), the initial solution for Algorithm SCD, introduced in the previous section, can be obtained using a low complexity *heuristic centralized scatternet capacity assignment* (HCSCA) 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* [13], 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:*

$$D_{ij}(c_{ij}) = \begin{cases} \frac{f_{ij}}{c_{ij} - f_{ij}} & c_{ij} > f_{ij} \\ \infty & c_{ij} \leq f_{ij} \end{cases}$$

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: $N' = \{ i \mid i \in N \cap |j \in Z(i)| > 1 \}$.

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 ($c_{ij} = f_{ij} \forall (i,j) \in L$). The algorithm selects a node from the nodes in N' and allocates the slack capacity to some of the links connected to that node. 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* (links in which $c_{kj} = f_{kj}$). The slack capacity is assigned to these links according to the square root assignment [13, p. 20]:

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

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 value of the delay derivatives of the links¹ 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 delay derivative is selected.

Consequently, the capacities of links with high absolute value of delay derivative, whose delay is more sensitive to the level of capacity, are assigned first. Moreover, the following proposition shows that the capacity vector obtained by the algorithm is always feasible.

Proposition 3. Algorithm HCSCA results in an allocation $\{\bar{c}\}$ that satisfies constraints (2) - (4) of Problem SCAB.

The proof appears in [18].

Algorithm HCSCA, which is based on the above methodology, is described in Fig. 4. The input to the algorithm 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 (Algorithm SCD).

1	Set $c_{ij} = f_{ij} \quad \forall (i, j) \in L$
2	Set $k = \underset{\substack{i \in N' \\ \exists m \in Z(i) \text{ such that } c_{im} = f_{im}}}{\text{arg max}} \frac{\sum_{m: m \in Z(i), c_{im} = f_{im}} \sqrt{f_{im}}}{1 - \sum_{m \in Z(i)} c_{im}}$
3	Set $c_{kj} = f_{kj} + \frac{s_k \sqrt{f_{kj}}}{\sum_{m: m \in Z(k), c_{km} = f_{km}} \sqrt{f_{km}}} \quad \forall j: j \in Z(k), c_{kj} = f_{kj}$
4	If there exists $(i, j) \in L$ such that $c_{ij} = f_{ij}$ then go to 2
5	Else stop

Fig. 4. Algorithm HCSCA for obtaining a heuristic solution to Problem SCAB

¹ According to Definition 2.4 the delay derivative of a link is always negative.

6 Conclusions and Future Study

This paper analytically analyzes the problem of capacity assignment 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 described¹.

The work presented here is the first approach towards an analytical 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, according to Assumption 1, the guard times are negligible. This is of course not the situation in a real scatternet. Thus, in the future we intend to investigate the effect of more realistic assumptions on the formulation of the problem.

Furthermore, recall that Definition 2 describes a few assumptions regarding the properties of the delay function, and that we have assumed that the average packet delay on a *bi-directional* link is a function of the *total* link flow and capacity. 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 in the future, capacity assignment protocols will interact with protocols responsible for scatternet formation, scheduling, and routing. Thus, the main challenge is to develop a combined capacity assignment and inter-piconet scheduling protocol that will operate efficiently in the presence of other protocols.

References

- [1] S. Baatz, M. Frank, C. Kühn, P. Martini and C. Scholz, "Adaptive Scatternet Support for Bluetooth using Sniff Mode", *Proc. IEEE LCN'01*, Nov. 2001.
- [2] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Massachusetts, 1999.
- [3] D. P. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall Inc., New Jersey, 1992.
- [4] P. Bhagwat and S. P. Rao, "On the Characterization of Bluetooth Scatternet Topologies", Submitted for publication, Oct. 2001.
- [5] The Bluetooth Special Interest Group, Documentation available at URL <http://www.bluetooth.com>, Oct. 2001.
- [6] Bluetooth Special Interest Group, *Specification of the Bluetooth System - Version 1.1*, Feb. 2001.
- [7] J. Bray and C. Sturman, *Bluetooth connect without cables*, Prentice Hall, 2001.

¹ Several numerical examples can be found in [18].

- [8] R. Bruno, M. Conti and E. Gregori, "Wireless Access to Internet via Bluetooth: Performance Evaluation of the EDC Scheduling Algorithm", *Proc. ACM WMI'01*, July 2001.
- [9] A. Das, A. Ghose, A. Razdan, H. Saran, and R. Shorey, "Enhancing Performance of Asynchronous Data Traffic over the Bluetooth Wireless Ad-hoc Network", *Proc. IEEE INFOCOM'01*, Apr. 2001.
- [10] M. Gerla, J.A.S. Monteiro and R. A. Pazos-Rangel, "Topology Design and Bandwidth Allocation in ATM Nets", *IEEE J. on Selected Areas in Comm.*, Vol. 7, pp. 1253-1262, Oct. 1989.
- [11] N. Johansson, U. Korner and P. Johansson, "Performance Evaluation of Scheduling Algorithms for Bluetooth", *Proc. IFIP TC6 Int. Conf. on Broadband Communications*, Nov. 1999.
- [12] P. Johansson, M. Kazantzidis, R. Kapoor and M. Gerla, "Bluetooth: An Enabler for Personal Area Networking", *IEEE Network*, pp. 28-36, Sep./Oct. 2001.
- [13] L. Kleinrock, *Communication Nets: Stochastic Message Flow and Delay*, McGraw-Hill, New York, 1964.
- [14] C. Law, A. M. Mehta and K.Y. Siu, "Performance of a New Bluetooth Scatternet Formation Protocol", *Proc. ACM MOBIHOC'01*, Oct. 2001.
- [15] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, John Wiley and Sons, 1988.
- [16] R. A. Pazos-Rangel and M. Gerla, "Express Pipe Networks", *Proc. Global Telecommunications Conf.*, pp. B2.3.1-5, 1982.
- [17] A. Racz, G. Miklos, F. Kubinszky and A. Valko, "A Pseudo Random Coordinated Scheduling Algorithm for Bluetooth Scatternets", *Proc. ACM MOBIHOC'01*, Oct. 2001.
- [18] G. Zussman and A. Segall, "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>.