Performance Aspects of Bluetooth Scatternet Formation

Gy. Miklós*, A. Rácz*, Z. Turányi*, A. Valkó*, P. Johansson**

*Traffic Analysis and Network Performance Lab, Ericsson Research
Budapest, Hungary
** Ericsson Radio Systems, Ericsson Research
Stockholm, Sweden

Abstract

The emergence of Bluetooth as a default radio interface allows handheld electronic devices to be rapidly interconnected into ad hoc networks. Bluetooth allows large numbers of piconets to form a scatternet using designated nodes that participate in multiple piconets. In this paper we study the performance implications of forming scatternets from piconets. The contribution of the paper is twofold. First, we establish a network model and define performance metrics for Bluetooth scatternets. Our model is derived from constraints specific to the Bluetooth technology, but is sufficiently abstract to relate to the more general field of ad hoc networking. Second, using a number of simulation studies, we relate scatternet parameters to performance metrics and discover correlations between scatternet formation rules and performance. These results reveal some important performance implications of scatternet design decisions and can serve as guidelines for future scatternet formation algorithms.

1 Introduction

Short-range radio technologies enable users to rapidly interconnect handheld electronic devices such as cellular phones, palm devices or notebook computers. The emergence of Bluetooth [1] as a default radio interface in these devices provides an opportunity to turn them from stand-alone tools into networked equipment. Building Bluetooth ad hoc networks also represents, however, a number of new challenges, partly stemming from the fact that Bluetooth was originally developed for single hop wireless connections. In this paper we study some performance aspects of Bluetooth ad hoc networks and derive guidelines for forming such networks.

Bluetooth is a short range radio technology operating in the unlicensed ISM (Industrial-Scientific-Medical) band using a frequency hopping scheme, where the hopping is performed on 79 RF channels spaced 1 MHz apart. Bluetooth (BT) units are organized into *piconets*. There is one Bluetooth device in each piconet that acts as the master, which can have any number of slaves out of which up to seven can be active simultaneously. Being a master or a slave is only a logical state: any Bluetooth unit can be a master or a slave. The Bluetooth

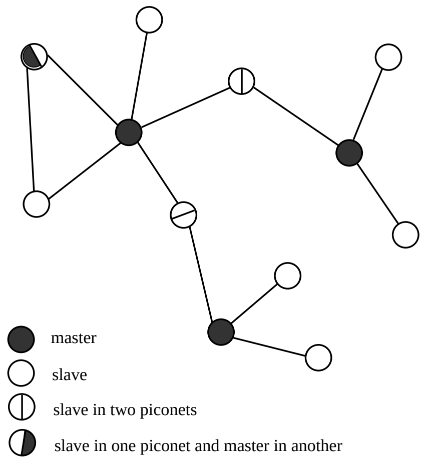


Figure 1: Example scatternet

system provides full-duplex transmission based on slotted Time Division Duplex (TDD) scheme, where each slot is 0.625 ms long. Master-to-slave transmission always starts in an even-numbered time slot while slave-to-master transmission always starts in an odd-numbered time slot. The communication within a piconet is organized by the master which polls each slave according to some polling scheme. A slave is only allowed to transmit in the current slave-to-master slot if it has been polled by the master in the previous master-to-slave slot. The master may or may not include data in the packet used to poll a slave. Bluetooth packets can be 1, 3 or 5 slots long and they can carry synchronous data (e.g., real-time traffic) on Synchronous Connection Oriented (SCO) links or asynchronous data (e.g., elastic data traffic, which is the case in our study) on Asynchronous Connectionless (ACL) links. For further information the reader is referred to [1, 2].

A Bluetooth unit can participate in more than one piconet at any time but it can be a master in only one piconet. A unit that participates in multiple piconets can serve as a bridge thus allowing the piconets to form a larger network. A set of piconets that are all interconnected by such bridging units is referred to as a *scatternet* network (Figure 1). Since a Bluetooth unit can transmit or receive in only one piconet at a time, bridging units must switch between piconets on a time division basis. Due to the need to synchronize its radio from one piconet to another and perform the necessary signaling, a Bluetooth unit necessarily loses some time while switching, which represents an important performance constraint in building scatternets.

A given set of physical nodes can be arranged in scatternets in a number of alternative ways within the hard constraint represented by physical distance. By grouping nodes into piconets and by selecting the bridging nodes we can significantly impact resulting performance parameters, such as the attained throughput. Knowing the scatternet topology that optimizes performance would be of prime importance for future Bluetooth implementations. However, the complexity of the problem seems to prohibit an analytical approach and the large number of free parameters makes it problematic to use numerical optimization through simulation.

As an alternative approach, in this paper we study the relationship between scatternet design rules and performance parameters and derive guidelines for design. To study this relationship we take a statistical approach.

First, we develop simple heuristic algorithms to generate scatternets with desired parameters such as the number of links or piconets. Next, we perturb these algorithms and re-run them a large number of times to generate a statistically representative sample of all possible scatternets, given the same set of physical nodes, that meet the desired parameters. Using a simple elastic traffic model we simulate the resulting scatternets and evaluate their performance in terms of throughput and scatternet efficiency. By repeating these steps for large numbers of alternative parameter settings, we effectively determine the relationship between scatternet formation rules and performance. These studies reveal a number of interesting connections between design rules and performance metrics which can serve as guidelines for future scatternet formation algorithms.

The remainder of this paper is structured as follows. In Section 2 we present an overview of related work available in the literature. Section 3 introduces our Bluetooth link and network model and defines the scatternet metrics. In Section 4 the statistical scatternet generation algorithms are discussed. Finally in Section 5 we present numerical results.

2 Related Work

The recent emergence of the Bluetooth technology has produced large numbers of new research questions, only a few of which have been investigated in details so far. Physical layer properties of the Bluetooth radio have been studied thoroughly in [6]. The authors find that the frequency hopping scheme makes Bluetooth robust against interference even in the case of large numbers of interfering devices.

A number of researchers have addressed issues related to medium access control (MAC) where a fundamental question is the scheduling discipline used by a piconet master in polling the slaves. In [5] the authors assume a simple round robin polling scheme and investigate queuing delays in master and slave units depending on the length of the Bluetooth packets used. In [4] Johansson *et al.* analyze and compare the behavior of three different polling algorithms. They conclude that the simple round robin scheme may perform poorly in Bluetooth systems and they propose a scheme called Fair Exhaustive Polling. The authors demonstrate the strength of this scheme and argue in favor of using multi-slot packets. Similar conclusions are drawn by Kalia *et al.* who argue that the traditional round robin scheme may result in waste and unfairness [7]. The authors propose two new scheduling disciplines that utilize information about the status of master and slave queues.

In inter-piconet communications where the bridging Bluetooth device participates in two or more piconets on a time sharing basis, the scheduling problem is augmented by the need to co-ordinate between neighboring piconets. This inter-piconet scheduling scheme, together with the scatternet topology which is the subject of the present paper, is an important component that determines the throughput, reliability and other performance parameters of a scatternet. Up to the authors' knowledge, no thorough analysis is available in the literature regarding these issues.

Some protocol architectural aspects of Bluetooth ad-hoc networking are discussed in [9] using a Bluetooth based Local Area Networks (LAN) access scenario. In this paper the principles of Cellular IP [11] and Mobile

IP [10] are reused to design a LAN access network architecture using Bluetooth access points.

Finally, routing issues in ad hoc networks in general have been exhaustively studied in the Mobile Ad Hoc Networks (MANET) group of the Internet Engineering Task Force (IETF) [3]. Routing algorithms developed therein are probably also applicable for Bluetooth scatternets. However, Bluetooth scatternets exhibit some characteristical properties that might differ from the assumptions used in some generic ad hoc routing algorithms. This fact suggests that routing algorithms designed specifically for Bluetooth ad hoc networks might provide better performance than some generic ad hoc routing protocols when used in a Bluetooth environment. A source routing algorithm tailored for Bluetooth is presented in [8].

3 Bluetooth Model

In the following subsections we will establish the Bluetooth link and network models. Then we will define a set of system parameters and performance metrics that will serve as basis for our studies.

3.1 Bluetooth Link Model

Bluetooth links support six different packet types denoted by DM1, DH1, DM3, DH3, DM5 and DH5 where DM refers to FEC (Forward Error Correction) coded packets, DH to uncoded packets and the numbers indicate the packet length in number of slots. Larger packets represent lower overhead and consequently higher user data throughput, as is illustrated in Table 1 [1]. Since in our investigation throughput is an important quality measure, we will assume that uncoded, 5-slot long packets (DH5) are used whenever possible, as suggested in [6]. In the case of links that carry asymmetrical traffic we will assume that DH5 packets are used in the direction with higher traffic (hereafter referred to as *forward direction*) and DH1, DH3 or DH5 packets in the other (*reverse*) direction.

Packet type	Length (slots)	User payload (bytes)	Capacity efficiency
DH1	1	27	$\mu_1 = 0.3296$
DH3	3	183	$\mu_3 = 0.7446$
DH5	5	339	$\mu_5 = 0.8276$

Table 1: Available bit rates using DH packets

Let r_f and r_r denote the available bit rate for user data in the forward and reverse directions of a given link, respectively. Since this includes both carried data and empty slots, let us define the throughput (t_f and t_r) as the actual amount of carried user payload in each direction. Finally, let a_f and a_r denote the capacity allocation in the forward and reverse directions, respectively. This includes the MAC protocol overhead, unused capacity in empty slots and the user payload.

Let us now consider the reverse packets in the link and let the fraction of DH1, DH3 and DH5 reverse packets be denoted by l_1, l_3 and l_5, respectively ($l_1 + l_3 + l_5 = 1$). Though in Bluetooth a forward packet is always followed by a reverse slot, reverse traffic may sometimes be so little that a DH5 forward packet is not followed by any user data in the reverse direction. In the limiting case where there is no reverse data carried over a link, let $l_1 = 1$. The average reverse packet length is

$$l = 1l_1 + 3l_3 + 5l_5. \tag{1}$$

We recall that in our model the forward direction uses DH5 packets only.

If the amount of allocation to a link is c, then using the previously introduced notations the capacity allocation in the forward and reverse directions can be expressed as

$$a_f = \frac{5}{5+l}c \text{ and } a_r = \frac{l}{5+l}c.$$

Let us define the asymmetry factor of allocations κ as

$$\kappa_a = \frac{a_f}{a_r} = \frac{5}{l}.$$

To determine the amount of available bitrates in the forward and reverse directions we rely on Table 1 from the Bluetooth baseband specification [1]. Here, capacity efficiency was defined as the number of user payload bits over the maximum number of bits that could be transmitted in the same number of slots by the physical layer. The capacity of the physical layer was 1 Mbit/sec, and one slot duration to be 1/1600 sec.

The available bit rate is then determined as

$$r_f = \mu_5 a_f; \tag{2}$$

$$r_r = (\mu_1 l_1 + \mu_3 l_3 + \mu_5 l_5) a_r. \tag{3}$$

That is, the capacity efficiency of the forward and reverse directions can be expressed as μ_5 and $\mu_1 l_1 + \mu_3 l_3 + \mu_5 l_5$, respectively. For a given l, the distribution of l_1, l_3 and l_5 should be determined such that the capacity efficiency be maximized. Let us first determine the optimal distribution for the case where $l \geq 3$. In this case, due to the fact that $\mu_1 < \mu_3 < \mu_5$ it is unreasonable to use DH1 packets. Therefore the optimal setting will have $l_1 = 0$ and, from Equation 1, $l_3 = (5 - l)/2$ and $l_5 = (l - 3)/2$. Let us consider now the case where $l < 3$. In this case Equation 1 yields $l_1 = 3/2 - l/2 + l_5$ and $l_3 = l/2 - 2l_5 - 1/2$. Let us use this result to express the capacity efficiency as a function of l_5. We obtain a linear expression of l_5 ($\mu_1(3/2 - l/2 + l_5) + \mu_3(l/2 - 2l_5 - 1/2) + \mu_5 l_5$) where the coefficient of l_5 is $\mu_1 - 2\mu_3 + \mu_5 = -0.3320 < 0$. We conclude that the capacity efficiency is maximized by setting $l_5 = 0$ which, using Equation 1, further leads us to $l_1 = (3 - l)/2$ and $l_3 = (l - 1)/2$.

The capacity efficiency of the link is

$$\mu = \frac{r_f + r_r}{a_f + a_r}$$

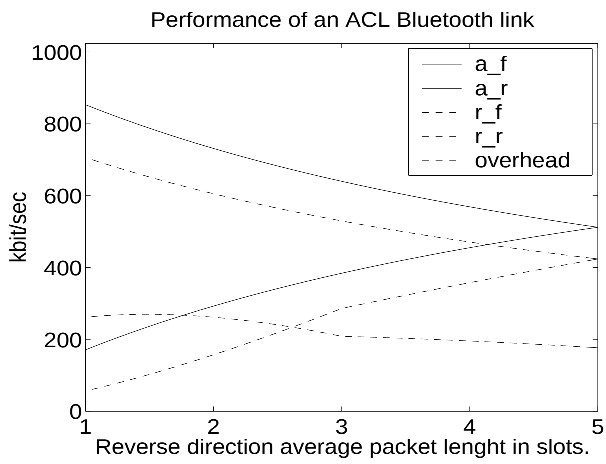


Figure 2: Capacity of an asymmetric (ACL) Bluetooth link: amount of allocation, available user rates in the forward and reverse direction.

In the analogy of κ_a, let us define κ_r, the asymmetry factor of available bitrates as

$$\kappa_r = \frac{r_f}{r_r} = \frac{5\mu_5}{(l_1 + 3l_3 + 5l_5)(\mu_1 l_1 + \mu_3 l_3 + \mu_5 l_5)}.$$

Since $1 = l_1 + l_3 + l_5$, it follows easily that $1 \leq \kappa_r \leq \frac{5\mu_5}{\mu_1} = 12.55 = \kappa_{r,\max}$.

Further, let us also define κ_t, the asymmetry factor of available throughput as

$$\kappa_t = \frac{t_f}{t_r}.$$

For the forward direction throughput, we can assume $t_f = r_f$ since $t_f < r_f$ would mean wasting capacity. In the reverse direction $t_r = r_r$ might not be possible due to the fact that we have a limit on the asymmetry factor, hence we have $r_r \geq r_f/\kappa_{r,\max}$. So if $\kappa_t > \kappa_{r,\max}$, then $t_r < r_r = r_f/\kappa_{r,\max}$, otherwise $t_r = r_r$.

Based on the equations that we have established, we can now calculate a number of important performance parameters of a Bluetooth ACL link as a function of l, the average reverse packet size. Figure 2 shows some of these parameters. Here we have plotted the capacity allocation and the available bit rate in the forward and reverse directions (a_f, a_r, r_f and r_r). In addition, we have plotted the overhead, defined as the total capacity minus the total available bitrate, $C - r_f - r_r$.

We note that the relative overhead in the reverse direction (a_r/r_r) is higher for small reverse packets (l close to 1). However the total overhead of the link does not decrease as steeply as the relative overhead of the reverse direction when l goes from 1 to 5 because the amount of reverse direction allocation decreases in absolute value. This explains why the overhead can even increase at $l = 1$.

From the user's perspective, the Bluetooth link can handle asymmetric traffic but the overhead on the link depends on the asymmetry. To quantify this dependence, we define the relative overhead on the link as

$$\omega = 1 - \frac{t_f + t_r}{a_f + a_r}.$$

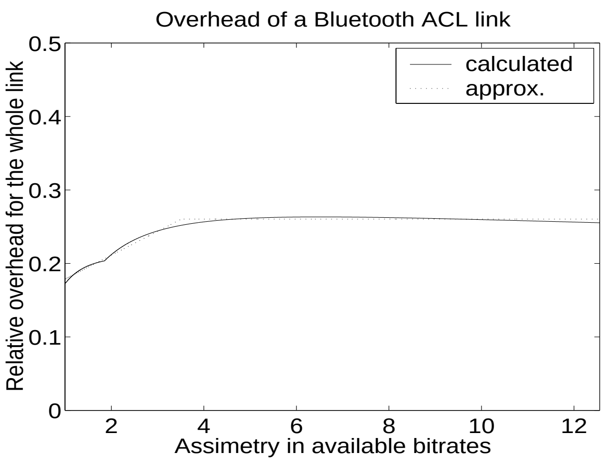


Figure 3: Overhead of an asymmetric (ACL) Bluetooth link.

In Figure 3 we have plotted ω as a function of the asymmetry factor of available bitrates, κ_r, where κ_r goes from 1 to $\kappa_{r,\max} = 12.55$ (note that in this range $\kappa_t = \kappa_r$). One can observe that the overhead increases from approximately 0.15 for $\kappa_r = 0$ to approximately 0.25 at $\kappa_r \sim 4$ and from there on it remains nearly constant. For ease of calculations we approximate this curve using two linear segments (shown in dotted lines in the figure) with the following coefficients (determined by least mean squares regression):

$$\omega_{model}(\kappa_r) = \begin{cases} 0.0325\kappa_r + 0.1463 & \text{if } 1 \leq \kappa_r \leq 3.5108 \\ 0.2604 & \text{if } 3.5108 < \kappa_r \leq \kappa_{r,\max} = 12.55 \end{cases}$$

When $\kappa_t \leq \kappa_{r,\max}$ then $\kappa_t = \kappa_r$ and the relative overhead is $\omega = \omega(\kappa_t)$. When $\kappa_t > \kappa_{r,\max}$ then the overhead of the system relative to the *total available bitrate* is $\omega(\kappa_{r,\max})$, but $r_f + r_r > t_f + t_r$, i.e., we have additional overhead due to the fact that the total throughput is less than the total available bitrates. This increases the overhead to

$$\begin{aligned} \omega &= 1 - \frac{t_f + t_r}{a_f + a_r} \\ &= 1 - \left(\frac{r_f + r_r}{a_f + a_r}\right)\left(\frac{t_f + t_r}{r_f + r_r}\right) \\ &= 1 - \omega(\kappa_{r,\max})\frac{t_f + t_r}{r_f + r_r} \\ &= 1 - \omega(\kappa_{r,\max})\frac{t_f(1 + t_r/t_f)}{t_f + t_f/\kappa_{r,\max}} \end{aligned}$$

which yields

$$\omega = 1 - \omega(\kappa_{r,\max})\frac{1 + 1/\kappa_t}{1 + 1/\kappa_{r,\max}}.$$

3.2 Bluetooth Network Model

Using the Bluetooth link model established in the previous subsection we can determine the amount of allocation needed to carry a certain amount of traffic over a Bluetooth link. We can extend our model for a Bluetooth network as follows. Each Bluetooth device is represented by a node in the network model and each Bluetooth link is modeled as in the previous subsection. Let $L(j)$ denote the set of links connected to node j in the scatternet graph. Then for each node j, the total capacity imposes the following constraint:

$$\sum_{i \in L(j)} a_f(i) + a_r(i) \leq C$$

where $a_f(i) + a_r(i)$ is the allocation on the link in the forward and reverse directions.

Using this model, it is possible to determine the amount of allocation and traffic that can be carried on a Bluetooth scatternet without determining the master and slave roles. We note, however, that this model has several simplifications. In particular, it does not take into account the intra piconet overhead (i.e., overhead associated with polling and with the various power save modes) and it does not take into account the overhead of bridging from one piconet to another and the losses due to interference.

In our studies we can neglect the intra piconet overhead but we must extend the model to take into account the overhead associated with bridging. We assume that for each additional piconet that a node participates in, the switching between piconets reduces the node capacity by B, the bridging overhead. The amount of B depends on the specifics of the bridging method which are not discussed here. However, it can be expected that the overhead is determined by the amount of time needed to perform a single piconet switch, and by the frequency of switching piconets. Increasing the frequency of piconet switching will necessarily increase the bridging overhead.

Finally, we use the following simple model to capture the effects of interference. For each piconet, we first determine the set $\mathcal{I}$ of other piconets that cause interference. Here we use the simple criterion that the interfering piconets are those whose master is less than $2R$ away from the master of the piconet under consideration. Let $\mathcal{I}_i$ denote the set of piconets that interfere with piconet i. The traffic in piconet j is T^j, which means that there is a probability T^j/C that there is transmission in a given time instant. The probability of using the same frequency in piconet j as in the considered piconet is 1/79, so the probability of loss due to interference from another piconet j is $T_j/(79C)$ in our model. A packet transmission in a given piconet i is successful if there is no interference from any of the piconets in $\mathcal{I}_i$. The rate of losses due to interference is computed as $1 - \prod_{j \in \mathcal{I}}(1 - \frac{T_j}{79C})$. For a given traffic flow, we compute the loss rate for all of the links in the flow, and decrease the offered traffic to each downstream link by the amount of losses. This simple interference model is not expected to be accurate but it is able to capture the frequency hopping nature of Bluetooth and to take into account the amount of traffic in the interfering piconets.

3.3 System Parameters and Performance Metrics

Our primary goal is to relate the performance metrics of the network to the parameters of the scatternet. The set of scatternets that can be formed is determined by the constraints of our model: the spatial distribution of nodes and the range of the radio. Besides the scatternet parameters, the performance of the network is affected by our assumptions on the network configuration, that is, the amount of bridging overhead and the traffic matrix.

See Table 2 for the scatternet parameters that we have defined.

For a given set of nodes and offered traffic, we define three networks for our performance investigations where the first two are used as reference cases. First, we define the so called *visibility graph network* which consists of all the nodes and has all the links (modeled by the Bluetooth link model) between all pairs of nodes that are in radio coverage to each other. This corresponds to the case that all possible links can be used in the network. Second, we define the *scatternet graph network* which is defined by the Bluetooth links that are setup from the set of all potential links. Third, we assign master-slave roles in the scatternet graph network to obtain the scatternet. In this scenario the bridging overhead and interference effects are also taken into account.

For a given scatternet setup, we can analyze the performance metrics of scatternet compared to the performance metric of the visibility graph. We define *scatternet efficiency*

$$\Theta = T_s/T_v.$$

where T_s and T_v are the traffic carried on the scatternet and the visibility graph, respectively, where the matrix of offered traffic is the same. Scatternet efficiency expresses the change in carried traffic in a scatternet as opposed to using all the possible links with the same capacity but not counting any Bluetooth-related losses (interference, bridging overhead and the fact that we do not have all the links available).

4 Scatternet Generating Algorithms

In what follows we will describe the heuristic algorithms that we used to generate scatternets with desired properties. These algorithms are randomized and are theoretically able to generate all possible scatternets for a given set of nodes and traffic matrix. However, generating all possible scatternets would be computationally infeasible and hence we took a statistical approach by taking scatternets with a wide range of scatternet parameters in order to have a statistically representative sample. We recall that this approach is motivated by the goal of finding relations between scatternet parameters and performance metrics due to the infeasibility of analytical topology optimization.

Our algorithms generate scatternets in two steps. The first step consists of building the scatternet graph. The scatternet graph is defined by its edges, that is, the connections that are established between pairs of Bluetooth devices. Master and slave roles are assigned to nodes in the second step. In the subsections that follow, we first define two algorithms that generate scatternet graphs. The first algorithm is independent of the traffic

Parameter	Notation	Remark	Equations
Visibility degree of node i	ρ_i	the number of nodes in radio range to node i	
Mean visibility degree	$\bar{\rho}$		$\bar{\rho} = \frac{1}{N}\sum_{i=1}^{N}\rho_i$
Connectivity degree of node i	ϱ_i	the number of Bluetooth links of node i	
Mean connectivity degree	$\bar{\varrho}$		
Number of potential links	L	the total number of links allowed by the radio coverage	$L = 1/2\sum_{i=1}^{N}\rho_i$
Number of masters	N_M	piconets notation $p^1, \ldots, p^{N_M}$	
Piconet density	ξ	number of piconets divided by the total number of nodes	$\xi = \frac{N_M}{N}$
Piconet size of piconet	n^j	number of nodes participating in piconet j including the master	
Average piconet size	$\bar{n}$		$\bar{n} = \frac{1}{N_M}\sum_{j=1}^{N_M} n^j$
Piconet membership count	m_i	number of piconets node i is member in	
Average piconet membership count	$\bar{m}$		$\bar{m} = \frac{1}{N}\sum_{i=1}^{N} m_i$
Number of Bluetooth links	l		$l = N\bar{m} - N_M = N_M\bar{n} - N_M, \bar{m} = \xi\bar{n}$
Bluetooth link coverage	ν	fraction of radio links that are built up as a Bluetooth link	$\nu = \frac{l}{L} = 2\frac{\bar{m}-\xi}{\bar{\rho}} = 2\frac{\xi(\bar{n}-1)}{\bar{\rho}}$
Mean node distance	$\bar{d}$	the average of the shortest path over all node pairs	
Standard deviation of node degrees	σ		$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\varrho_i - \bar{\varrho})^2}$

Table 2: Scatternet parameters

matrix, which means that it does not take into account traffic flows when building the scatternet. This algorithm will be referred to as Traffic Independent Scatternet Graph (TISG) algorithm. The second algorithm is called Traffic Dependent Scatternet Graph (TDSG) algorithm and it selects the links of the scatternet graph taking into account the amount of traffic flowing on the links. Finally we describe an algorithm that assigns master and slave roles to nodes in an established scatternet graph.

4.1 TISG Algorithm

The TISG algorithm proceeds as follows. We select edges of the scatternet graph one by one out of the possible links (i.e. between the pair of devices within radio range). The total number of edges, l, is a parameter of the algorithm, and must be $N - 1 \leq l \leq L$. The next edge is chosen randomly out of all potential edges that are not yet chosen, according to a weight, defined below. For each potential edge AB we define two metrics.

The *P metric*, $\phi_{P,AB}$, for a potential edge characterizes the length of the shortest alternative path between the two end-points of the potential edge in the current scatternet graph. For two nodes which are not yet connected in the scatternet graph, $\phi_P = \infty$.

The *D metric*, $\phi_{D,AB}$, for a potential edge characterizes the degree of the two endpoints, $\phi_{D,AB} = \max\{\pi_A, \pi_B\}$, where π_A and π_B are the degrees of the two endpoints of the potential edge in the current scatternet graph.

For each of the potential new edges of the scatternet graph, we derive the weight of edge AB from $\phi_{P,AB}$ and $\phi_{D,AB}$ as follows.

$$w_{P,AB} = (\alpha_P)^{\phi_{P,AB} - \bar{\phi}_P} \quad (4)$$

$$w_{D,AB} = (\alpha_D)^{\phi_{D,AB} - \bar{\phi}_D} \quad (5)$$

$$w_{AB} = w_P w_{P,AB} + w_D w_{D,AB} \quad (6)$$

Here, $w_{P,AB}$ and $w_{D,AB}$ characterize the relative weight of the length of the shortest alternating path and maximum end node degree corresponding to the edge, respectively. $\phi_{P,AB}$ and $\phi_{D,AB}$ are compared to their mean over all potential edges, $\bar{\phi}_P$ and $\bar{\phi}_D$, so that the weights will reflect the relative value of these metrics compared to other edges. We apply an exponential function with base α_P and α_D to the metrics so that edges of high metric will get a considerably high probability of getting chosen.

The weight of edge AB is computed from the linear combination of $w_{P,AB}$ and $w_{D,AB}$. The constants w_P and w_D determine the relative importance of the P and D metrics in the selection of the new edge.

The selection rule is that a new potential edge is chosen randomly from all the potential edges according to its weight w_{AB}. However if there is at least one potential edge with $\phi_{P,AB} = \infty$ (i.e., there is no alternate path, A and B are not yet connected), then the new edge is selected from such edges only according to the weight $w_{AB} = w_D w_{D,AB}$. This means that first we pick edges that form a spanning forest in the scatternet graph.

Note that by selecting a new edge in the scatternet graph, we change the P and D metrics for the other potential edges. The D metric of a potential edge is increased for the edges having a common node with the newly chosen edge. In the case of $\alpha_D > 0$, this gives us a positive feedback loop and tunes the algorithm in favour of nodes with high degree.

4.2 TDSG Algorithm

In this case we take the offered traffic into account when building the scatternet graph in the following way. We take the visibility graph network and introduce the offered traffic to it using our Bluetooth network and link model. This models the case that all possible links are set up in Bluetooth. Let us define the *T metric* for an edge AB as the amount of throughput that is achieved on the link, i.e., $\Phi_{T,AB} = t_{f,AB} + t_{r,AB}$.

We build the scatternet graph similarly to section 4.1 and in addition we introduce the weight

$$w_{T,AB} = (\alpha_T)^{\phi_{T,AB} - \bar{\phi}_T}.$$

We now compute the weight for a new potential edge as

$$w_{AB} = w_P w_{P,AB} + w_D w_{D,AB} + w_T w_{T,AB}$$

In this way, we can give higher weight to those potential edges that carry more traffic in the visibility graph, and this can increase the likelihood of selecting edges that are more advantageous for the current traffic flows.

4.3 Assigning Master and Slave Roles

Once we have created the scatternet graph, we define the master and slave roles. We assign master and slave roles to the two end-nodes of each edge of the scatternet graph. If the master and slave roles are defined for an edge, we say that this edge is directed from the master to the slave. If the roles are not yet defined for an edge then it is undirected. To determine the directions of the edges, we pick the masters (and accordingly the piconets) one by one as follows.

We pick a new master from the nodes which have at least one undirected edge (i.e., undefined master-slave role). This new master defines a new piconet. We direct all undirected edges of the new master towards the node on the other end of the edge. This defines a new piconet and its slaves. We pick new masters again and again until all edges become directed.

When selecting the new master, we assign a weight to each node which has at least one undirected edge as

$$w_N = (\alpha_M)^{\phi_{U,N}}$$

where $\phi_{U,N}$ is the number of undirected edges for node N, and α_M is a constant that determine the relative importance of the number of undirected edges in selecting the new masters. We pick the next new master randomly out of the nodes which have at least one undirected edge, according to weight w_N.

5 Numerical Results

In what follows, we describe the simulation environment followed by a number of case studies with numerical results. All of these studies focus on the relationship between certain scatternet parameters and performance metrics. First, we investigate the performance impact of the number of links in a scatternet. Next, we compare scatternets consisting of different numbers of piconets while the total number of links is kept constant. Following this we attempt to capture the impact of scatternet topology by looking at the effect of node degrees. Then, we study scatternets whose topology is adjusted to the carried traffic and compare their performance to traffic independent scatternet topologies. Finally, we investigate scatternet efficiency in the function of the bridging overhead.

5.1 Simulation Setup

Each of our studies start by randomly generating N nodes in a square area of size D. The nodes are uniformly spread in the area and are stationary during simulation. After generating the nodes we determine the visibility graph. Two nodes are connected in the visibility graph if their distance is less then R which is now set to be 10 m. Next, we build the scatternet graph using one of the algorithms described in Sections 4.1 or 4.2 and assign master and slave roles using the algorithm of Section 4.3.

Once the scatternet network is complete we generate a random traffic matrix that will represent the load in the network. We assume traffic flows to be fluid, which means that we do not model individual packet transmissions and traffic flows are infinitely divideable. We take the pairs of nodes one by one and add the offered traffic to the scatternet network. When adding a new flow to the network shortest path routing is used to find the route between source and destination. We determine the amount of traffic that can actually be carried over the network by calculating the resources that are needed for the new flow, taking into account our Bluetooth link and network model of Section 3, as well as the bridging overhead B. We use the following iterative algorithm to introduce the traffic into the network. New traffic is added by a quantum of τ. This means that in one iteration we introduce at most τ between each pairs of source and destination nodes, and then iterate again to add another quantum of at most τ. By specifying a small value of τ such as 1 kbps, we ensure that it is possible for many flows to share a given link, i.e., one flow will not starve the others. We have found that in typical cases a value of $\tau = 1$ kbps is an appropriate choice which avoids unfairness and achieves high network throughput.

For illustrative purposes we plotted a scatternet sample generated by our algorithm in Figure 4. Edges are

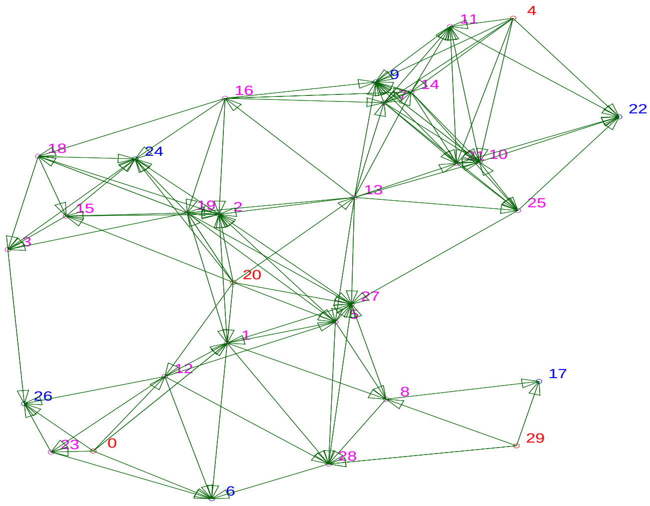


Figure 4: An example scatternet network

directed from the master to the slave.

5.2 Impact of Number of Links

In what follows we investigate the performance impact of the number of links in a scatternet. One can expect that, for any given set of nodes and traffic matrix, increasing number of links can carry increasing amount of traffic. However, in Bluetooth increasing number of links might also require increasing number of bridging nodes which represents increasing overhead. Our goal is to analyze these phenomena.

We will study the impact of the number of links using a "homogeneous" traffic matrix, that is, a situation where each node offers a traffic flow of T kbps toward each other node. We set T such that the offered traffic be considerably higher than the scatternet's capacity.

A set of simulation results are illustrated in Figure 5 (a) where each dot represents the simulation result gained for one scatternet. These scatternets all used the same set of nodes and the same traffic matrix but had different numbers of links. The horizontal axis represents ν, the Bluetooth link coverage while on the vertical axis we plotted the total amount of carried traffic.

In accordance with expectations, as we start from very small link coverage, an increasing number of links can carry increasing amount of traffic. This is due to the fact that higher number of links correspond to more piconets (Figure 5 (d)). Another aspect of the same phenomenon is illustrated in Figure 5 (c) where the hop count is illustrated as a function of the link coverage. Increasing number of links decreases the mean amount of hops from source to destination. However, as the link coverage exceeds 0.3 the carried traffic begins to decline. We attribute this phenomenon to the fact that increasing the number of links means that nodes have to become members in an more and more piconets and this increases the bridging overhead.

The amount of carried traffic is a good measure of the performance of the scatternet network but it has a disadvantage that it depends on our traffic model. Next, we introduce a rough approximation to the system capacity in order to estimate the system performance without the need to introduce traffic to the network.

The capacity of a single piconet is estimated by $\max(C - (\bar{m} - 1)B, 0)$, i.e., each membership in another piconet decreases capacity by B. This is multiplied by the total number of piconets to obtain the capacity of the whole network, and is divided by the mean hop count between nodes to take into account the fact that a single traffic flow takes the resources on many links:

$$\Psi = N_M \max(C - (\bar{m} - 1)B, 0)\frac{1}{\bar{d}}$$

This is of course only a very rough approximation of the system capacity which in reality depends on the actual scatternet graph, the exact master and slave roles, the distribution of traffic sources and sinks, the position of the devices and the interference. The purpose of defining Ψ is not to get an accurate estimate of the amount of carried traffic, but to have a traffic model independent metric which follows the same trends as the amount of carried traffic given a set of traffic sources and sinks.

Figure 5 (b) shows that it is indeed the case. Here we plotted the estimated capacity (Ψ) versus the amount of carried traffic. It shows that although the two metrics are not proportional, nevertheless Ψ increases monotonously as the amount of carried traffic increases. The point where Ψ is maximized is marked by a vertical line in Figure 5 (a): it shows that maximizing Ψ selects a network which is close to the optimal one in terms of carried traffic in this configuration.

Let us now use a more realistic traffic model based on a client-server assumption. In this model we randomly select 10 nodes that represent servers and to each server we select 4 clients. Servers offer a load of T kbps to each of their clients. Simulation results using this model are presented in Figure 6 (a). Compare the results of Figure 5 (a) with Figure 6 (a): with the server-client traffic model, the amount of carried traffic is significantly less, and the increase of link density causes the amount of carried traffic to drop to a very low value. However, as seen in Figure 6 (b), there is again a monotonously increasing relation between Ψ and the carried traffic, and so maximizing Ψ again yields a nearly optimal configuration.

5.3 Impact of Number of Piconets

In Figure 7 we investigate the effect of changing the number of piconets while the number of links is kept constant. Figure 7 (a) shows results obtained for relatively large networks (in terms of number of nodes, $N = 40$) while in Figure 7 (b) we have plotted results for small networks ($N = 8$). The number of piconets is varied by changing the parameter α_M. Increasing this parameter increases the probability of choosing nodes with higher degrees as masters (see Section 4.3), which has the effect of decreasing the number of masters needed to direct each edge in the scatternet graph, and in this way decreases the number of piconets. Looking at the figure one can observe a decreasing trend in the carried traffic as the number of piconets increases. A scatternet that consists of a larger number of piconets, given the same number of links, will contain nodes that

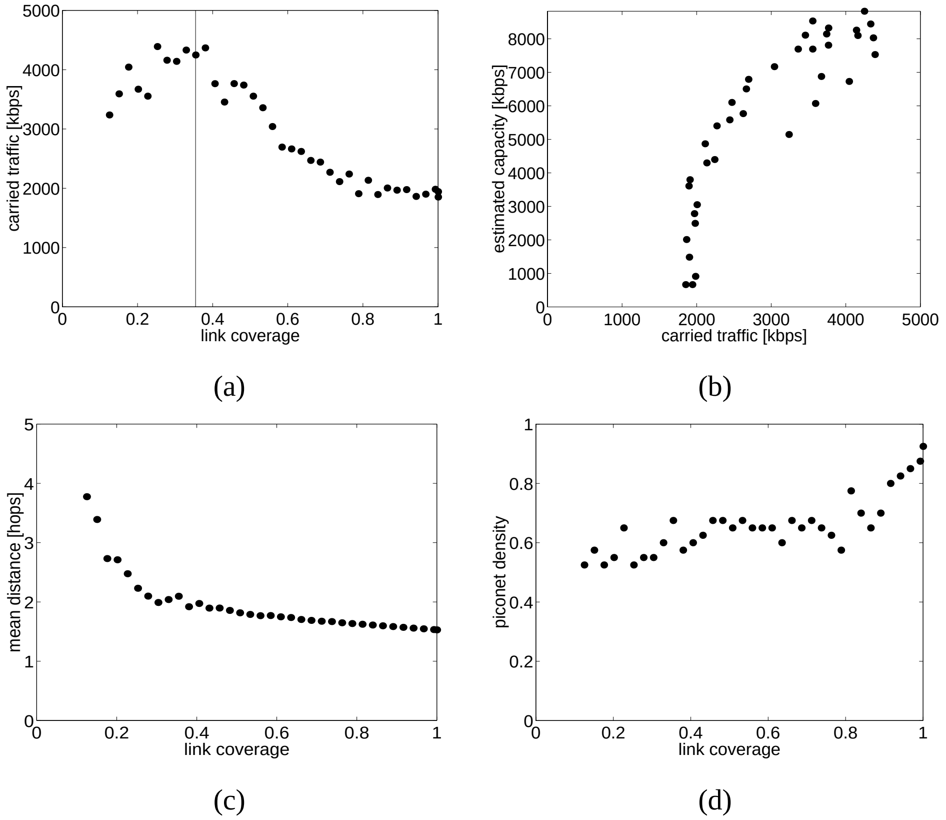


Figure 5: Effect of changing the number of links. Parameters: $N = 40$, $D = 20m$, $T = 1000$, Traffic: homogeneous, $B = 100$, $w_P = 0.5$, $\alpha_P = 2$, $w_D = 0.5$, $\alpha_D = 2$, $w_T = 0$, $\alpha_T = 5$, $\alpha_M = 2$

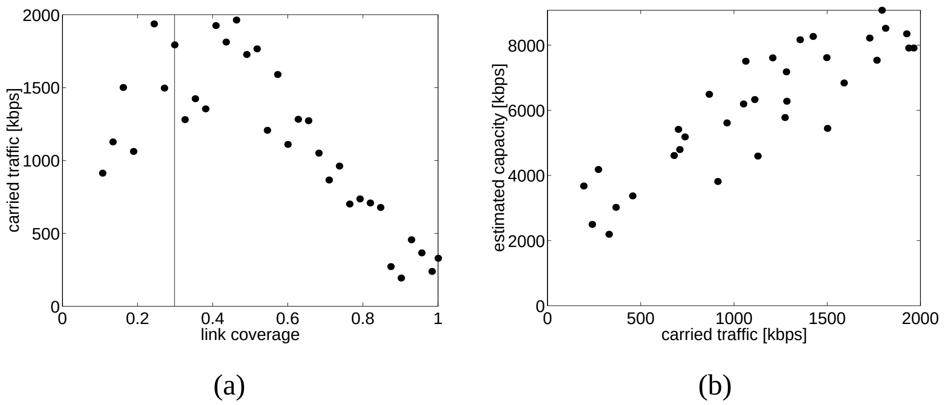
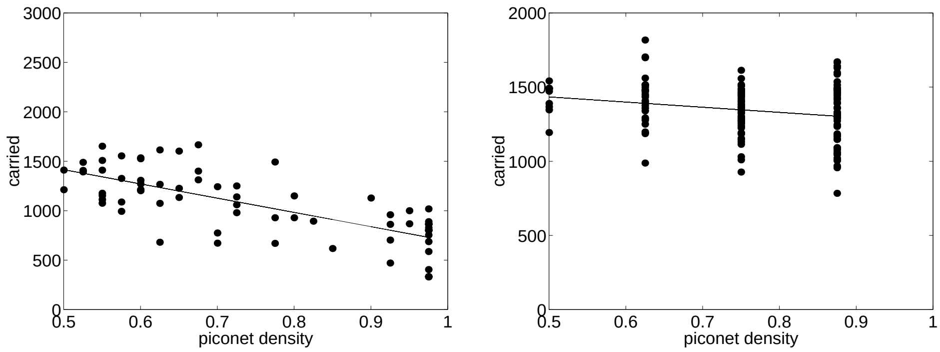
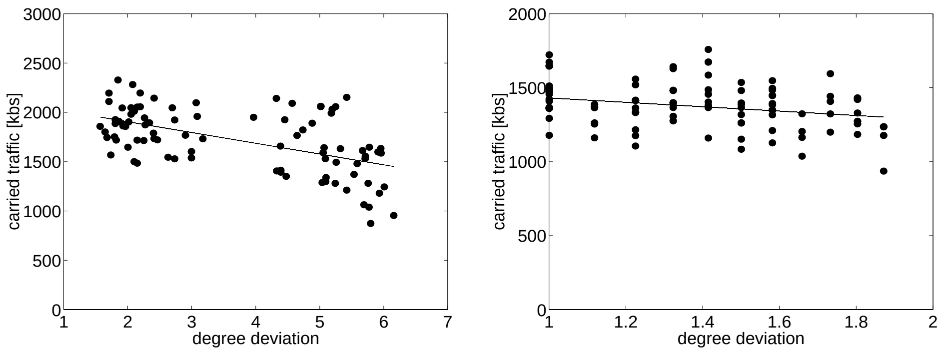


Figure 6: Effect of changing the number of links. Parameters: $N = 40$, $D = 20m$, $T = 1000$, Traffic: SERVER-10-CLIENT-4, $B = 100$, $w_P = 0.5$, $\alpha_P = 2$, $w_D = 0.5$, $\alpha_D = 2$, $w_T = 0$, $\alpha_T = 5$, $\alpha_M = 2$



(a) (b)

Figure 7: Effect of changing the number of piconets for large and small networks. Parameters: $N_a = 40,\ N_b = 8$, $D_a = 20m,\ D_b = 10m$, $T = 1000$, SERVER-10-CLIENT-4, $l_a = 100,\ l_b = 16$, $w_P = 1$, $\alpha_P = 2$, $w_D = 1$, $\alpha_D = 2$, $w_T = 0$, $\alpha_T = 6$, $\alpha_M = 0.25 \ldots 3$



(a) (b)

Figure 8: Effect of changing the degree deviation of nodes, for large and small networks. Parameters: $N_a = 40,\ N_b = 8$, $D_a = 20m\ D_b = 10m$, $T = 1000$, Traffic: SERVER-10-CLIENT-4, $B = 100kbps$, $l_a = 100,\ l_b = 16\ w_P = 1$, $\alpha_P = 2$, $w_D = 1$, $\alpha_D = 0.5 \ldots 4$, $w_T = 0$, $\alpha_T = 6$, $\alpha_M = 2$

necessarily belong to a higher number of piconets at the same time. This implies higher bridging overhead which, in turn, results in a loss of capacity. We also note that this phenomenon is more significant in large networks than in small ones.

5.4 Impact of Degree Deviation

In Figure 8 (a) and (b) the carried traffic is plotted as a function of the standard deviation of node degrees for large and small networks, respectively. As we consider scatternets with higher deviation of node degrees the topology resembles more to a “star” like topology with a few central nodes where most of the traffic flows go through. These central nodes easily become the bottlenecks in the network which reduces the throughput. On the other hand small deviation refers to a topology where on each node approximately the same number of traffic flows go through, which allows a more even distribution of traffic in the network. We can observe that

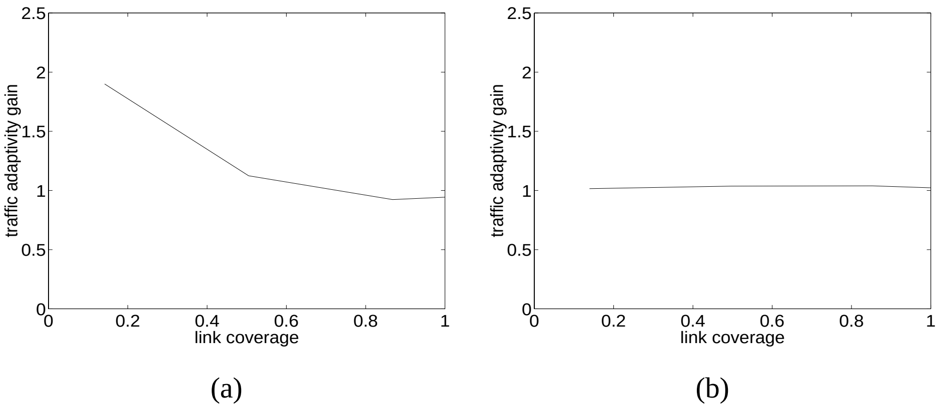


Figure 9: Comparison of traffic dependent ($w_P = 0, w_D = 0, w_T = 1$) and traffic independent ($w_P = 0.5, w_D = 0.5, w_T = 0$) scatternet graphs. Parameters: $N = 40$, $D = 30m$, $T = 1000$, Traffic: SERVER-10-CLIENT-4 (a) homogeneous (b), $B = 100$kbps (a,b), $\alpha_P = 2$, $\alpha_D = 0.5 \ldots 4$, $\alpha_T = 6$, $\alpha_M = 2$

the performance of the network is better with lower degree deviation, but again this effect is less significant for small networks.

5.5 Comparing Traffic Independent and Traffic Dependent Scatternet Graphs

We have compared the traffic dependent ($w_P = 0, w_D = 0, w_T = 1$) and traffic independent ($w_P = 0.5, w_D = 0.5, w_T = 0$) scatternet graphs. Figure 9 shows the ratio of the carried traffic in the traffic dependent over traffic independent scatternet graphs. Figure 9 (a) shows that traffic dependent scatternet graphs can improve performance by up to a factor of 2 in the case of low link coverage when the SERVER-CLIENT type traffic model was used. In the case of homogeneous traffic, the traffic dependent algorithm does not help since all links carry traffic.

5.6 Scatternet Efficiency and Bridging Overhead

Finally we investigate the scatternet efficiency in the function of bridging overhead. Scatternet efficiency shows the overhead of Bluetooth scatternet formation. We have plotted it in Figure 10 (a) and (b) for a large and a small network respectively, as a function of bridging overhead B. Both traffic independent and traffic dependent cases were considered, and we have plotted the maximal scatternet efficiency that can be achieved by varying the number of links in the network.

We can observe that bridging overhead has a major impact on scatternet efficiency; this suggests that efficient ways of switching between Bluetooth piconets are fundamental for good scatternet performance. The figure also shows that traffic dependent algorithms can increase efficiency and the amount of possible improvement significantly increases when the bridging overhead is high. Note that for the small network case at the bridging overhead of $B = 50$ kbps traffic dependent scatternet formation does not improve efficiency. This is because for small networks the maximal efficiency is achieved when all the links are used.

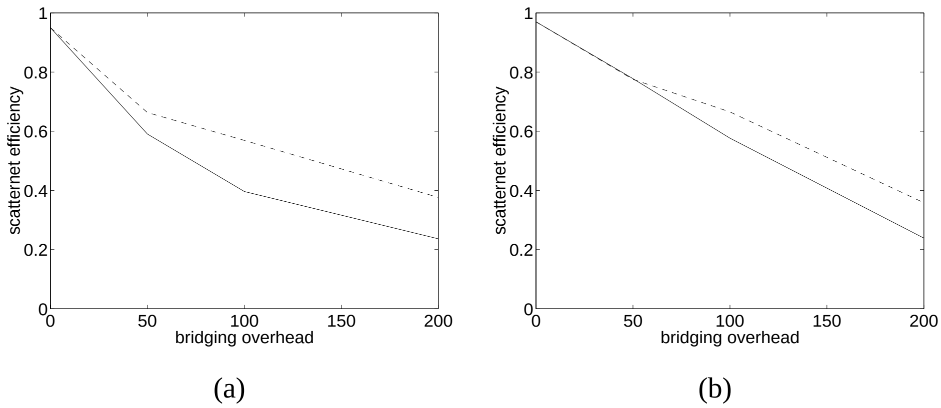


Figure 10: Scatternet efficiency versus bridging overhead for traffic dependent ($w_P = 0, w_D = 0, w_T = 1$) and traffic independent ($w_P = 0.5, w_D = 0.5, w_T = 0$) scatternet graphs. Parameters: Parameters: $N_a = 40$, $N_b = 8$, $D_a = 20m$, $D_b = 10m$, $T = 1000$, Traffic: SERVER-10-CLIENT-4, $B = 100kbps$, $\alpha_M = 2$

6 Conclusions

We have investigated the performance of using Bluetooth devices to build ad-hoc networks through a statistical approach. For this, we have established a Bluetooth network model and studied the relationship between network characteristics and the performance of the network.

Due to the statistical nature of our investigations, the results show a high degree of variance, but at the same time we have been able to find clear trends. We have found that two characteristics, namely the amount of bridging overhead and the number of established Bluetooth links, have a major impact on system performance. Namely, there is a link number where the throughput is maximized. This implies that for good performance, it is fundamental to decrease bridging overhead as much as possible, and that the number of established Bluetooth links has to be well controlled by any scatternet formation procedure. We have also given a traffic-independent guideline on how to set the number of links in the network and we have demonstrated that it gives good performance.

We have found that other aspects of scatternet formation, namely traffic dependency, number of piconets (given the number of links), and the standard deviation of node degrees also affect performance but this dependency is less significant compared to the impact of bridging overhead and of the number of links. In this latter case study we introduced the notion of scatternet efficiency and we have shown that it is highly dependent on the characteristics of the network. This motivates future research on the scatternet forming Bluetooth procedures.

References

[1] "Bluetooth Baseband Specification Version 1.0 B," *http://www.bluetooth.com/*

[2] J. Haartsen, "BLUETOOTH - the universal radio interface for ad-hoc, wireless connectivity," *Ericsson Review*, No. 3, 1998.

[3] Mobile Ad-hoc Networks Working Group of IETF, *http://www.ietf.org/html.charters/manet-charter.html*

[4] N. Johansson, U. Körner, P. Johansson, "Performance Evaluation of Scheduling Algorithms for Bluetooth," *Proceedings of IFIP TC6 Fifth International Conference on Broadband Communications'99*, Hong-Kong, November 10-12, 1999.

[5] P. Johansson, N. Johansson, U. Körner, J. Elg, G. Svennarp, "Short Range Radio Based Ad-hoc Networking: Performance and Properties," *Proceedings of ICC'99*, Vancouver, 1999.

[6] S. Zürbes, W. Stahl, K. Matheus, J. Haartsen, "Radio Network Performance of Bluetooth," *to appear in the Proceedings of ICC'2000*, New Orleans, 18-22 June, 2000.

[7] M. Kalia, D. Bansal, R. Shorey, "MAC scheduling and SAR policies for Bluetooth: a master driven TDD Pico-cellular wireless system," *Proceedings of Mobile Multimedia Communications*, 1999.

[8] P. Bhagwat, A. Segall, "A Routing Vector Method (RVM) for Routing in Bluetooth Scatternets," *Proceedings of Mobile Multimedia Communications*, 1999.

[9] P. Bhagwat, A. Segall, "IP Services over Bluetooth: Leading the Way to a New Mobility," *Proceedings of Local Computer Networks Conference*, 1999.

[10] Charles Perkins, editor, "IP Mobility Support," *Internet RFC* 2002, October 1996.

[11] A. Valkó, A. Campbell, J. Gomez, "Cellular IP," *Internet Draft*, draft-valko-cellularip-00.txt, Work in Progress, November 1999.