

On the Impact of Inter-Piconet Scheduling in Bluetooth Scatternets

Manthos Kazantzidis, Mario Gerla
UCLA WAM Lab

Abstract- The current Bluetooth specification defines the notion of interconnected piconets, called scatternets, but does not define the actual mechanisms and algorithms necessary to set up and maintain them. A scatternet will typically be set up in an ad-hoc fashion to interconnect piconets. For instance, a typical personal area network (PAN) would comprise of a user's wearable and portable electronic devices in a piconet formation, while scatterneting enables networking between many PANs. A PAN interconnect will serve its own traffic as well as traffic from the Internet. The inter-piconet Bluetooth units, i.e. the gateways interconnecting the piconets in a scatternet, need to time-division multiplex their presence in each of their piconets. The time division in gateways effectively limits the network capacity and may introduce bottleneck points in the network. Therefore an inter-piconet scheduling (IPS) algorithm is required in order to efficiently coordinate the inter-piconet Bluetooth units and maximize the network capacity. This may be achieved, for example, by using rendezvous points –pre-negotiated slots that a piconet master agrees to poll the gateway, and the gateway agrees to have switched to that master's frequency hopping sequence and accept the poll. The impact of the rendezvous IPS algorithm to the overall scatternet is studied in this paper. We consider this an important question because the inter-piconet scheduling functions may require support from low layers usually implemented in silicon [2]. We assume a distributed approach is required and evaluate performance of a range of different proposed rendezvous scheduling algorithm. The difference of our approach to existing work is that our algorithms are based on more information exchange between nodes. While this creates higher traffic and memory overhead it allows for more efficient scheduling of the scarce bandwidth. On one extreme we have a minimum intelligence/silicon space totally random assignment and on the other a maxmin optimum forwarding throughput assignment. While we are convinced that random assignments have scalability advantages, crucial for some applications esp. those of embedded systems, we study more silicon-intelligent approaches that may prove necessary for multimedia applications over Bluetooth links. Audio and video streaming is bandwidth consuming, usually spans considerable time lengths, and generally requires a steady network response due to its real time nature. The maxmin optimum algorithm is shown by simulation experiments to have a total network throughput of more than a 30% than simpler non-random solutions and without creating bottleneck gateways, and at least 2 times better than random schemes. This improvement may be considered important to some applications, especially considering the Internet typical client data rates, but comes at the cost of a higher design complexity/cost.

I. INTRODUCTION

The Bluetooth wireless technology is a low power, low cost, short range RF technology that focuses on replacement of cables between electronic devices. The nodes in a Bluetooth network are organized in piconets, a star formation of up to 8 time-slot and frequency synchronized nodes. Bluetooth piconets allow for multipoint communication while scatternets extend the

functionality by further interconnecting piconets [1]. This enables a wide range of applications ranging from single personal area networks (PAN) to larger ad hoc networking environments e.g. sensors, large events, conferences etc. Scatternets are supported through the designation of nodes that may synchronize to more than one piconets (masters), and therefore may receive and send data between them. Any node may become a gateway if it senses the presence of a foreign master node, through the INQUIRY/PAGING procedure [1]. The gateway nodes then time-divide their presence in the piconets. This scatternet arrangement presents an interesting situation, where slave nodes in a piconet may exchange data at full link bit rates [12], taking advantage of the frequency hopping pattern frequency reuse, while nodes in different piconets may communicate at a fraction of this rate and variant and increased delays, as defined by the presence intervals of gateway nodes in the path.

Most of the published work on Bluetooth so far has focused on the single piconet and issues related to scheduling of units within one piconet, e.g., [13] [15]. Some work has also looked at the task of actual formation of Bluetooth piconets and scatternets, [3], [14], [15]. In a scatternet resulting from any formation algorithm, the inter-piconet units still need to be scheduled in an efficient way. Thus, the inter-piconet scheduler (IPS) is crucial for the operation of scatternets. Few pieces of work on inter-piconet scheduling issues in scatternets can so far be found in the literature. In [16] an inter-piconet scheduler is analyzed for a case where central control of the entire scatternet is assumed, which is costly to achieve. The work herein will focus on analyzing inter-piconet schedulers that operate in a distributed manner in scatternets and thus eliminate the need for scatternet-wide coordination. Recently [19] has proposed a pseudo-random algorithm for IPS that is based on check-points (similar to rendezvous-point but random). In the same paper, a theoretical algorithm is described called Ideal Coordinated Scatternet Scheduling (ICSS). The pseudo random algorithm is undoubtedly advantageous in situations where scalability is important and sustained bandwidth of lesser importance, as in non-real time TCP communication. Even though we cannot test these algorithms (the simulation models used in that Ericsson paper were not available to us) our results are in accordance with them. In fact ICSS, seems to have very similar performance to our RV-maxmin optimal forwarding throughput algorithm. RV-maxmin deals with practical optimality issues by dealing with the forwarding throughput equations that allow an efficient local optimization. [20] is another algorithm that recently emerged based on rendezvous points. It is based on pseudo-random rendezvous windows and focuses on minimizing RV-point collisions. Unfortunately there is no presentation of any

performance results and therefore comparison with that algorithm is not possible.

Some architectures that have been proposed already ([4] and [10]) select the Gateways and attempt to create the links on application and routing demand, but in order to test the IPS we attempt all possible gateway connections under imposed limits of number of gateways per piconets (GP) and number of piconets per gateway (PG). We are studying the IPS performance rather than the selection of the Gateways per se. It is a more challenging problem of IPS when all the possible connections are attempted.

How should then gateways across the scatternet adjust and coordinate their presence intervals to each piconet they act as forwarders? Ideally a gateway can forward traffic at half the link bandwidth (if no traffic has the gateway as destination, it ideally spends half its time receiving traffic from one master and the other half forwarding it to the other master). However, what are the best slots for a gateway to rendezvous with each piconet master, given that each piconet has different traffic requirements, different number of slaves and different gateways present? In order to understand the impact of this decision, first consider a totally random slot selection. The presence interval may then be arbitrarily small and therefore may result in wasted slots, very low data forwarding rates and highly variant delays. A more intelligent inter-piconet scheduling (IPS) is therefore needed. Now consider a simple algorithm that plainly equally distributes the gateways presence among the piconets, as proposed in [18], the maximum distance rendezvous point allocation algorithm. It ensures that the presence interval as large as it can be but the forwarding data rate may still be very low. Consider, for example, when a gateway switches during very heavy traffic from other nodes or when all other gateways are also present during the same time. This may happen e.g. when a five-node piconet has three gateways and all decide to switch in the piconet in successive slots. Then, there will be a very busy interval in which the master will have to exchange traffic with all four nodes and an underutilized interval that the master may only exchange data with the one slave. In a network that this occurs often, throughput will be low. In this paper, we are presenting a rendezvous point based IPS algorithm that coordinates so that this occurs minimally, and maximizes overall network throughput by maximizing the forwarding throughput between two adjacent piconets. It represents an improvement of approximately 30% over the maximum distance scheme, while avoiding the creation of bottleneck gateways.

An efficient way to define those presence intervals is through the establishment of rendezvous points as for example supported by the architecture proposed in [2]. The rendezvous point is a pre-arranged slot that a master agrees to poll a gateway node, and the gateway node agrees to accept the poll. The master decides after how many slots it will poll the gateway and lets the gateway know. The gateway then arranges to switch back right before that poll slot. In this way poll slots are not wasted. After establishment of a rendezvous point it may be periodically honored for the duration of the link. Rendezvous points are supported in the Bluetooth specification [1] by using the SNIFF primitive [7]. A gateway may accept the poll if it is currently time-slot and frequency synchronized to the master's hopping pattern. Therefore a gateway node is able to keep track of frequency-hopping sequences for all masters it wishes to communicate with. Since different piconets are not time slot-

synchronized either, up to one full slot is lost whenever a gateway switches from one piconet to another. Furthermore, if its clock has skewed more than this slot timing difference, the switch may completely miss the poll slot (the rendezvous point). Clock skew (in relation to the piconet master clock) may occur during the absence of the gateway node from the piconet, and therefore absence intervals should not be large. Large absence intervals are also cause for larger end-to-end delays. Therefore, there is a trade-off in deciding how often to switch. The overhead of one lost slot per switch directs towards infrequent switches, whereas the clock skew and end-to-end delay constraints direct toward frequent switches.

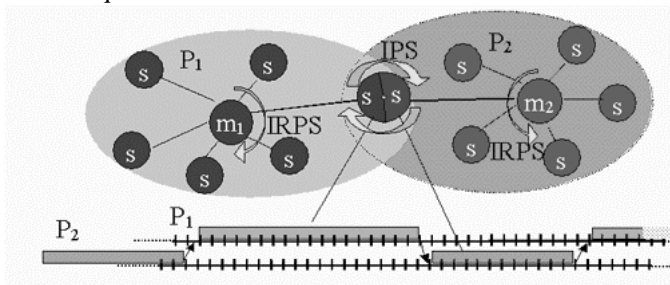


Fig. 1 A scatternet with one inter-piconet unit that divides its time between the two piconets

The paper is organized as follows. In section II, we define the general problem including the calculation of forwarding throughput of a gateway between two piconets, and in III the allocation schemes. In IV we describe the simulation models and in V the results. We conclude in VI.

II. BACKGROUND

Bluetooth nodes perform periodic INQUIRY/PAGING in order to sense the existence of other nodes. When a master senses a new node they exchange configuration parameters so that the node may join the piconet and vice versa [1]. When a node already belongs other piconet(s) (as e.g. the home piconet, the piconet it first joined) it may join the new piconet and become a gateway between the two (or more). In a rendezvous point based IPS, if the node was not a gateway previously, two rendezvous points need be established, otherwise one new point for the new piconet is enough. Whether a node should actually attempt to become a gateway or not, is decided by a gateway selection algorithm as in [3], [5]. Existing rendezvous points of the new master and the gateway may be exchanged during this gateway establishment phase, so that the new rendezvous slot does not conflict with others. As discussed before scalable and simple random algorithms have to tolerate RV-point collisions. Our algorithm uses this information, not only to choose non-conflicting new rendezvous point(s) but also to decide the optimum new rendezvous point(s) (by optimum we refer to the best possible given the existing topology, i.e. without re-arranging any previously allocated rendezvous slots). Note that the establishment of a new rendezvous point in a visiting piconet, does not only affect the forwarding throughput of the (home, visiting) piconet pair. It also affects the forwarding throughput of all connections of the visiting piconet. The algorithm picks the point so that the minimum forwarding throughput (f.t.) from the visiting piconet to any of its connected piconets is maximum. In this way bottleneck links are not created in the network and overall forwarding throughput is close to optimal. We also experiment with a greedy algorithm –one that maximizes the f.t

between visiting and home piconet and found that it may create bottleneck piconet pairs. Simpler algorithms are also used in simulations that require less computation, but have worse performance and/or flexibility.

Let us assume that a number of piconets is formed and as they move within a terrain they come into proximity to each other and may form scatternets. This is for example, similar to a conference PAN scenario where one person's devices are arranged in a piconet and people move about a conference show room and communicate. A slave of piconet i , that recognizes another master's (piconet j) INQUIRY, may become a gateway between its home piconet and the visiting piconet. There may be other gateways already present in the home piconet as well as in the visiting piconet. Also, the prospective gateway node may already be a gateway (i.e. connecting its home piconet with some other piconets). Therefore, all the parameters that come into play are:

$\{M_h RV_1, \dots, M_h RV_n\}$ existing RV point s for home Master
 $\{M_v RV_1, \dots, M_v RV_m\}$ existing RV point s for visiting Master ^E
 $\{G_g RV_1, \dots, G_g RV_k\}$ existing RV point s for Gateway
 equation 1 h= home (original) piconet, v=visiting piconet, g=a gateway index, e.g. MvRV2 may be the second rendezvous point for the visiting piconet at the time of visit.

Note that we have previously defined a rendezvous point as the number of slots after which the master will poll. In order to use absolute numbers for rendezvous points (rather than relative to the establishment time) we also introduce the notion of the superframe cycle (in [20] it is called a Rendezvous Window). A superframe cycle defines the period (measured in slots) after which the rendezvous point will re-occur. Therefore all operations from now on may be assumed to be in modulo superframe arithmetic. Note that the superframe notion is logical only, and does not imply any inter-piconet coordination or synchronization. It simply defines the RV point in absolute terms and can be a pre-configured number (faster switched may still occur on-demand with the establishment of multiple rendezvous points). Say for example the superframe for a Master is 90 slots and a gateway's RV point is 30. The gateway knows after how many slots to switch at any slot t (that is $30-t$ if $t \bmod 90 \leq 30$ or $90-t+30$ if $t \bmod 90 > 30$). In an implementation the exchange of these parameters constitute the extra communication overhead that is traded off for the improved bandwidth.

Now we are ready to calculate the possible forwarding throughput (ft) from one piconet to another based on the RV points of its gateways. The calculation means to provide a way to efficiently choose a new RV-point rather than be exact and absolute in value. An intuitive depiction of the situation is attempted in Fig 2.

The fraction f that a gateway G_g spends with master M_k :

$$f_{G_g M_k} = \frac{(G_g RV_k - G_g RV_{k+1})_{\text{superframe arithmetic}}}{\text{superframe}}$$

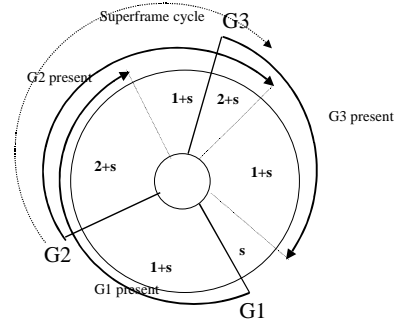


Fig 2. The number of nodes active in the polling cycle during the superframe cycle of a piconet with s non-gateway slaves and 3 gateway nodes.

The average number of nodes that actually participate in the polling cycle depends on the polling scheme used. Let us call s the number of active nodes in the polling cycle and \tilde{s} its average in the superframe. For example, in round robin non-exhaustive, if the masters know when gateways are present, it becomes:

$$\tilde{s}_{Mk} = \sum_{\forall s} f_{G_s M_k}$$

For a pure round robin scheme, for example, the fraction f of a slave that is not a gateway is 1. Let us now calculate the f.t. for a scatternet link from (adjacent) piconets i to j interfaced by a gateway g .

$$FT(i, j) = \min(FT_g(i), FT_g(j)),$$

$$FT_g(i) = \frac{b * f_{G_g M_i}}{\tilde{s}_i + \text{overhead}(\tilde{s}_i)} \quad (1)$$

In the above formula b is the Bluetooth bandwidth (1Mbps). The formula is validated using our simulation models in the next section. It can be extended for more than one hop (not adjacent piconets).

A. Validation

We validated the above equation using our GlomoSim Bluetooth model and our NS model. The part of the topology involved in our validation is shown in Fig 3. In order to calculate the f.t. of gateways we had every node shown participating in a high bandwidth (640Kbps) CBR intra-piconet or inter-piconet connections. In this way, the network is saturated, the queues fill up and the gateways, being the bottleneck points, work at their f.t.s. Gateway 10 in the experiment forwards packets from master 8 to master 24. The simulation gave us a goodput of 99Kbps for this connection. According to our equation let us first calculate the average number of active slaves in the piconet of master 8 when gateway 10 is present; that is from 80 to 20 in a 120 slot superframe. The two slaves S11 and S12 are present through the whole gateway presence interval. Since this is a pure round-robin run we count a fraction of 1 for each. A fraction of 1 is also counted for the gateway itself and 20/60 for gateway 9. This is because the G10 presence interval is 60 slots and for 20 of those, slot 80 to 100, G9 is also present. Similarly we get 20/60 for gateway 7. This gives an \tilde{s} of 3.66. Similarly for master 24 we get an average number of slaves of 3. The G10 presence fraction f is 60/120=0.5 for both piconets and so the f.t. is defined by the piconet of master 8 (with the minimum FT_i and the max \tilde{s}) as $0.5 * 721 / 3.66 = 98.3\text{Kbps}$, which is very accurate compared with the simulations!

III. RV-ALLOCATION SCHEMES

The proposed algorithm calculates the possible FTs between the visiting piconet and all piconets it connects to, as if the proposed connection had already been established. For each possible rendezvous slot it looks at the minimum among the FTs. It then keeps the point that results in the maximum of the minimum FTs. If the node has not been a gateway so far, then all slot pairs are examined and the best (maximum) pair is chosen. The logic behind it is to avoid establishing RV points that take away forwarding throughput from other piconet pairs. We also experimented with different RV assignment algorithms. We found that the overall average possible scatternet throughput with this algorithm is close to the greedy one (see 2 below), which has optimal average throughput but creates bottleneck links. We refer to this algorithm as RV-maxmin. It can be extended to handle traffic requirements when those are known at the IPS module at RV assignment time. In that case, instead of simply choosing the RV point that gives the maximum minimum f.t., the choice can be limited in the RV points that satisfy the given traffic requirements, again based on equation 1.

Other possible algorithms that represent the range of options in this type of IPS algorithms are:

- 1) *RV-Separation*: The new RV point returned is the midpoint of the two more distant already established points.
- 2) *RV-Greedy*: The new RV point returned is the one that maximizes the FT between the home and visiting piconet (directly from eq. 1)
- 3) *RV-tree*: The RV-tree algorithm chooses RV points from a predefined global list of available RV points. In our implementation each master divides its superframe in 3 intervals and one and only gateway may be present during such an interval. For example, if the superframe is 90 slots the RV-points 0, 30 and 60 are given out. This scheme is very simple and has optimal performance when exactly 3 gateway nodes are assigned per piconet. When more than 3 gateway nodes are needed they have to share the RV slot and take turns.

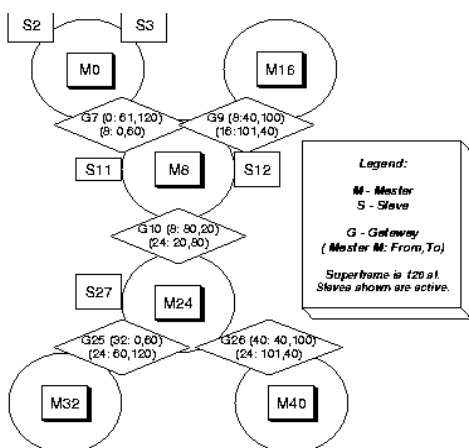


Fig. 3 The topology for the validation of the equation.

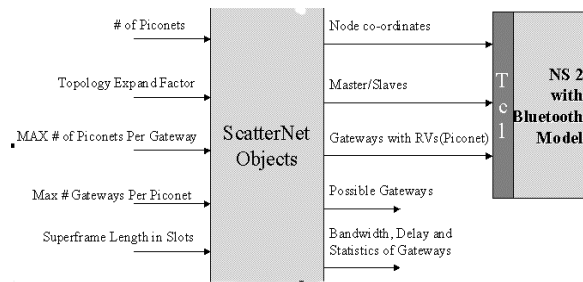


Fig. 4. Scatternet Simulator and Interface with NS and GlomoSim Bluetooth model

IV. SIMULATION MODEL

A. Scatternet Model

We have modeled in C++ a set of objects that describe the nodes, piconets and gateways and interface both with NS and GlomoSim [8] (see fig. 4). Initially the terrains dimensions are decided based on the number of piconets, the range of 10m and a topology expand factor (0.7). The latter can be used to have super-imposed or totally expanded piconets. Both dimensions then are:

$$terX = \sqrt{\# Piconets * ExpandFactor * 2 * Range} \quad (2)$$

Then, the piconets are formed by uniformly distributing the master nodes and a random number of slave nodes around them. Next a list of all the possible gateway nodes is produced according to the distances from foreign masters. This list is sorted based on a random *eventId*. This simulates the Beaconing process described in [1], as it is random which node pair becomes aware of their proximity first. Each gateway is tried in turn against system limitations (number of slaves in visiting piconets) and imposed limitations. The imposed limitations are parameters that significantly determine IPS performance. These are the Maximum Gateway per Piconet (GP) allowed limit, and the Maximum Piconet per Gateway (PG) allowed limit. The limits affect greatly the resulting topology and therefore we examine all possible values of them. That is 2 to 7 GP and 2 to 7 PG (for the RV-tree only more than 2PG are not tested). Note that the (2 GP, 2 PG) case is the totally serial connection and the (3GP,2PG) is the tree structure. After a gateway node is decided the RVs are calculated based on the algorithm proposed. After the scenario is simulated, bandwidth, delay and other statistics of the specific assignment are collected.

V. RESULTS

The following results are obtained from the Scatternet Object Simulator. The number of piconets tested ranges from 3 to 123. We run one experiment for each configuration incrementing the number of piconets by 3. The nodes ranged from 10 to 513. The terrain ranged from 20*20 to 140*140 square meters according to (2) with a 0.7 expand factor. Each point in the aggregate graphs represents the average over all gateways and over all the experiments performed for this number of piconets (this is 41 -123/3- experiments per point). In all the experiments presented, the average f.t. is not generally affected by the number of piconets but only from the imposed limits. Therefore each point in this graph shows approximately the average f.t. for each experiment as well. In the graphs we have used the 1Mbps as the link throughput.

The connectivity achieved by a scheme is an important attribute of the network created. It affects the network performance through alternate routing paths. Assume each

piconet is a node. One edge exists on the graph if there is one gateway connecting the two piconets. The average connectivity degree of the graph then is affected by the expand factor of the topology and the imposed limits. Connectivity ranges from 2 to 9 in our experiments as shown in Fig. 5. The RV-scheme is independent of the connectivity degree since in our runs the decision of whether a node becomes a gateway or not depends on the topology limitations and the imposed limits only. We present results from the case of pure round-robin. We have also obtained results from a gateway priority polling scheme. On each polling cycle the master polls the present gateways and one slave only, instead of all slaves. In a gateway priority scheme an IPS algorithm has an even larger impact, and our algorithm appears even more beneficial.

In Fig.6 and Fig. 7 we present the minimum forwarding throughput that each experiment had over all its piconet pairs connected by a gateway. Note that each point in the graph is the average of the 41 experiments performed for 3 to 123 piconets. The two axes are the imposed limit GP (x axis) and PG (y axis). Fig.6 shows the RV-maxmin against the RV-separation scheme. The RV-separation did an equal or worse job than RV-maxmin. However, as shown below, RV-separation has poor average f.t. performance. Fig. 7 shows RV-maxmin, compared with the two other schemes RV-tree and RV-greedy. The difference there is even more notable. The RV-maxmin scheme allocated RV-points so that the minimum forwarding throughput across any two directly connected piconets was double than the other schemes. Fig. 8 and Fig. 9 compare the average, instead of the minimum, performance of the RV-maxmin scheme to the three others. The average on one experiment is obtained by averaging the forwarding throughput across all piconet pairs directly connected through a gateway. One point is shown on the graph for each imposed limits pair (GP, PG). This is the average over the 41 experiments (from 3 to 123 participating piconets). The RV-maxmin has average performance very close to RV-greedy. Since it also has much better minimum f.t. performance it is the ideal scheme for Bluetooth Scatternets based on rendezvous points. Note that the minimum f.t.'s are similar between RV-greedy and RV-maxmin in the case of allowing only two piconets per gateway. The RV-greedy problem of creating bottlenecks exists particularly when a node is allowed to become a gateway of more than two piconets. Specifically, assume a gateway already connects two (or more) piconets. Assume that RV-greedy assigns RV-points that are superframe/2 apart. When a third piconet is attempted by this gateway the RV-points that maximize the f.t. for one pair may turn out to cause a bad f.t. for another pair. The RV-maxmin scheme deals effectively with this problem because it is maximizing the minimum f.t. that may occur for any pair from a new RV-point assignment. Let us now look more closely at the f.t.'s accomplished by the schemes and how they distributed across the piconet pairs. Fig. 10, Fig. 11 and Fig. 12 show all f.t.'s for each piconet pair connected through a gateway node. For illustration purposes we ordered the f.t.'s. The x-axis is an incrementing number of a connected piconet pair. Each point in the graph is the f.t. of a piconet pair in the 123 piconet case (less piconets exhibit the same shape in the graphs). Fig. 10 shows a characteristic case. We may observe the performance of the three schemes (RV-Separation –noted NOOB1, RV-greedy –noted OPTB1, and RV-maxmin– noted OPTMB1), for imposed limits (GP, PG) equal to (2,3). Note how the RV-greedy scheme on average has connected the piconet

pairs with better f.t., but there are some pairs (14 in number) that only allow an f.t. of about 10Kbps. The RV-maxmin scheme, in contrast, does not have these tails. All pairs have more than 100Kbps f.t. Similar trends are show in Fig. 11– the (3,3) (GP,PG) case and in Fig. 12 – the no limits case.

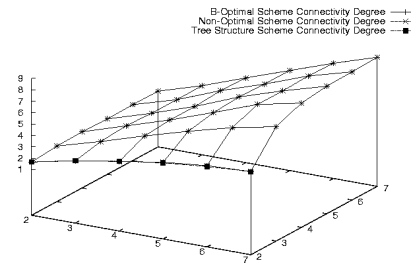


Fig. 5. Average connectivity degree versus PG (x axis) and GP (y axis) limits.

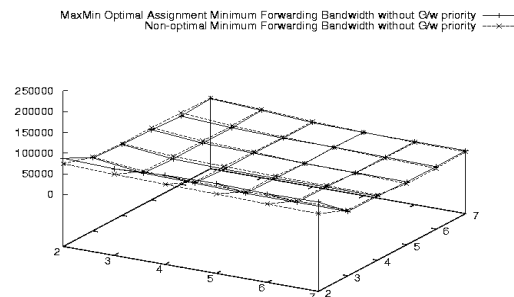


Fig.6. Round Robin f.t. without gateway priority, RV-maxmin and RV-Separation : Minimum averaged F.T. against GP (x axis) and PG (y axis) limits.

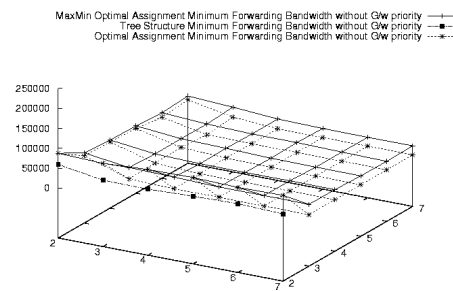


Fig. 7. Round Robin f.t. without gateway priority, RV-maxmin, RV-greedy, RV-tree : Minimum averaged F.T. against GP (x axis) and PG (y axis) limits.

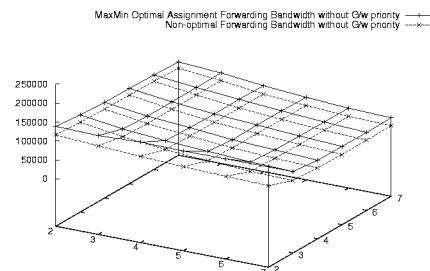


Fig. 8. Round Robin f.t. without gateway priority, RV-maxmin and RV-Separation : Averaged F.T. against GP (x axis) and PG (y axis) limits.

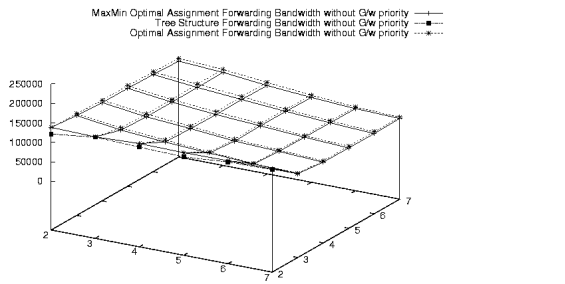


Fig. 9. Round Robin f.t. without gateway priority, RV-maxmin, RV-greedy and RV-tree: Average F.T. against GP (x axis) and PG (y axis) limits.

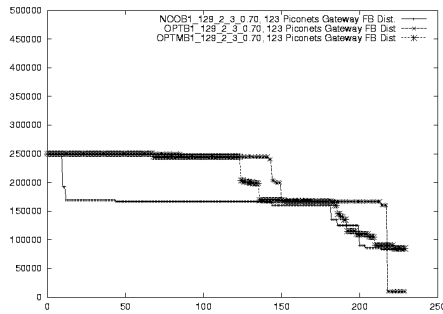


Fig. 10. Piconet pair F.t. distribution for 123 piconets (2,3) (GP,PG).

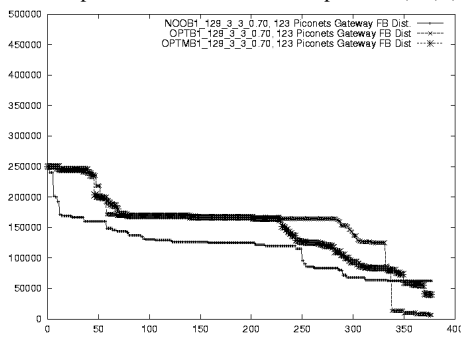


Fig. 11. Piconet pair F.t. distribution for 123 piconets (3,3) (GP,PG).

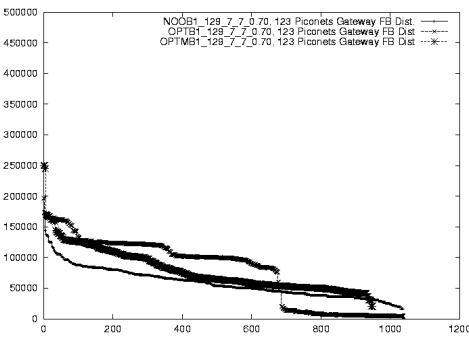


Fig. 12 Piconet pair F.t. distribution for 123 piconets and no imposed limits - (7,7) (GP,PG).

Next we look at scatternet end-to-end results. The graphs presented here are obtained by running several simulations of a scatternet consisting of 123 piconets. In order to obtain end-to-end results we run a Dijkstra algorithm to get routes formed with (i) the minimum hop distance path between all Piconet pairs and (ii) the maximum forwarding throughput path. In Fig 13 we show the min hop distances achieved for each combination of (GP, PG) limit. The min hop distance depends on the actual distance of the piconets, the applied limits and the random order that nodes sense foreign masters. The graph shows that the no limit ((GP,PG)=[7,7]) min hop distance –which is the best average min hop- can be achieved for any limit combination that allows 3 or more gateways per piconet. In fact full connectivity is

achieved for limits higher than (3,3) as well. In Fig 14 we see the hop distribution for the (3,3) fully connected case. In Fig 15 we see the average end-to-end bandwidth that can be achieved for each limit pair. The rendezvous point allocation scheme used is RV-MaxMin-Optimal. We see that the highest end-to-end throughput can be achieved for the (3,2) case and then for the (3,3) case. The worst case is the serial connection (2,2). In Fig 16 we look at the fully connected case (3,3) and show the distribution of the end-to-end throughputs for each pair of piconets. We note that while the RV-optimal case has slightly higher average throughput, few piconet pairs can only communicate with less than 25Kbps. The RV-MaxMin-Optimal does not create these bottleneck points and the minimum end-to-end throughput pairs can communicate with 100Kbps instead.

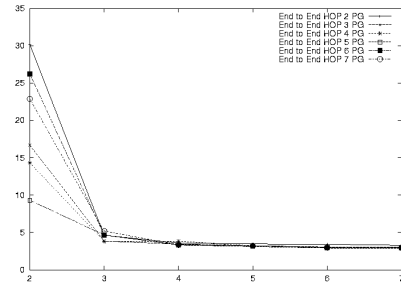


Fig 13. Hop distance averaged over all piconet pairs in scatternet. (x axis is the GP limit)

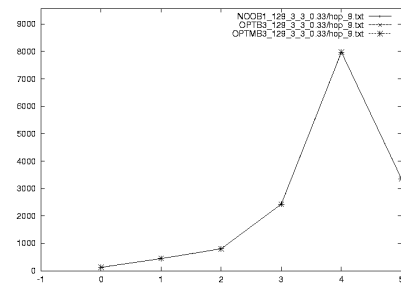


Fig 14. Hop distance distribution for the (3,3) (GP,PG) limit (x axis is hop distance, y axis is number of piconet pairs that are connected with the hop distance with a min-hop route)

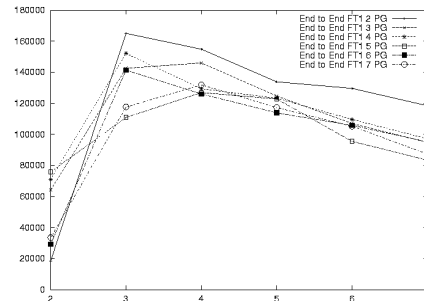


Fig 15. End-to-end route throughput averaged over all piconet pairs for the RV-maxmin case. (x axis is the GP limit, y axis is the route minimum f.t.)

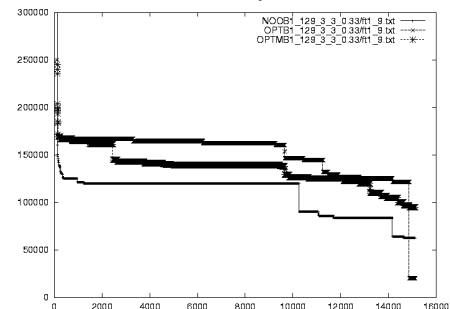


Fig 16. Route throughput distribution for the (3,3) (GP,PG) limit (x axis is route throughput in bps, y axis is an increasing number that represents a piconet pair, it is ordered by route throughput) .

VI. CONCLUSION

The introduction of Bluetooth scatternets enables a wider and more flexible use of Bluetooth network as the means to create local and short range connectivity. However, the scatternet also presents a technical challenge with respect to network formation and scheduling of traffic within a Bluetooth network. In this paper the focus was on rendezvous inter-piconet scheduling (IPS) issues in particular. We studied the impact of the IPS algorithm on the scatternet performance. We consider this an important question because the inter-piconet scheduling functions require support from low layers usually implemented in silicon [2]. We argued that some intelligence in the inter-piconet scheduling is necessary and used a general architecture for scheduling in scatternets that enables the deployment of IPS optimizing algorithms.

We specifically studied the rendezvous point assignment issues in Bluetooth scatternet architectures that are based in rendezvous points established between master and gateway nodes. Establishing honored RV-points is a promising approach because it sets the ground for possible QoS guarantees and does not waste slots on failed master/gateway attempts to communicate. We found that the ideal algorithm in terms of gateway forwarding throughput is the RV-maxmin. This algorithm has average performance very close to the one provided by RV-greedy and does not have the disadvantage of creating low throughput gateway connections. This is highly likely to happen in RV-greedy when more than 2 piconets are allowed per gateway. Note that optimality here is local in time and space, i.e. as the problem is known at the time and place of solution. When an event occurs – a node is to become the gateway between a new pair of piconets – it optimally solves this local problem. We structured the design space according to imposed limits on the number of Piconets allowed per Gateway (PG) and on the number of Gateway allowed per Piconet (GP) and explored all possibilities of a scatternet architecture based in fixed (for the life of a gateway connection) Rendezvous Points that are always honored by master and gateway. Intelligence added to the IPS increases the space (silicon) and time cost but improves overall scatternet performance by 30%, a percentage that is crucial for some applications especially in the scatternet expected throughput range. The algorithm can be extended to support RV point re-negotiation and take into account traffic requirements, if those can be made known to the IPS module. We showed that RV-maxmin, based on maximizing the minimum forwarding throughput across the involved piconet pairs is ideal for scatternets. In a nutshell, we argue that a totally random one is inappropriate for most applications. The maxmin optimum algorithm on the other hand is shown by simulation experiments to have a total network throughput of more than a 30% than simpler non-random solutions without creating bottleneck gateways. This improvement may be considered important to some applications, especially for the range of typically expected scatternet data rates, but comes at the cost of a higher design complexity/cost.

REFERENCES

[1] "Bluetooth Specification" (*Base band specification*) and <http://www.bluetooth.com>

- [2] Per Johansson (Ericsson), YZ lee, Mario Gerla, Manthos Kazantzidis (UCLA) – "Bluetooth an Enabler of Personal Area Networking" - *IEEE Network, Special Issue in Personal Area Networks, Sept-Oct 2001*
- [3] T. Salonidis, P. Bhagwat, L. Tassiulas, R. La Maire "Distributed topology construction of Bluetooth personal area networks." *Infocom 2001*
- [4] Miklos, G.; Racz, A.; Turanyi, Z.; Valko, A.; Johansson, P. "Performance aspects of Bluetooth scatternet formation" *2000 First Annual Workshop on Mobile and Ad Hoc Networking and Computing. MobiHOC*
- [5] Raman, B.; Bhagwat, P.; Seshan, S. (Edited by: Ikeda, K.; Douglass, F.) "Arguments for cross-layer optimizations in Bluetooth scatternets" *Proceedings 2001 Symposium on Applications and the Internet*
- [6] A. Aggarwal, M. Kapoor, L. Ramachandran, A. Sakar "Clustering Algorithms for Wireless Ad Hoc Networks" *Fourth International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL M for Mobility) 2000*
- [7] Mario Gerla, Rohit Kapoor, Manthos Kazantzidis (UCLA), Per Johansson (Ericsson) "Ad hoc Networking with Bluetooth" *WMI at Mobicom 2001*
- [8] M. Takai L, Bajaj, R, Ahuja, R. Bagrodia and M. Gerla, "GloMoSim: A Scalable Network Simulation environment" *Technical report 990027, UCLA, Computer Science Department, 1999.*
- [9] Jaap Haartsen "Bluetooth--The Universal Radio Interface For Wireless Connectivity" *Ericsson Review, No. 3, 1998*
- [10] Haartsen, J.C., "The Bluetooth Radio System" in *IEEE Personal Communications Volume: 7 1, February 2000, Page(s): 28 –36*
- [11] Bhagwat, P.; Segall, A. "A routing vector method (RVM) for routing in Bluetooth scatternets" *1999 IEEE International Workshop on Mobile Multimedia Communications (MoMuC'99)*
- [12] Kalia, M.; Bansal, D.; Shorey, R. "MAC scheduling and SAR policies for Bluetooth: a master driven TDD pico-cellular wireless system." *1999 IEEE International Workshop on Mobile Multimedia Communications (MoMuC'99)*
- [13] Capone, R. Kapoor and M. Gerla: "Efficient Polling Schemes for Bluetooth Picocells" *ICC 2001*
- [14] Basagni, I. Chlamtac, G. V. Záruba, "Bluetrees – Scatternet Formation and Routing in Bluetooth-Based Ad Hoc Networks" *Proceedings of IEEE INFOCOM 2001, Anchorage, Alaska, USA, April 22-26, 2001.*
- [15] Das, A. Ghose, A. Razdan, H. Saran, R. Shorey, "Enhancing Performance of Asynchronous Data Traffic over the Bluetooth Wireless Ad-Hoc Network", *Proceedings of IEEE INFOCOM 2001, Anchorage, Alaska, USA, April 22-26, 2001.*
- [16] N. Johansson, U. Körner, L. Tassiulas, "A Distributed Scheduling Algorithm for a Bluetooth Scatternet," *Seventeenth International Teletraffic Congress, ITC'17, Salvador da Bahia, Brazil, September 24-28, 2001.*
- [17] P. Johansson, R. Kapoor, M. Kazantzidis and M. Gerla – "Personal Area Networks: Bluetooth or IEEE 802.11?" - *International Journal of Wireless Information Networks SPECIAL ISSUE ON MOBILE AD HOC NETWORKS (MANETs) Standards, Research, Applications, April 2002*
- [18] P. Johansson, R. Kapoor, M. Kazantzidis, M. Gerla - Rendezvous Scheduling in Bluetooth Scatternets – *ICC 2002*
- [19] A. Racz, G. Miklos, F. Kubinszky, A. Valko "A Pseudo Random Coordination Scheduling Algorithm for Bluetooth Scatternets", *ACM MobiHoc 2001.*
- [20] N. Johansson, F. Alriksson, U. Jonsson "JUMP Mode- A Dynamic Window-based Scheduling Framework for Bluetooth Scatternets", *ACM MobiHoc 2001.*