

Enhancing Scatternets Performance via Scheduling Algorithm Parametrization

Werner Priess¹, José Ferreira de Rezende², and Luci Pirmez¹

¹ NCE – Universidade Federal do Rio de Janeiro
Caixa Postal 2324,
20001-970 Rio de Janeiro, RJ, Brazil
`priess@posgrad.nce.ufrj.br`
`luci@nce.ufrj.br`

² COPPE/PEE – Universidade Federal do Rio de Janeiro
Caixa Postal 68504,
21945-970 Rio de Janeiro, RJ, Brazil
`rezende@gta.ufrj.br`

Abstract. Bluetooth specification still has open issues, including the intra and interpiconet scheduling topics. This paper proposes an interpiconet scheduling algorithm, referred to as AISA (Adaptive Interpiconet Scheduling Algorithm). AISA is characterized by: its adaptability to varying network traffic conditions; and its ability to optimize specific performance metrics via parametrization. Both features enable it to be employed in a variety of scenarios with improved performance showed by the simulation results.

Keywords: *ad hoc* networks, Bluetooth, scatternet, scheduling, QoS.

1 Introduction

Bluetooth is a promising radio technology for *ad hoc* networking. Its interfaces are small in size and are energy efficient at an increasingly lower cost. The Bluetooth network topology, or *piconet*, has a maximum of eight devices. In order to extend the network range and the number of devices, the scatternet concept was created. A scatternet is a network formed of two or more piconets interconnected by shared nodes that will be referred to as *bridges* throughout the paper.

Research in scatternets is recent and a number of open issues remains. Interpiconet scheduling is one of them and it defines the mode by which the bridges participate in the piconets that they are connected to. Since a bridge is usually attached to only one Bluetooth interface, it must alternate in time its participation in multiple piconets. Most existing interpiconet scheduling proposals evaluate aggregate throughput and packet delay metrics, but few proposals handle the power consumption performance metric. Moreover, their application is usually limited to specific scenarios.

This paper presents an algorithm for interpiconet scheduling, referred to as AISA (Adaptive Interpiconet Scheduling Algorithm). Parametrization is the key concept behind AISA, enabling the bridges to be configured so as to optimize one chosen performance metric like, for example, throughput, packet delay or power consumption. The fact that AISA is applied solely to the bridges minimizes the need for adaptations to the Bluetooth standard. In this study AISA performance was evaluated by way of three simulated scenarios whose focused metrics differed. The results showed that the algorithm performed well in all three situations.

This paper is organized as follows: Section 2 presents a brief overview of Bluetooth technology; Section 3 summarizes the related work on interpiconet scheduling; Section 4 describes AISA; Section 5 presents the simulated scenarios and their results; and the conclusions are presented in Section 6.

2 Bluetooth Specification

Bluetooth [1,2] is a short range and low power radio technology intended to replace cable connections between electronic devices. It may also be used to create small wireless *ad hoc* networks. The Bluetooth Special Interest Group (SIG) [1] released an open specification with two parts: the Core and the Profiles. The Core Specification defines radio characteristics and the protocol stack. The Profiles define which protocols of the stack should be implemented for each application. In this section we provide some information about the Core Specification.

2.1 Physical and Link Layers

The Bluetooth radio operates in the unlicensed Industrial, Scientific and Medical (ISM) band at 2.4 GHz and it uses a fast frequency hopping scheme. It hops over 79 channels (23 in some countries) displaced by 1 MHz at the rate of 1600 hops per second, corresponding to a 625 μ s time slot.

The baseband is responsible for creating piconets and links. The piconet is a network with at most eight active devices that share the same frequency hopping scheme. One device becomes the piconet master and the others behave as slaves. The master dictates the hopping sequence and its phase.

A time division duplex (TDD) scheme is used where master and slaves alternatively transmit. A slave is allowed to transmit only if it has been addressed by the master in the prior slot. In every new slot, the devices of a piconet hop to the next frequency of the hopping sequence.

The baseband handles two types of links: Synchronous Connection-Oriented (SCO) and Asynchronous Connectionless (ACL). SCO is a symmetric point-to-point link between the master and a single slave, maintained with slot reservation at regular intervals by the master. ACL is a point-to-multipoint link between the master and all slaves participating on the piconet. ACL traffic may only occupy slots not reserved for SCO links.

A major Bluetooth concern is power consumption. There are three low power operation modes: Sniff, Hold and Park. In Sniff mode, a slave listens to the

piconet only at periodic time slots, called sniff slots. In Hold mode, a slave goes into sleep for a specified time period, after which it returns to Active mode. In Park mode, a slave releases its active member address, but remains synchronized to the piconet for future activation.

The Link Manager Protocol (LMP) and the Logical Link Control and Adaptation Protocol (L2CAP) form the Bluetooth link layer. LMP is responsible for configuring and managing baseband connections. For example, when a bridge wants to enter Hold power saving mode, it communicates this fact to the piconet master through LMP signaling packets. L2CAP provides connection-oriented and connectionless services to upper layer protocols with protocol multiplexing capability and segmentation and reassembly operation (SAR).

2.2 Scatternets

If devices on different piconets want to communicate, these piconets may be interconnected, creating a scatternet. The scatternet is formed when at least one device, referred to as a bridge, participates of two or more piconets. Bridges may be attached to only one Bluetooth interface, preventing them from being active in more than one piconet at the same time. Therefore, a bridge alternates in time its participation in multiple piconets. This task is called interpiconet scheduling. A bridge can be a slave in more than one piconet, but a master in only one.

The scatternet concept brought along new issues regarding its implementation. These issues have been addressed by several research studies. Research related to interpiconet scheduling is discussed in the next section.

3 Related Work

Research on scatternets is concentrated in three main topics: topology formation [3,4], packet forwarding [5,6] and interpiconet scheduling [7,8,9,10,11]. The first two topics are out of scope of this paper.

According to our point of view, the interpiconet scheduling algorithms may be divided into two categories, depending on the device that coordinates the scheduling process [12]: mechanisms with isolated decision and with distributed decision. In the first case, the bridge itself decides about its presence in the piconets it belongs to and it may communicate (or not) this decision to the masters of these piconets (if acting as a slave). These algorithms are usually simpler to implement, requiring few modifications in Bluetooth Specification. In mechanisms with distributed decision, decisions about future meeting points between the bridge and each master from the piconets in which it participates as a slave result from agreements between both devices. These agreements enable information exchange about the scatternet topology, making better throughput results possible. However, the distributed decision approach requires new LMP messages, which increases the implementation complexity.

Initial studies dealt with scatternet scheduling in a generic way. Their major concern was to study traffic behavior and not specifically scheduling algorithms [13]. Gerla et al. [14] introduced the *rendezvous point* concept, meaning the slot in which a bridge and a piconet master decide to communicate. They also presented the *rendezvous window* which is basically the rendezvous time period.

Among the distributed decision proposals, Johansson et al. [7] presented a scatternet scheduling framework, based on their proposed JUMP mode. A device in JUMP mode on a piconet is, by default, absent of that piconet. A bridge is able to alternate between piconets without explicitly notifying it. However, the inclusion of a new mode to the Bluetooth link controller may not be possible.

Kapoor et al. [8] presented another distributed decision mechanism based on the *rendezvous point* (RP) concept. Each master maintains a list containing its RPs with the bridges, and a list containing its bridges RPs with other piconets. This information enables the master to optimize further RP allocation. It is not completely clear in the paper the way information is passed to the master. The main drawback of this algorithm is that it demands a lot of messages to be exchanged to keep masters up-to-date.

The Tree Scatternet Scheduling (TSS) scheme [9] was designed to work with the Tree Scatternet Formation mechanism from the same authors. A tree-based network topology simplifies the scheduling task, and it enables a global coordination among all piconets to be achieved. On the other hand, the algorithm applicability is restricted to some specific topologies without loops.

As for the research of isolated decision mechanisms, Racz et al. proposed the Pseudo-Random Coordinated Scatternet Scheduling (PCSS) algorithm [10]. Devices assign meeting points with their peers through a pseudo-random process. These meeting points will be different for each pair of nodes. An advantage of PCSS is the coordination among devices with no explicit signaling needed. However, as the number of devices increases, the meeting points are likely to collide.

Har-Shai et al. proposed the Load Adaptive Algorithm (LAA) [11] that operates only on bridges. In this algorithm, each bridge adapts to traffic variations by observing its queues and receiving information about the other end node queue. Probably a new field will be necessary in data packets to piggyback this queue information. Currently, LAA is applicable to small scatternets, in which bridges connect only two piconets.

Summarizing, most proposals present the following limitations. A bridge may belong to only two piconets, always working as a slave. They do not scale in terms of the number of scatternet nodes. Proposals evaluate aggregate throughput and packet delay metrics, but only PCSS [10] handles the power consumption performance metric.

4 Proposed Interpiconet Scheduling Algorithm: AISA

This section presents the Adaptive Interpiconet Scheduling Algorithm (AISA). AISA differs from other interpiconet scheduling mechanisms in that it enables the

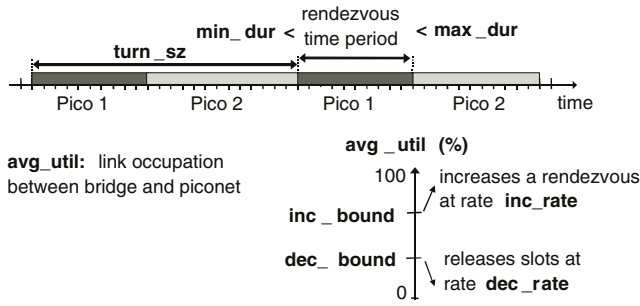


Fig. 1. AISA parameters

choice of a performance metric to be optimized by configuring parameters that work as performance metric tuning knobs. More specifically, depending on the algorithm parametrization it is possible to prioritize one of three metrics: traffic aggregate throughput, packet delay or power consumption. Moreover AISA adapts to varying traffic conditions and provides fairness among flows that cross a bridge.

AISA operates only on bridges. Therefore, it is up to a bridge to decide how long it will remain in each piconet (the corresponding of a RP window). AISA was developed to behave in this way to avoid having to create new signaling packets specific to the scheduling task. As a result, it is possible to minimize modifications in the Bluetooth Specification. According to the classification described in the previous section, AISA fits into the isolated decision mechanism category. The remaining of this section explains how AISA works in terms of its parameters.

A bridge¹ schedules its piconets in a Weighted Round Robin (WRR) [15] fashion. Time is divided up into turns all with the same time period (*turn_size* in Figure 1). During each turn the bridge will be connected to each one of the piconets for a certain length of time (rendezvous time period in Figure 1). The lower limit of the rendezvous time period is referred to as *min_dur* and the upper limit as *max_dur*. When leaving a piconet, the bridge calculates the rendezvous time period with this piconet for the next turn, based on the percentage of slots occupied by data packets in the current rendezvous, i.e., the average link occupation (*avg_util* in Figure 1). As the link occupation increases or decreases, the bridge updates the next rendezvous time period.

If *avg_util* goes below the *dec_bound* limit, the bridge will try to reduce the rendezvous time period by releasing slots at the *dec_rate* rate (see Figure 1). Conversely, if *avg_util* exceeds the *inc_bound* limit, the bridge will try to extend the next rendezvous time period by acquiring slots at the *inc_rate* rate. Note that the *turn_size* is kept constant during all bridge operation.

¹ All explanations consider bridges as slaves in all piconets. When a bridge works as a master in one piconet, it controls this piconet and doesn't need to signal its presence.

It may happen that the bridge requires an extension of a rendezvous time period but there are not enough free slots available. In this case, the bridge tries to remove slots from the longest rendezvous it participates in. However, the rendezvous from which slots are removed cannot have a shorter time period than the one that is being extended. This rule aims to ensure fairness for all piconets.

Before leaving a piconet, the bridge sends a `LMP_Hold_Req` packet to the piconet master, informing the exact moment that it will enter Hold mode and the moment it will return to Active mode. Thus, the piconet master knows about the bridge absence and removes it from the intrapiconet scheduling scheme during the agreed period.

Bridges try to save power by entering Hold mode, what may happen in two ways. In the first case, as long as the traffic between a bridge and a piconet reduces, slots are set free. These slots are accumulated, and in the remaining slots at the end of each turn the bridge may enter Hold mode until the beginning of the next turn. The second case takes place when a rendezvous time period is already at minimum (*min_dur*) and even so the link utilization (*avg_util*) between the bridge and the piconet is low. Then, the bridge may not schedule this piconet on the next turn, returning to schedule it on the following turn. A piconet may be skipped only if the boolean parameter *skip_pico* is enabled.

Summarizing, the following parameters were introduced: *turn_size*, *min_dur*, *max_dur*, *inc_bound*, *dec_bound*, *inc_rate*, *dec_rate*, and *skip_pico*. Depending on their configuration, a metric performance may be improved or not.

5 Simulations and Results

This section presents three simulated scenarios and their results. Each scenario focuses on one specific performance metric, which are: aggregate throughput, packet delay and power consumption. Depending on the chosen metric, some parameters remain unchanged, while others vary in order to obtain the best metric configuration.

We developed a Bluetooth extension to the Network Simulator (ns-2) [16]. It is referred to as BlueNetS (Bluetooth Network Simulator) and was introduced and validated in [12]. BlueNetS tool models physical and link layer Bluetooth characteristics necessary to simulate traffic communication. The connection establishment procedures are not simulated, and we employ static scatternet configurations. Regarding upper layer protocols (TCP, UDP, IP) and applications, the available ns-2 modules were used.

5.1 Scenario 1 – Throughput Metric

In Scenario 1, AISA parameters were configured to maximize the interpiconet traffic throughput. A scatternet, composed of three piconets interconnected by way of one bridge, was used to demonstrate this bridge capability via parametrization (Figure 2).

Table 1. Fixed AISA parameters in Scenario 1

min_dur	max_dur	inc_rate	dec_rate	skip_pico
20 slots	turn_size slots	20%	20%	0 (bool)

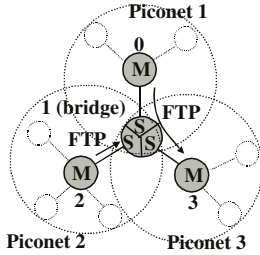


Fig. 2. Scenario 1

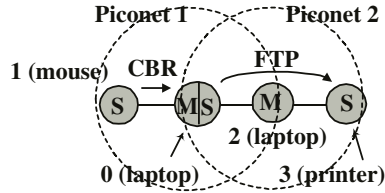


Fig. 3. Scenario 2 (M=master; S=slave)

File transfer traffic (FTP) was configured from 0 to 3, and from 2 to 1. After each file transfer, the FTP sources wait an interval before they initiate the next file transfer. Both file size and time interval between files follow exponential distributions. In traffic between 0 and 3, these distributions have mean values of 30 Kbytes and 1 sec, respectively, and between 2 and 1, 40 Kbytes and 1 sec, respectively. As an example, this traffic model may represent photo file transmissions from a digital camera to a laptop or small printing jobs.

Table 1 shows the parameters that remained unchanged throughout Scenario 1 simulations. The parameters were chosen to enable fast bridge adaptation to traffic changes. In order to maximize aggregate throughput the minimum rendezvous time period between the bridge and a piconet (*min_dur*) should be as small as possible, which allows the bridge to grant slots to piconets with more interpiconet traffic load. As for the increase rate (*inc_rate*) and decrease rate (*dec_rate*), their effect is limited by the number of free and busy slots, respectively, and it has minimum influence on the scenario (so, the chosen values were an average of previously tested values). Finally, as the goal of *skip_pico* is power saving, it was disabled.

5.2 Simulation Results from Scenario 1

The simulation provides comparative results between AISA and Round Robin (RR). AISA was tested with two different configurations: AISA 1, with the limits *inc_bound* (above which a bridge tries to increase the rendezvous time period) and *dec_bound* (below which a bridge tries to release slots) equal to 80% and 60%, respectively; and AISA 2, with *inc_bound* and *dec_bound* equal to 90% e 50%, respectively. The results are averages of 10 simulation runs, each of them with 120 seconds of simulation time. The uncertainty is expressed as 95% confidence intervals (CI).

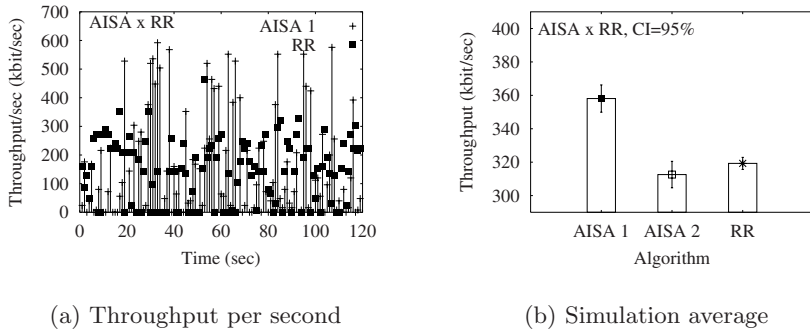


Fig. 4. Comparative results between AISA and RR algorithms

Since different piconets are not synchronized in time, a bridge loses up to two slots while switching from one piconet to the other. Therefore, as the turn gets smaller, the percentage of lost slots caused by piconet switches grows. In this simulation, the turn duration (*turn_size*) was set to 240 slots² (equivalent to 80 slots/piconet in the beginning of each run).

Figure 4 (a) shows the FTP aggregate throughput, measured in each subsequent second interval, to configurations AISA 1 and Round Robin (RR). The curve oscillation is caused by the silence and transmission periods in the modeled traffic. AISA enables traffic peaks above 500 kbit/sec, represented by vertical bars. These peaks are caused by the dynamic slot redistribution of AISA. In RR case, almost all points are under 400 kbps.

Figure 4 (b) shows the average aggregate throughput of FTP flows in the entire simulation. AISA 1 obtained a 15% gain over RR. This difference is even larger with bigger files since peak periods are longer.

AISA 1 provided a better scatternet performance than AISA 2. As *inc_bound* approaches to 100% (the link occupation *avg_util* needs to become larger to reach it), it is more difficult to increase a rendezvous time period. Similarly, as *dec_bound* decreases, it is more difficult to release slots. AISA 2 configuration makes the rendezvous time period more stable, reducing AISA adaptability.

5.3 Scenario 2 – Delay Metric

This scenario is formed by two piconets. At a certain moment, the laptop (master of Piconet 1) establishes a connection to the computer (master of Piconet 2) in order to print a file. In Piconet 2, the laptop is a slave, defining a master/slave bridge. Figure 3 illustrates the scenario.

Since the mouse (in Piconet 1) is an interactive device, it may be considered a delay-sensitive traffic source. It generates one 16-byte packet every 65 ms (similar to the one used by Racz et al. [10]). Background traffic is characterized by the printing traffic flow, which is modeled as a long-sized file transfer (FTP).

² *turn_size* values above 240 slots give close results in terms of throughput (see [12]).

Table 2. AISA fixed parameters in Scenario 2

inc_bound	dec_bound	inc_rate	dec_rate	max_dur	skip_pico
80%	60%	20%	20%	turn_size slots	0 (bool)

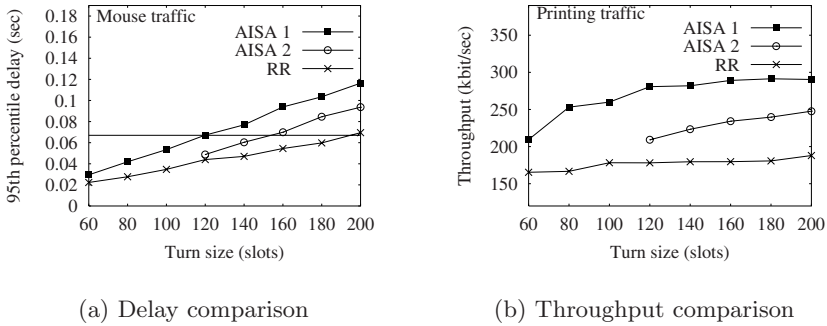


Fig. 5. Comparative results between AISA and RR, with two piconets

Table 2 presents AISA parameters that remained unchanged throughout simulations of Scenario 2. Since the mouse traffic rate is low, parameters that influence on how fast a bridge increases or decreases a rendezvous time period (including *inc_bound*, *dec_bound*, *inc_rate* and *dec_rate*) have no effect over mouse generated traffic. However, these parameters affect background traffic throughput. So, the same values tested in AISA 1 from Scenario 1 were adopted in this scenario.

5.4 Simulation Results from Scenario 2

The mouse traffic was tested in the presence of the printing job. The results are averages of 10 simulation runs, each of them with 120 seconds of simulation time. Two AISA configurations and Round Robin (RR) were compared. The AISA configurations are AISA 1, with *min_dur* (minimum rendezvous period) equal to 20 slots, and AISA 2, with *min_dur* equal to 50 slots. AISA 1 and RR were tested with the turn size (*turn_size*) varying from 60 to 200 slots. In AISA 2, simulations started at 120 slots (equivalent to 60 slots/piconet).

Figure 5 (a) shows the 95th percentile delay for the three configurations. Since the mouse traffic was modeled with one packet per 65 ms, we decided to use this value as the maximum acceptable delay.

AISA 1 reaches the maximum acceptable delay close to 120 slots. By increasing the minimum rendezvous period (in AISA 2), this delay limit is reached close to 150 slots. This difference is due to the fact that increasing a minimum rendezvous time period (*min_dur*) reduces the number of slots left to other piconets.

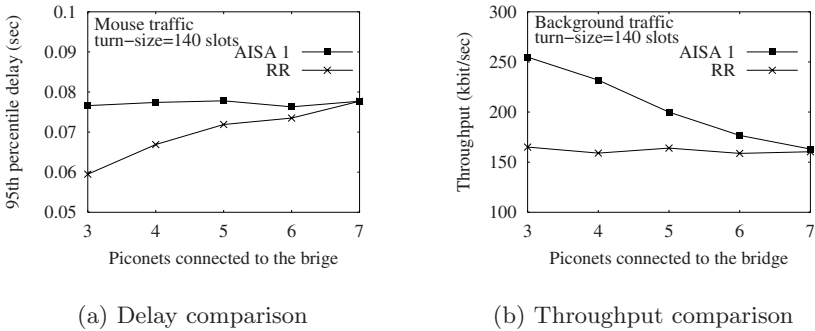


Fig. 6. AISA and RR results, increasing the number of piconets connected to the bridge

Since the mouse throughput is low, the bridge stays the minimum possible period (*min_dur*) in its piconet. When *min_dur* is 20 slots and *turn_size* is 120 slots, 100 slots (62.5 ms) are left to the printer piconet, which is close to the maximum acceptable delay.

Round Robin (RR) guarantees smaller delay values than AISA, but this reduction comes with smaller values of background traffic throughput. This effect is presented in Figure 5 (b). For a *turn_size* of 120 slots, the AISA throughput result (280 kbit/sec) is 60% better than the best RR result (180 kbit/sec).

We have also observed the mouse packet delays when the bridge participates in more than two piconets. The number of piconets that the bridge participates in was varied from three to seven. The simulation was run with a 140 slot turn size, ensuring that even with seven piconets, the 20-slot *min_dur* is honored for all piconets. The 95th percentile mouse packet delay and background traffic aggregate throughput are shown in Figure 6.

Regardless of the number of piconets connected to the bridge, the AISA 1 delay curve is stable, since the bridge always stays the minimum rendezvous period in Piconet 1. The bridge distributes the rest of the turn among other piconets. On the other hand, RR distributes the turn equally among all piconets. Consequently, the delay increases as the number of piconets increases.

Regarding the throughput, RR curve is almost constant, but AISA curve decreases as the number of piconets increases. Each new piconet connected to the bridge reduces the printing job throughput. Besides, more slots are lost because of piconet switching. Even so, in the configuration with six piconets AISA outperforms RR in about 10%.

5.5 Scenario 3 – Power Consumption Metric

The goal of Scenario 3 is to show that a correct AISA parametrization may reduce bridge power consumption and, consequently, overall consumption. This metric evaluation is important to *ad hoc* networks, and, more specifically, it may be useful to sensor network implementations.

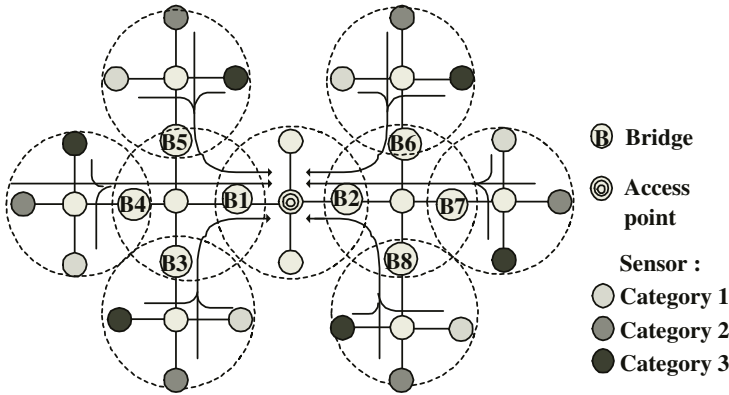


Fig. 7. Scenario 3

The topology is a scatternet formed by nine piconets as illustrated in Figure 7. Only the border piconets have data sources. Sources are divided in three categories, representing different sensing data values. Each piconet has one source from each category. The central node is an access point, which is the sink for all sources. Sources generate packets at a 3 kbit/sec constant rate. Categories 1, 2 and 3 use 300-, 100- and 20-byte packet length, respectively, resulting in different packet intervals.

The AISA parameters that remained unchanged throughout this simulation set are: *min_dur* (20 slots), *dec_rate* (20%) and *skip_pico* (1). The minimum rendezvous period (*min_dur*) should be small so that a bridge stays less time in piconets with low link occupation. The parameter *skip_pico* enables power saving.

In order to measure power consumption we defined an energy unit (e.u.). One e.u. is the transmission power of a one-slot packet³. Note that in Bluetooth a packet transmission dissipates almost the same power of a packet reception ([17, 18]).

5.6 Simulation Results from Scenario 3

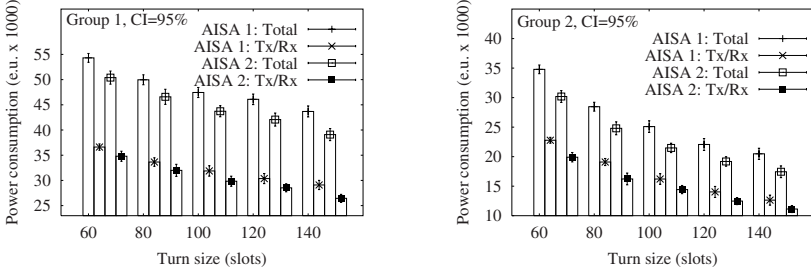
Two AISA configurations and Round Robin (RR) were simulated. RR does not save power but it was simulated for packet delay comparison. Although the delay metric is not focused in this scenario, power saving should not cause excessive packet delay. AISA 1 and AISA 2 parameter configurations are presented in Table 3.

Power consumption and packet delay were measured with AISA 1, AISA 2 and RR. The results are averages of 10 simulation runs, each of them with 120 seconds of simulation time. The uncertainty is expressed as 95% confidence intervals (CI). In each run, the turn size was varied from 60 to 140 slots.

³ More details about power levels used in this paper may be obtained in [12].

Table 3. AISA 1 and AISA 2 configurations

	inc_bound	dec_bound	inc_rate	max_dur
AISA 1	80%	60%	20%	turn_size slots
AISA 2	90%	70%	10%	(60% of turn_size) slots



(a) Group 1 power consumption

(b) Group 2 power consumption

Fig. 8. Power consumption of bridge Groups 1 and 2, with configurations 1 and 2

There are eight bridges in this scenario. The bridges are divided in two groups, regarding its position in the network topology: Group 1, with Bridges 1 and 2, and Group 2, with Bridges 3 to 8. Due to the traffic source positions in the scatternet, the bridges from Group 1 will dissipate almost the same power. Analogously, results within Group 2 will be close to each other. Therefore, all results are presented in terms of averages obtained for each group. Figure 8 presents both groups dissipated power in this simulation ⁴.

As expected, the results show that, for all turn sizes, the bridges from Group 1 consume more power than those from Group 2, because all packets addressed to the access point are routed by Group 1. Certainly, the lifetime of Group 1 will be smaller than the lifetime of Group 2. A solution to this problem will be discussed later.

In all cases, as the turn size increases bridge power consumption is reduced. This result is explained as follows. The bridges normally remain the minimum rendezvous time period (min_dur) in each piconet. Increasing the turn size will enable bridges to accumulate free slots at the end of each turn. During these free slots bridges enter Hold mode.

Comparing AISA 1 and AISA 2 results, one may note that AISA 2 causes less power consumption than AISA 1 for both groups (Figure 8). In AISA 2, it is easier for a bridge to release slots (because $dec_bound_{AISA2} > dec_bound_{AISA1}$ in Table 3), and more difficult to increase a rendezvous time period (because

⁴ RR does not save power, since bridges are always active in a piconet. As an example, for Group 1 with turn of 60 slots, the average power consumption was 70461 ± 1023 e.u., representing about 30% more power than with AISA 1.

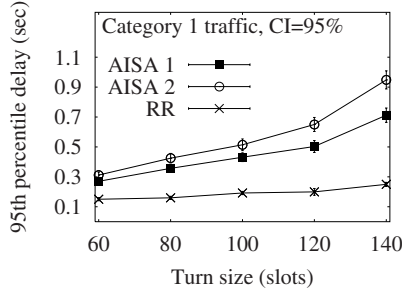


Fig. 9. Packet delay for category 1 traffic

$inc_bound_{AISA2} > inc_bound_{AISA1}$). When the link utilization (avg_util) goes beyond inc_bound , AISA 2 increasing rate is smaller than that of AISA 1. Finally, AISA 1 lets a bridge occupy all free slots in a rendezvous, while AISA 2 has an upper limit for a rendezvous time period (max_dur).

RR was included in the study of packet delay. Figure 9 shows the average packet delay for category 1 sources. The other two categories presented similar curves.

RR algorithm causes the smallest delay values, but without power saving. AISA 2 presents delay values larger than AISA 1. So, there is a trade off between power saving and packet delay.

Due to the bridges position in the scatternet, Bridges 1 and 2 consume more power than those belonging to Group 2. If all bridges start with the same battery power, Bridges 1 and 2 will cease to function before the others (Group 2), and there will be no route to the access point. So, we should find a specific configuration for each bridge group in order to equalize power consumption. Also, new configurations should not increase packet delay.

Scenario 3 was tested with one different parametrization for each bridge group. Table 4 summarizes both configurations. Note that Group 1 was configured similarly to AISA 2 above in order to minimize power consumption by this group. However, delay increases with this configuration. To counterbalance the delay problem, although increasing consumption, Group 2 was configured similarly to AISA 1.

The average power consumption was calculated for each bridge group, with uncertainty expressed as 95% confidence intervals (CI). Table 5 contains this result. One important conclusion is that the difference between the dissipated power by bridges from Groups 1 and 2 was less than 3%.

The average packet delay for the three traffic categories was also calculated. The results for categories 1, 2 and 3 were, respectively, in seconds: $0,627 \pm 0,015$; $57 \pm 0,032$; and $0,58 \pm 0,024$. From the power consumption and packet delay results, we conclude that Bridges 1 and 2 consumed 20% less power than in case of AISA 2 for the same delay results (AISA 2 with 120 slot turn in the previous simulation). This confirms that it is possible to configure AISA to extend network lifetime.

Table 4. AISA parametrization to equalize power consumption

	turn_size	inc_bound	dec_bound	inc_rate	max_dur
Group 1	140 <i>slots</i>	90%	70%	10%	80 <i>slots</i>
Group 2	60 <i>slots</i>	80%	60%	20%	turn_size <i>slots</i>

Table 5. Power consumption, with different configurations for Groups 1 and 2

	Group 1	Group 2
Total consumption (e.u.)	35134 ± 775	34436 ± 591
Consumption of transmitted and received packets (e.u.)	23136 ± 534	22952 ± 467

Table 6. AISA parametrization guidelines

Parameter	Performance metric		
	Throughput	Delay	Consumption
turn_size	↑*	↓*	↑*
inc_bound	↓*	↓	↑*
dec_bound	↑*	↑	↑*
inc_rate	↑	↑	↓
dec_rate	↓	↓	↑
max_dur	↑*	↑	↓*
min_dur	↓*	↑*	↓
skip_pico (boolean)	0	0*	1*

“*” indicates the most relevant parameters for each metric.
“↑” means that increasing the parameter improves the metric performance.
“↓” means that reducing the parameter improves the metric performance.

5.7 Parametrization Summary

After several simulations, AISA parameters that significantly affect each metric performance were identified. Although absolute values are topology dependent, Table 6 presents general guidelines about parameter configuration.

6 Conclusions and Future Work

This paper presented an interpiconet scheduling algorithm herein referred to as Adaptive Interpiconet Scheduling Algorithm (AISA). AISA enables bridges to adapt to varying traffic conditions. Moreover, AISA parameters can be configured so that one chosen performance metric will be optimized. Also, the creation of explicit signaling packets was avoided by applying AISA solely to the bridges, thereby simplifying its implementation and facilitating adherence to the standard.

Three scenarios with different performance metric constraints were created to test AISA. The three chosen metrics were: traffic aggregate throughput, packet

delay and power consumption. Simulations were performed in BlueNetS, an adaptation of the ns-2 simulator. Various attempts were made to configure the parameters in order to optimize the performance of each scenario metric with minimum degradation to the other two. This goal was achieved and a set of guidelines to optimize the performance of each metric was established. Encouraged by these results, plans are underway to apply AISA to additional traffic models that would include video and web traffic for the purpose of studying performance metrics like, for example, jitter and response time.

References

1. Bluetooth Special Interest Group. <http://www.bluetooth.com>. visited nov. 2002.
2. J. Haartsen. The bluetooth Radio System. *IEEE Personal Communications*, 7(1):28–36, February 2000.
3. T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaire. Distributed Topology Construction of Bluetooth Personal Area Networks. *IEEE Infocom*, April 2001.
4. C. Law, A. Mehta, and K.-Y. Siu. Performance of a new Bluetooth Scatternet Formation Protocol. *Proc. the 2001 ACM MobiHoc*, October 2001.
5. P. Bhagwat and A. Segall. A Routing Vector Method (rvm) for Routing in Bluetooth Scatternets. *The 6th IEEE MOMUC*, November 1999.
6. M. Sun, C.K. Chang, and T.H. Lai. A Self-routing Topology for Bluetooth Scatternets. *Proc. I-SPAN 2002*, May 2002.
7. N. Johansson, F. Alriksson, and U. Jönsson. Jump Mode - A Dynamic Window-based Scheuling Framework for Bluetooth Scatternets. *Proc. 2001 ACM MobiHoc*, October 2001.
8. P. Johansson, R. Kapoor, M. Kazantzidis, and M. Gerla. Rendezvous Scheduling for Bluetooth Scatternets. *Proc. ICC 2002*, April 2002.
9. G. Tan. Self-organizing Bluetooth Scatternets. *Master Thesis*, January 2002.
10. A. Racz, G. Miklos, F. Kubinszky, and A. Valko. A Pseudo Random Coordinated Scheduling Algorithm for Bluetooth Scatternets. *Proc. 2001 ACM MobiHoc*, October 2001.
11. L. Har-Shai, R. Kofman, G. Zussman, and A. Segall. Interpiconet Scheduling in Bluetooth Scatternets. *Proc. OPNETWORK 2002 Conference*, August 2002.
12. W. Priess. Scheduling Mechanisms with Quality of Service for Bluetooth Networks. *Master Science Thesis, NCE/UFRJ*, January 2003.
13. P. Johansson, N. Johansson, U. Körner, J. Elgg, and G. Sennarp. Short Range Radio Based Ad Hoc Networking: Performance and Properties. *Proc. ICC'99*, 1999.
14. P. Johansson, R. Kapoor, M. Gerla, and M. Kazantzidis. Bluetooth: an Enabler of Personal Area Networking. *IEEE Network, Special Issue in Personal Area Networks*, October 2001.
15. E. L. Hahne. Round Robin Scheduling for Fair Flow Control in Data Communication Networks. *Phd Thesis*, December 1986.
16. The Network Simulator (ns-2). <http://www.isi.edu/nsnam/ns>, visited jun. 2002.
17. Philips Semiconductors. UAA3558 Bluetooth RF Transceiver. www.semiconductors.philips.com/technologies/bluetooth, visited sep. 2002.
18. Ericsson Microelectronics. PBA 313 01/3 Bluetooth Radio. www.ericsson.com/microe/products/bluetooth_solutions, visited sep. 2002.