Available online at www.sciencedirect.com
SCIENCE DIRECT®
Ad Hoc Networks 3 (2005) 141–155
Ad Hoc Networks
www.elsevier.com/locate/adhoc

Adaptive interpiconet scheduling for multipurpose scatternet scenarios

Werner Priess [a,*], José Ferreira de Rezende [b], Luci Pirmez [a]

[a] NCE, Universidade Federal do Rio de Janeiro, 20001-970 Rio de Janeiro, RJ, Brazil
[b] GTA/COPPE/PEE, Universidade Federal do Rio de Janeiro, 21945-970 Rio de Janeiro, RJ, Brazil

Available online 17 August 2004

Abstract

Bluetooth specification still has open issues, including the intra and interpiconet scheduling topics. This article 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 parameterization. Both features enable it to be employed in a variety of scenarios with improved performance shown 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 article.

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.

* Corresponding author.
E-mail addresses: priess@posgrad.nce.ufrj.br (W. Priess), rezende@gta.ufrj.br (J.F. de Rezende), luci@nce.ufrj.br (L. Pirmez).

1570-8705/$ - see front matter
doi:10.1016/j.adhoc.2004.07.003

This article presents an algorithm for interpiconet scheduling, referred to as AISA (Adaptive Interpiconet Scheduling Algorithm). Parameterization 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 four simulated scenarios whose focused metrics differed. The results showed that the algorithm performed well in all four situations.

This article 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 and a supporting intrapiconet scheduling algorithm; 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 μs 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 [7–13]. The first two topics are out of scope of this article.

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 [14]: 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 [15]. Gerla et al. [16] 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.

Like the previous algorithm, the Traffic-Aware Scatternet Scheduling (TASS) proposal [12] also defines a scheduling table, maintained by each master, with traffic information of all masters that its bridges are connected to. Each bridge is responsible for passing this information to the master when it switches piconet. This enables TASS to dynamically adjust each bridge service time according to its master traffic load. The algorithm has basically the same drawbacks of the previous one.

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.

A different approach was taken by Misic [13], whose work used queueing theory to compare scatternet topologies formed by master/slave and slave/slave bridges. Though not fitting into any of the proposed scheduling categories, it provides important analytical results regarding delays involved in packet transmission.

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 were created to enhance performance of a single metric, usually aggregate throughput or packet delay, and only PCSS [10] handles the power consumption metric.

4. Proposed scheduling algorithm

This section is divided in two subsections. The first one presents the Adaptive Interpiconet Scheduling Algorithm (AISA), which is the central concept of this article. Section 4.2 introduces an intrapiconet scheduling algorithm employed in the third simulated scenario from Section 5.

4.1. AISA: Adaptive Interpiconet Scheduling Algorithm

AISA differs from other interpiconet scheduling mechanisms in that it enables the choice of a performance metric to be optimized by configuring parameters that work as performance metric tuning knobs. More specifically, depending on the algorithm parameterization it is possible to prioritize one of the following 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 bridge [1] schedules its piconets in a Weighted Round Robin (WRR) [17] fashion. Time is divided up into turns of fixed time period (*turn_size* in Fig. 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 Fig. 1). It is possible to establish a lower and an upper limit for the rendezvous time period. The lower limit 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 Fig. 1). As the link occupation increases or decreases, the bridge updates the next rendezvous time period.

If the *avg_util* between the bridge and a piconet 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 Fig. 1). The bridge keeps information about the amount of free slots. 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.

[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 does not need to signal its presence.

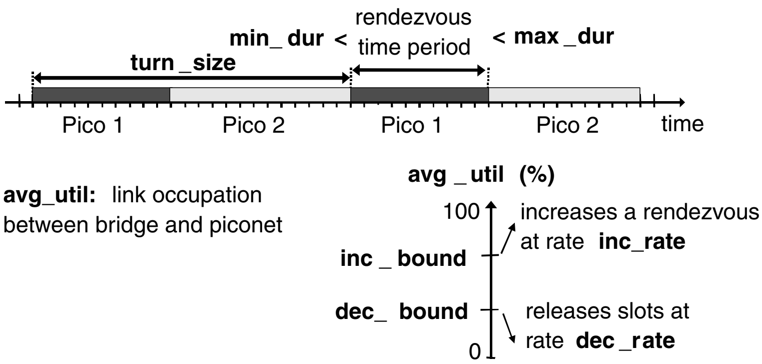

Fig. 1. AISA parameters.

Note that the *turn_size* is kept constant during all bridge operation.

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 among 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, which 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 following 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 will not schedule this piconet on the next turn, returning to schedule it on the following turn. However, a piconet may be skipped only if the boolean parameter *skip_pico* is enabled.

Table 1 summarizes the parameters that were introduced in this section. Depending on their configuration, a metric performance may be improved or not.

4.2. Supporting intrapiconet scheduling algorithm

Although intrapiconet scheduling is not the major concern of this article, it performs an

Table 1
Summary of AISA parameters

Parameter	Description
turn_size	A fixed time period during which a bridge schedules all piconets it belongs to
min_dur	Lower limit of a rendezvous time period between the bridge and a piconet
max_dur	Upper limit of a rendezvous time period between the bridge and a piconet
inc_bound	If the link utilization exceeds this parameter, the bridge tries to increase the rendezvous time period
dec_bound	If the link utilization is lower than this parameter, the bridge tries to reduce the rendezvous time period
inc_rate	Slot increasing rate employed above *inc_bound*
dec_rate	Slot decreasing rate employed under *dec_bound*
skip_pico	Boolean parameter that allows a piconet not to be scheduled in a turn

important role in the third simulated scenario, described in next section. This subsection introduces an algorithm known as Deficit Round Robin with Classes of Service (DRR-CoS). DRR-CoS was first presented in [14], in the pursue of an algorithm that would allow the use of delay-sensitive traffic, like voice, in ACL links, which were planned to support best-effort traffic. Its goal is to keep bounded the delay of voice packets and, at the same time, to maximize the aggregate throughput of best-effort traffic.

DRR-CoS is similar to Deficit Round Robin algorithm (DRR) [18] with some minor changes. It defines an interval between successive scheduling of slaves involved in delay-sensitive transmissions through a parameter called *PI* (Polling Interval). The master maintains a single value of PI. Therefore, if there are multiple delay constraints, the smallest one should be chosen for PI in order to assure QoS for all delay-sensitive traffic sources. The use of a distinct parameter for each source would probably avoid unnecessary scheduling of certain stations. However, the current approach simplifies the algorithm implementation.

DRR-CoS works on the following way. Each piconet master keeps a counter, initially set to PI, which is decremented in time until zero, at which moment the master starts scheduling slaves belonging to a list of delay-sensitive sources. After all these slaves have been addressed, the master restarts the traditional DRR scheduling mechanism from where it was interrupted.

Instead of guaranteeing precise scheduling intervals as in SCO links, DRR-CoS provides intervals close to PI. There are no exact values between these schedules because, finished a PI length interval, the master waits until the current transmission is completed, before scheduling stations belonging to the list of delay-sensitive sources. In the worst case, current transmission may last 10 slots (5 slots in each direction), allowing a delay variation of up to 6.25 ms.

5. Simulations and results

This section presents four simulated scenarios and their results. Each scenario focuses on one specific performance metric, including: 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.

Each simulation scenario was extensively run using a large number of AISA parameter configurations. In this article, we present the results for two of these configurations in each scenario. The round robin scheduling mechanism was also simulated. Certainly, it will not provide better results than AISA, but it is a simple algorithm to implement and serves as basis for comparison purposes.

We developed a Bluetooth extension to the Network Simulator (ns-2) [19], referred to as BlueNetS (Bluetooth Network Simulator). It was introduced and validated in [14], and is available in [20]. BlueNetS tool models physical and link layer Bluetooth characteristics necessary to simulate traffic communication. The connection establishment procedures were not implemented, and only static scatternet configurations were employed. 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 parameterization (see Fig. 2).

File transfer traffic (FTP) was configured from node 0 to 3, and from node 2 to 1. After each file

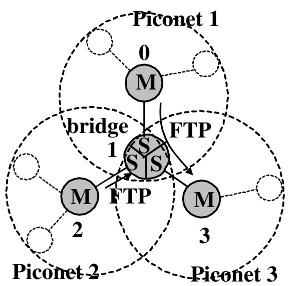

Fig. 2. Scenario 1.

Table 2
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)

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 nodes 0 and 3, these distributions have mean values of 30 Kbytes and 1 s, respectively, and between 2 and 1, 40 Kbytes and 1 s, respectively. As an example, this traffic model may represent photo file transmissions from a digital camera to a laptop or small printing jobs.

Table 2 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, since the goal of *skip_pico* is power saving, it was disabled.

5.2. Simulation results from Scenario 1

The simulation provides comparative results between two AISA configurations, referred to as AISA 1 and AISA 2, and Round Robin (RR). AISA 1 has 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, has *inc_bound* and *dec_bound* equal to 90% and 50%, respectively. The results are averages of 10 simulation runs, each of them with 120 s of simulation time. The uncertainty is expressed as 95% confidence intervals (CI).

Since different piconets are not synchronized in time, a bridge loses up to two slots while switching

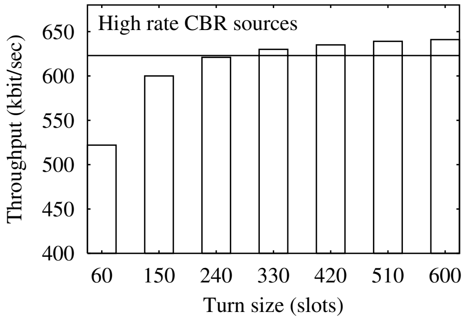

Fig. 3. Effect of turn size in aggregated throughput.

from a piconet to another one. Therefore, as the turn size gets smaller, the percentage of lost slots caused by piconet switches grows. Fig. 3 illustrates the effect of piconet switches on the aggregated throughput. Constant bit rate (CBR) sources were applied to all masters in order to keep high traffic load. The figure shows that values above 240 slots give close results in terms of throughput. However, average packet delay increases with larger values of turn size, because the bridge will remain absent of each piconet for longer time periods. So, in this scenario, the turn size was set to 240 slots (equivalent to 80 slots/piconet in the beginning of each run).

Fig. 4(a) shows the FTP aggregate throughput, measured in each subsequent one second interval, to configurations AISA 1 and Round Robin (RR). The curve oscillation was caused by the silence and transmission periods in the modeled traffic. AISA enables traffic peaks above 500 kbit/s, 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.

Fig. 4(b) shows the average aggregate throughput of FTP flows in the entire simulation. AISA 1 obtained a 15% gain over RR. This gain could be even larger with bigger files since peak periods would be 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

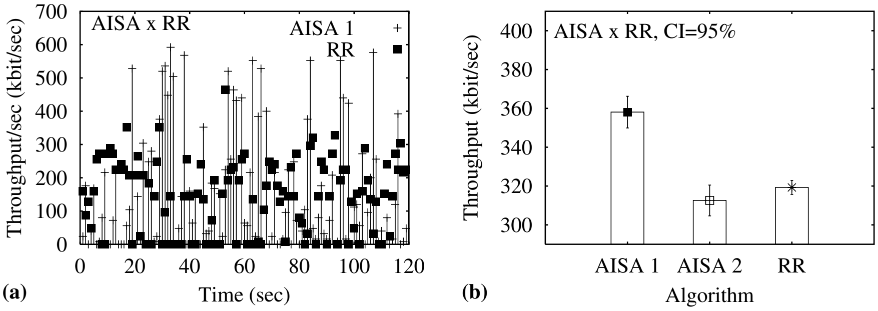

Fig. 4. Comparative results between AISA and RR algorithms: (a) throughput per second and (b) simulation average.

time period more stable, reducing AISA adaptability.

5.3. Scenario 2—Delay metric

This scenario is formed by two piconets as illustrated in Fig. 5. 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 has a slave role, configuring a master/slave bridge. 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 was modeled as a long-sized file transfer (FTP).

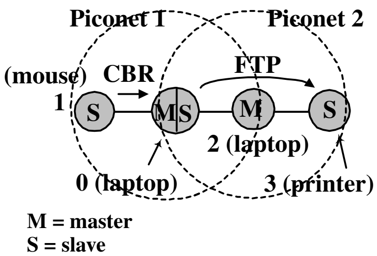

Fig. 5. Scenario 2.

Table 3 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 s 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, because of the 50-slot *min_dur*, simulations started at 120 slots (equivalent to 60 slots/piconet).

Table 3
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)

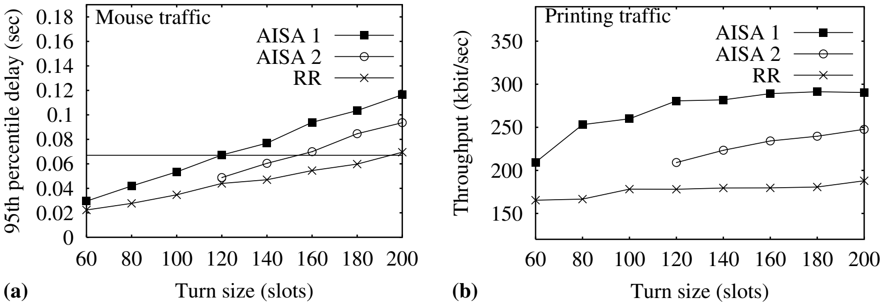

Fig. 6. Comparative results between AISA and RR, with two piconets: (a) delay comparison and (b) throughput comparison.

Fig. 6(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 the value of 65 ms as the maximum acceptable delay.

AISA 1 reaches the maximum acceptable delay when the turn size is close to 120 slots. By increasing the minimum rendezvous period, this delay limit is reached close to 150 slots (AISA 2). 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.

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 Fig. 6(b). For a *turn_size* of 120 slots, the AISA throughput result (280 kbit/s) is 60% better than the best RR result (180 kbit/s).

We have also observed the mouse packet delays when the bridgea participates in more than two piconets. The number of piconets that the bridg 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 Fig. 7.

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 time 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.

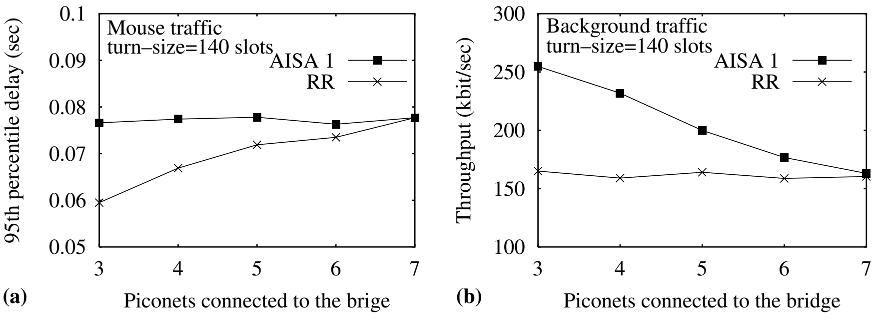

Fig. 7. AISA and RR results, increasing the number of piconets connected to the bridge: (a) delay comparison and (b) throughput comparison.

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—Delay metric

Scenario 3 was derived from Scenario 2 by adding new stations to Piconet 1, as shown in Fig. 8. One PDA (device 1) downloads files from a laptop (device 4) while the other PDA (device 2) downloads files from the master. Both file transfers were modeled as non-persistent FTP sources, whose file size and time interval between files follow exponential distributions with mean values of 30 Kbytes and 1 s, respectively. Now, mouse-generated traffic competes for master scheduling share not only with printing jobs (interpiconet traffic), but also with intrapiconet traffic. As in the previous scenario, we are looking for the configuration that bounds the delay for mouse-generated packets, whilst maximizing the aggregate throughput of background traffic.

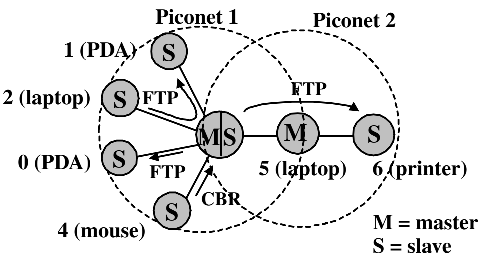

Fig. 8. Scenario 3.

What distinguishes this scenario from others is the fact that it integrates both intrapiconet and interpiconet scheduling algorithms.

The interpiconet scheduling algorithms tested in Scenario 3 were AISA with *min_dur* parameter set to 20 slots and other parameters following Table 3, and Round Robin (RR). The intrapiconet scheduling algorithms employed in piconet 1 were Round Robin (RR), Deficit Round Robin (DRR) and Deficit Round Robin with Classes of Service (DRR-CoS).

5.6. Simulation results from Scenario 3

Simulation conditions were the same of Scenario 2, including the number of simulation runs and their duration.

Fig. 9(a) shows the 95th percentile delay of mouse-generated packets for various *turn_size* values. Each curve represents one combination of intrapiconet and interpiconet scheduling algorithms. Since traffic load in piconet 1 is high, the bridge tends to divide equally its presence in both piconets. Therefore, both printing job and intrapiconet traffic affect mouse transmissions.

DRR presents the highest delay values for mouse packets, because slave queues in Piconet 1

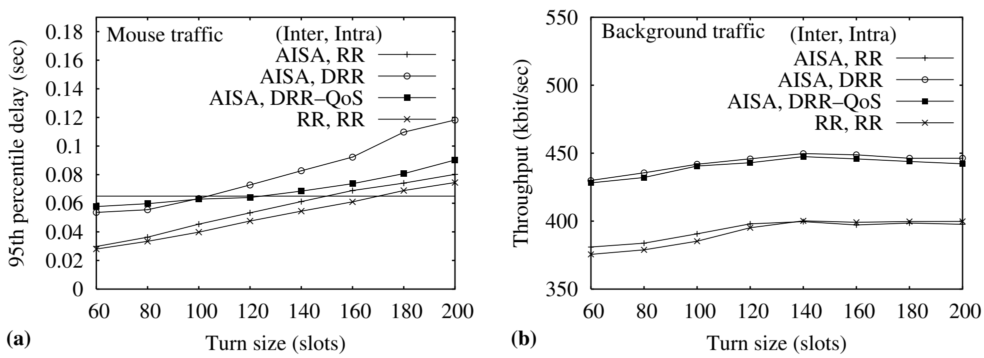

Fig. 9. AISA and RR results, increasing the number of piconets connected to the bridge: (a) delay comparison and (b) throughput comparison.

always have packets to transmit, except for the mouse, which generates packets at low rate. Results for combinations (RR, RR) and (AISA, RR) are alike, because the high traffic load between the bridge and both piconets makes AISA to perform like a RR scheduling algorithm.

DRR-CoS has the concern of keeping mouse traffic delay limited. Therefore, it presents the least inclined curve. The 65 ms upper bound is reached with the turn size of 120 slots. This is the same result obtained in Scenario 2 for AISA 1 configuration, meaning that the joint use of AISA for interpiconet scheduling and DRR-CoS for intrapiconet scheduling is a good choice to honor delay constraints.

Regarding aggregate throughput, Fig. 9(b) shows two distinct results. Configurations that use DRR-based algorithms for intrapiconet scheduling provide over 10% gain compared to RR-based ones. This happens because RR schedules stations equally, regardless of source rates.

Considering delay and throughput results exposed, the configuration (AISA, DRR-CoS) outperforms other scheduling algorithm configurations. Also, depending on the scatternet topology, intrapiconet scheduling may be as important to overall performance as interpiconet scheduling.

5.7. Scenario 4—Power consumption metric

The goal of Scenario 4 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.

The topology is a scatternet formed by nine piconets as illustrated in Fig. 10. Only the border piconets have data sources. There are three types of data sources, representing different sensoring devices. Each piconet has one source from each type. The central node is an access point, which is the sink for all sources. Sources generate packets at a 3 kbit/s constant rate. Types 1–3 use 300-, 100- and 20-byte packet length, respectively, resulting in different packet intervals.

Table 4 shows the parameters that remained unchanged throughout this simulation set. The minimum rendezvous time period (*min_dur*) should be small so that a bridge stays less time in piconets with low link occupation. The value of *dec_rate* should be configured so to allow a fast slot release from a rendezvous. The parameter *skip_pico* enables power saving. Therefore, it should be activated.

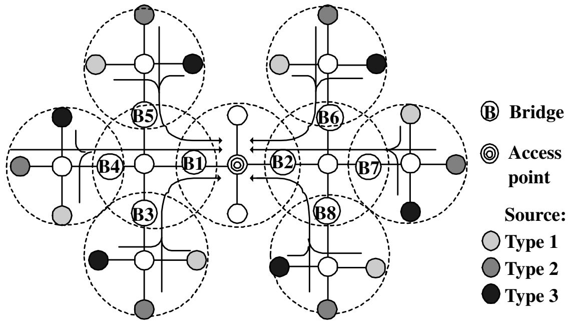

Fig. 10. Scenario 4.

Table 4
Fixed AISA parameters in Scenario 4

min_dur	*dec_rate*	*skip_pico*
20 slots	20%	1 (bool)

In order to measure power consumption a few considerations were made. In Bluetooth, a packet transmission dissipates almost the same power of a packet reception [21,22]. Three levels of consumption were adopted: the most consuming one, during a packet transmission; the intermediate one, during Active mode, but without any transmission; and the least consuming one, during Hold mode. By taking this information into account, we defined an energy unit (e.u.). One e.u. is the transmission power of a one-slot packet. [2]

5.8. Simulation results from Scenario 4

Two AISA configurations and Round Robin (RR) were simulated. RR does not save power but it was simulated for packet delay comparison.

[2] More details about power levels used in this article may be obtained in [14].

Table 5
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

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 5.

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 s 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.

There are eight bridges in this scenario. The bridges are divided in two groups, regarding their position in the network topology: Group 1, with Bridges 1 and 2, and Group 2, with Bridges 3–8. 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. Fig. 11 presents both groups dissipated power in this simulation. [3] For each value of turn size, the graphs present the total dissipated power by each group and the power dissipated by transmitting and receiving packets.

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 in the minimum rendezvous time period (*min_dur*) in each piconet, because of the low traffic condition. 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 (Fig. 11). In AISA 2, it is easier for a bridge to release slots (because $dec\_bound_{\text{AISA2}} > dec\_bound_{\text{AISA1}}$ in Table 5), and more difficult to increase a rendezvous time period (because $inc\_bound_{\text{AISA2}} > inc\_bound_{\text{AISA1}}$). 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. Fig. 12 shows the average packet delay for the three source types. The three types presented similar results. Comparing the scheduling mechanisms, RR algorithm caused the smallest delay values, but without power saving. AISA 2 presented larger delay values than AISA 1, but, as showed previously in Fig. 11, AISA 2 has proven to be the least power consuming configuration. 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 4 was tested with one different parameterization for each bridge group. Table 6

[3] 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 70,461 ± 1023 e.u., representing about 30% more power than with AISA 1.

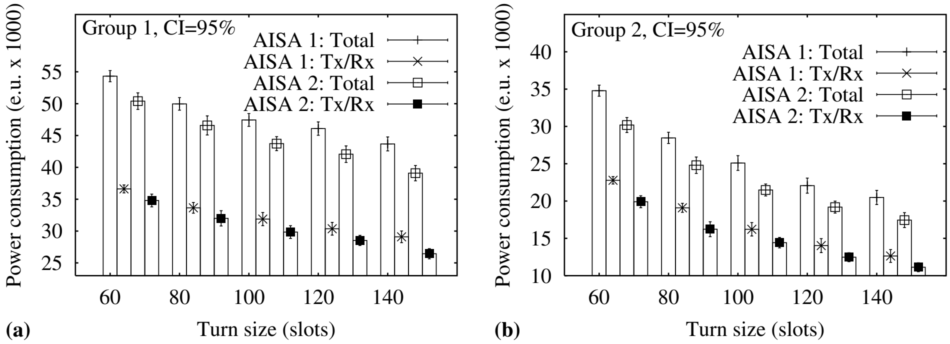

Fig. 11. Power consumption of bridge Groups 1 and 2, with configurations 1 and 2: (a) group 1 power consumption and (b) group 2 power consumption.

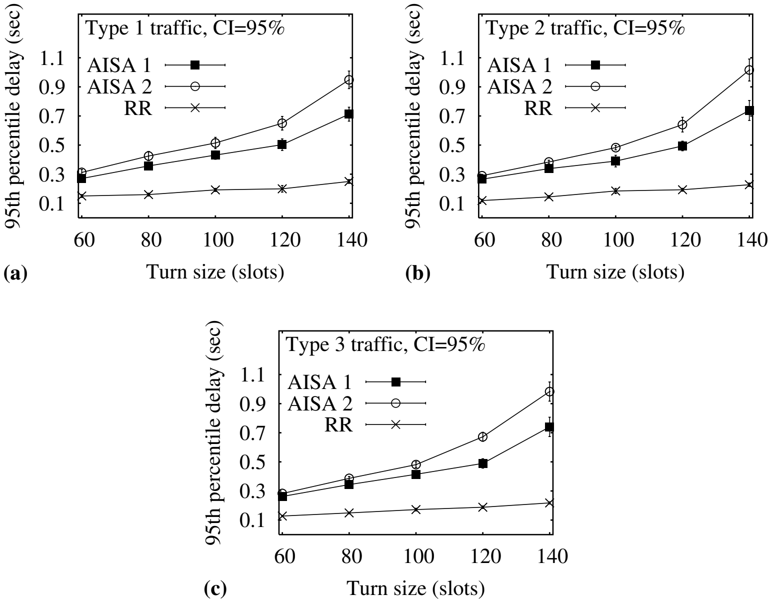

Fig. 12. Packet delay results: (a) Type 1, (b) Type 2 and (c) Type 3.

summarizes both configurations. Note that Group 1 was configured similarly to AISA 2 described above in order to minimize power consumption by this group. However, packet 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 7 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 source types was also calculated. The results are presented in Table 8. 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 6
AISA parameterization to equalize power consumption

	turn_size	*inc_bound*	*dec_bound*	*inc_rate*	*max_dur*
Group 1	140 slots	90%	70%	10%	80 slots
Group 2	60 slots	80%	60%	20%	*turn_size* slots

Table 7
Power consumption in the case of distinct configurations for Groups 1 and 2

	Group 1	Group 2
Total consumption (e.u.)	35,134 ± 775	34,436 ± 591
Consumption of transmitted and received packets (e.u.)	23,136 ± 534	22,952 ± 467

Table 8
Packet delay in case of distinct configurations for Groups 1 and 2

	Type 1	Type 2	Type 3
Avg delay (s)	0.627 ± 0.015	0.57 ± 0.032	0.58 ± 0.024

5.9. Parametrization summary

After several simulations, the AISA parameters that significantly affect each metric performance were identified. Although absolute values are topology dependent, Table 9 presents general guidelines about parameter configuration.

Table 9
AISA parameterization guidelines

Parameter	Performance metric		
	Throughput	Delay	Consumption
turn_size	↑*	↓*	↑*
max_dur	↑*	↑	↓*
min_dur	↓*	↑*	↓
inc_bound	↓*	↓	↑*
dec_bound	↑*	↑	↑*
inc_rate	↑	↑	↓
dec_rate	↓	↓	↑
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.

6. Conclusions and future work

This article presented an interpiconet scheduling algorithm herein referred to as Adaptive Interpiconet Scheduling Algorithm (AISA). AISA enables the 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.

Four 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> (accessed March 2003).
[2] J. Haartsen, The Bluetooth radio system, IEEE Personal Communications 7 (2000) 28–36.
[3] T. Salonidis, P. Bhagwat, L. Tassiulas, R. LaMaire, Distributed topology construction of Bluetooth personal area networks, in: IEEE Infocom, April 2001.
[4] C. Law, A. Mehta, K.-Y. Siu, Performance of a new Bluetooth scatternet formation protocol, in: Proc. 2001 ACM MobiHoc, October 2001.

[5] P. Bhagwat, A. Segall, A routing vector method (RVM) for routing in Bluetooth scatternets, in: The 6th IEEE MOMUC, November 1999.
[6] M. Sun, C. Chang, T. Lai, A self-routing topology for Bluetooth scatternets, in: Proc. I-SPAN 2002, May 2002.
[7] N. Johansson, F. Alriksson, U. Jönsson, Jump mode—A dynamic window-based scheduling framework for Bluetooth scatternets, in: Proc. 2001 ACM MobiHoc, October 2001.
[8] P. Johansson, R. Kapoor, M. Kazantzidis, M. Gerla, Rendezvous scheduling for Bluetooth scatternets, in: Proc. ICC 2002, April 2002.
[9] G. Tan, Self-organizing Bluetooth scatternets, Master Thesis, January 2002.
[10] A. Racz, G. Miklos, F. Kubinszky, A. Valko, A pseudo random coordinated scheduling algorithm for Bluetooth scatternets, in: Proc. 2001 ACM MobiHoc, October 2001.
[11] L. Har-Shai, R. Kofman, G. Zussman, A. Segall, Inter-piconet scheduling in Bluetooth scatternets, in: Proc. OPNETWORK 2002 Conf., August 2002.
[12] J.-P. Sheu, C.-H. Cheng, K.-P. Shih, S.-C. Tu, A traffic-aware scheduling for Bluetooth scatternets, in: The Fifth European Wireless Conf., February 2004.
[13] V.B. Misic, J. Misic, Performance of Bluetooth bridges in scatternets with exhaustive service scheduling, in: Proc. 36th Int. Conf. on System Sciences (HICSS'03), January 2003.
[14] W. Priess, Scheduling mechanisms with quality of service for Bluetooth networks, Master Science Thesis, NCE/UFRJ, January 2003.
[15] P. Johansson, N. Johansson, U. Körner, J. Elgg, G. Svennarp, Short range radio based ad hoc networking: performance and properties, in: Proc. ICC'99, 1999.
[16] P. Johansson, R. Kapoor, M. Gerla, M. Kazantzidis, Bluetooth: an enabler of personal area networking, IEEE Network, Special Issue in Personal Area Networks 15 (5) (2001) 28–37.
[17] E.L. Hahne, Round Robin scheduling for fair flow control in data communication networks, PhD Thesis, December 1986.
[18] M. Shreedhar, G. Varghese, Efficient fair queueing using deficit round-robin, in: Proc. of the ACM Sigcomm, September 1995.
[19] The Network Simulator (ns-2), <http://www.isi.edu/nsnam/ns> (accessed June 2003).
[20] Bluetooth Network Simulator, <http://www.gta.ufrj.br/BlueNetS> (accessed December 2003).
[21] Philips Semiconductors, UAA3558 Bluetooth RF Transceiver, <www.semiconductors.philips.com/technologies/bluetooth> (accessed January 2003).
[22] Ericsson Microeletronics, PBA 313 01/3 Bluetooth Radio, <www.ericsson.com/microe/products/bluetooth_solutions> (accessed February 2003).

Werner Priess received his B.Sc. in Computer Engineering from the Instituto Militar de Engenharia (IME), Rio de Janeiro, in 1997 and his M.Sc. in Computer Science from Universidade Federal do Rio de Janeiro (UFRJ) in 2003. He is currently a D.Sc. student in UFRJ. His research interests include quality of service and mobility issues in wireless networks and sensor networks.

José Ferreira de Rezende received the B.Sc. and M.Sc. degrees in Electronic Engineering from the Universidade Federal do Rio de Janeiro (UFRJ) in 1988 and 1991, respectively. He received the Ph.D. degree in Computer Science from the Université Pierre et Marie-Curie, Paris, France in 1997. He was an associate researcher at LIP6 (Laboratoire d'Informatique de Paris 6) during 1997. Since 1998 he is an Associate Professor at UFRJ. His research interests are in distributed multimedia applications, multipeer communication, performance evaluation and QoS aspects of high-speed, wireless and sensor networks.

Luci Pirmez received a B.Sc degree on Computer Science in 1981, a M.Sc. degree on Computer Science in 1986 and the D.SC degree on Computer Science in 1996 from the Universidade Federal do Rio de Janeiro, Brazil. She is a member of research staff of the Computer Center of Universidade Federal do Rio de Janeiro. Her research interests include wireless networks, wireless sensor networks, networks management and security and formal description techniques for communication protocols.