

Um Mecanismo de Escalonamento Parametrizável para Scatternets Bluetooth

Werner Priess¹, José Ferreira de Rezende²,
Luci Pirmez¹, Luiz Fernando Rust da Costa Carmo¹

¹Núcleo de Computação Eletrônica – NCE/UFRJ

²Grupo de Teleinformática e Automação – GTA/COPPE/UFRJ

wpriess@posgrad.nce.ufrj.br, rezende@gta.ufrj.br
{luci,rust}@nce.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: (1) its adaptability to varying network traffic conditions; and (2) its ability to optimize specific performance metrics via parametrization. Both features enable this algorithm to be employed in a variety of scenario settings, and three of them were simulated. The results showed that AISA performs efficiently in this context.*

Keywords: *wireless communication, ad hoc networks, Bluetooth, scatternet, QoS.*

Resumo. *A especificação Bluetooth ainda apresenta questões em aberto, incluindo os mecanismos de escalonamento das estações. Este trabalho apresenta um algoritmo de escalonamento interpiconet, denominado AISA (Adaptive Interpiconet Scheduling Algorithm). AISA é um algoritmo adaptativo e parametrizável. A primeira característica permite o ajuste das estações às condições do tráfego. A segunda possibilita que uma determinada métrica de desempenho seja priorizada. Estas duas características permitem que este algoritmo seja empregado em cenários diferentes. O algoritmo proposto foi testado em três cenários através de simulações, apresentando bons resultados em todos os casos.*

Palavras-chave: *comunicação sem fio, redes ad hoc, Bluetooth, scatternet, QoS.*

1. Introdução

Dentre as tecnologias de rádio existentes, Bluetooth reúne as condições necessárias para a implementação de diversos cenários de computação ubíqua. Nesses cenários, diversos dispositivos sem fio, fixos ou móveis, comunicam-se de forma espontânea, sem a necessidade da intervenção do usuário. As interfaces Bluetooth têm dimensão reduzida, baixo consumo de energia e seu custo vem baixando. Sua popularidade está crescendo e já existem diversos produtos comerciais disponíveis.

A rede formada com Bluetooth é chamada *piconet*, que pode conter até oito dispositivos. Visando o aumento do alcance e da quantidade de estações da rede, definiu-se a *scatternet*. A *scatternet* é a rede composta por várias *piconets*, interligadas através de uma

ou mais estações (aqui denominadas pontes) pertencentes, alternadamente, a múltiplas *piconets*.

O estudo das *scatternets* é recente e ainda há questões em aberto, necessitando de padronização. Este trabalho propõe uma solução para o escalonamento *interpiconet*, que é a forma como as pontes dividem o seu tempo entre as múltiplas *piconets* de que fazem parte. A ponte é a estação responsável por encaminhar o tráfego entre *piconets* distintas e seu funcionamento influi diretamente na vazão e no retardo dos pacotes desse tráfego, bem como no consumo de energia da rede.

O algoritmo de escalonamento *interpiconet* proposto, chamado AISA (*Adaptive Interpiconet Scheduling Algorithm*), é parametrizável e atua somente nas pontes. A parametrização permite que as pontes sejam configuradas para favorecer, primordialmente, uma métrica de desempenho pré-determinada como, por exemplo, a vazão agregada, o retardo dos pacotes ou a economia de energia nas pontes. O fato do AISA atuar somente nas pontes busca minimizar a quantidade de alterações na especificação Bluetooth. O AISA foi avaliado através de simulações em cenários com requisitos distintos. Os resultados mostram que o algoritmo teve um bom desempenho em todos os casos.

O artigo está estruturado da seguinte maneira. A próxima seção apresenta conceitos básicos sobre a especificação Bluetooth. A Seção 3 resume os principais trabalhos relacionados ao escalonamento *interpiconet*. A Seção 4 apresenta o algoritmo AISA. A Seção 5 descreve os cenários de simulação e os resultados obtidos. Finalmente, as conclusões são apresentadas na Seção 6.

2. Especificação Bluetooth

Bluetooth [1, 2] é um padrão de interface de rádio de curto alcance, criado para permitir a comunicação entre dispositivos eletrônicos sem a utilização de cabos de conexão, que pode ser empregado, também, na formação de redes *ad hoc*. O Bluetooth SIG (*Special Interest Group*) [1] lançou uma especificação industrial aberta, contendo duas partes: o núcleo e os perfis. O núcleo define as características do rádio e a pilha de protocolos para a comunicação entre dois dispositivos. Os perfis especificam quais protocolos da pilha devem ser implementados para determinada aplicação.

2.1. Camadas Física e de Enlace Bluetooth

O rádio Bluetooth opera na banda ISM (*Industrial, Scientific and Medical*) de 2.4 GHz e emprega a técnica de saltos de frequência com espalhamento espectral (FHSS) na transmissão. O espectro de frequências é dividido em 79 canais de rádio-frequência (23 em alguns países), cada um com 1 MHz de largura de banda, e a frequência de saltos é de 1600 por segundo.

A banda base é a camada física do Bluetooth, sendo responsável, entre outras funções, pela criação das *piconets* e dos enlaces. A *piconet* é uma rede com até oito integrantes ativos, onde os dispositivos compartilham o mesmo esquema de saltos de frequência. Um dos dispositivos assume o papel de mestre e os outros se comportam como escravos. O mestre de uma *piconet* é quem dita a seqüência de saltos e a fase nesta seqüência. Os saltos de frequência ocorrem a cada 625 microssegundos e esse intervalo de tempo é chamado de *slot*.

A comunicação bidirecional ocorre através de duplexação por divisão no tempo (TDD - *time division duplex*), onde os dispositivos transmitem e recebem alternadamente. Um escravo somente poderá transmitir em um *slot* se tiver sido endereçado pelo mestre no *slot* anterior. A cada novo *slot*, os dispositivos de uma mesma *piconet* mudam para a próxima frequência da seqüência de saltos. Além do esquema de saltos de frequência, a banda base dispõe de técnicas de retransmissão de pacotes e de correção de erros, para fornecer robustez e confiabilidade ao Bluetooth.

Há dois tipos de enlaces físicos possíveis em uma *piconet*: síncrono, orientado a conexão (SCO) e assíncrono, sem conexão (ACL). O SCO é um enlace ponto-a-ponto, simétrico, entre o mestre e um escravo, mantido pelo mestre em *slots* reservados a intervalos regulares. O ACL é um enlace ponto-a-multiponto, entre o mestre e todos os escravos da *piconet*, ocupando os *slots* não reservados para o enlace SCO.

Como uma das preocupações do Bluetooth é a economia de energia, existem três modos de operação de baixo consumo de energia: *sniff*, *hold* e *park*. No modo *sniff*, o escravo dorme durante um intervalo pré-definido e acorda, periodicamente, para escutar as transmissões do mestre. No modo *hold*, o escravo entra em modo de baixo consumo por um intervalo de tempo fixo, após o qual volta à atividade normal. Finalmente, no modo *park*, o escravo libera seu endereço na *piconet*, mas permanece sincronizado à rede, para uma possível reativação futura.

LMP (*Link Manager Protocol*) e L2CAP (*Logical Link Control and Adaptation Protocol*) compõem a camada de enlace de dados Bluetooth. LMP é responsável pela configuração e gerenciamento das conexões banda base. Por exemplo, os modos de operação de baixo consumo de energia são definidos na banda base, mas são configurados e ativados através de transações LMP entre dois dispositivos. L2CAP faz a interface com os protocolos de nível superior. O protocolo usa o conceito de canais lógicos para os fluxos. L2CAP realiza a segmentação e remontagem de pacotes (SAR), função necessária para permitir a utilização, por parte das camadas superiores, de pacotes maiores que o tamanho máximo aceito pela banda base.

2.2. As Scatternets

Se dispositivos de *piconets* distintas quiserem se comunicar, pode haver a interconexão dessas redes, criando uma *scatternet*. A *scatternet* se forma quando ao menos uma estação, chamada ponte, participa de duas ou mais *piconets*. Como normalmente as estações possuem apenas uma interface Bluetooth, a ponte não pode estar ativa em duas *piconets* simultaneamente. Por isso, ela divide, no tempo, sua participação nas múltiplas *piconets*, tarefa essa chamada de escalonamento *interpiconet*. A ponte pode desempenhar o papel de escravo em várias *piconets*, mas somente pode ser mestre em uma.

A ponte pode trabalhar com aplicações independentes em cada *piconet* ou encaminhar pacotes de uma *piconet* para outra. O primeiro caso está exemplificado na Figura 1 (a), onde uma impressora é compartilhada por duas *piconets* distintas. A Figura 1 (b) apresenta dois exemplos do segundo caso, sendo um a transmissão de vídeo entre *piconets* e o outro a comunicação de voz na *scatternet*.

Ao mesmo tempo em que propiciam versatilidade ao Bluetooth, as *scatternets* trazem consigo novas questões relativas à sua implementação. Estas questões têm sido foco de diversas pesquisas. As principais pesquisas relacionadas ao escalonamento *interpiconet* são abordadas na próxima seção.

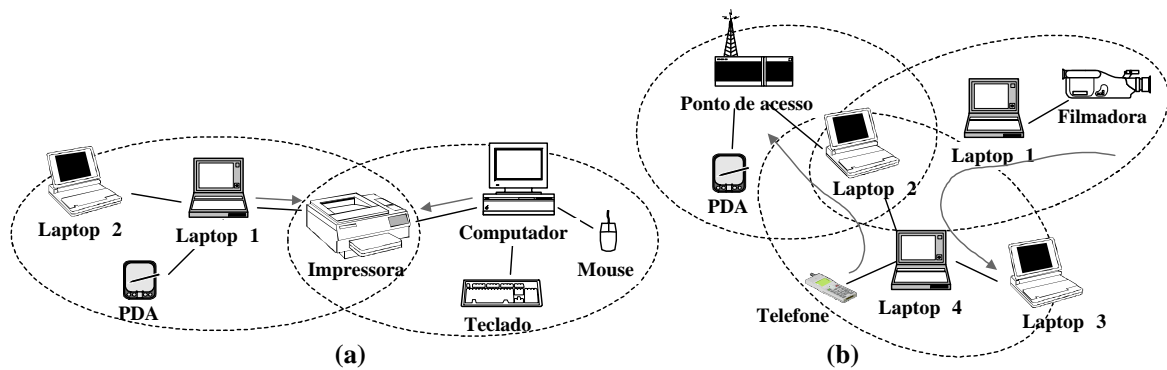


Figura 1: Exemplos de cenários de scatternets.

3. Trabalhos Relacionados

A inexistência de padronização relacionada a vários aspectos das *scatternets* têm possibilitado várias pesquisas, principalmente nos seguintes tópicos: formação da topologia [3, 4], roteamento ou encaminhamento de pacotes [5, 6] e escalonamento *interpiconet* [7, 8, 9, 10, 11]. Esta seção aborda o escalonamento *interpiconet*.

O grande desafio do escalonamento *interpiconet* é garantir que o tráfego atravessa a ponte da maneira mais eficiente possível, sem, contudo, degradar o tráfego interno às *piconets*. No funcionamento ideal, a ponte sempre estará presente em uma *piconet* no momento em que o mestre a escalonar. Além disso, o mestre deve retirar a ponte de seu processo de *polling*, quando esta estiver ausente (conectada a outra *piconet*).

Os algoritmos de escalonamento *interpiconet* podem ser divididos, basicamente, em duas categorias, segundo a responsabilidade pela coordenação dos mesmos [12]: mecanismos com decisão isolada e com decisão distribuída. No mecanismo com decisão isolada, a ponte decide unilateralmente sobre sua presença nas *piconets* e pode comunicar (ou não) essa decisão aos mestres das *piconets* de que faz parte como escravo. Esses algoritmos são mais simples, requerendo poucas modificações na especificação Bluetooth. No mecanismo com decisão distribuída, as decisões sobre os futuros encontros entre a ponte e os mestres das *piconets* de que participa como escravo ocorrem através de acordos entre as partes. Estes acordos possibilitam ganhos em termos de vazão agregada da *scatternet*, mas requerem novas mensagens de sinalização, aumentando a complexidade do algoritmo.

Os trabalhos iniciais abordam o escalonamento em *scatternets* de uma forma genérica, partindo de topologias muito simplificadas [13], preocupando-se em estudar o comportamento do tráfego e não exatamente propondo algoritmos de escalonamento. O artigo de Johansson *et al.* [14] apresenta, de forma teórica, o termo “ponto de encontro” (*rendezvous point*) como sendo o *slot* em que a ponte e o mestre de uma *piconet* decidem se comunicar. O artigo também define a “janela de encontro” (*rendezvous window*), que representa a duração do encontro.

Dentre as propostas com decisão distribuída, Johansson *et al.* [7] apresentam uma arquitetura para o escalonamento em *scatternets*, centrada em um novo modo de funcionamento, o modo JUMP. Uma estação em modo JUMP em uma *piconet* está, por padrão, ausente desta rede. Com isso, a ponte pode alternar entre *piconets* sem precisar comunicar o fato a cada troca. A incorporação de um novo modo de funcionamento ao controlador de enlace Bluetooth pode não ser viável.

O trabalho de Kapoor *et al.* [8] apresenta outro mecanismo de decisão distribuída baseado no conceito de ponto de encontro (PE) acima definido. O mestre mantém uma lista dos PE das pontes com sua *piconet* e uma lista dos PE de suas pontes com as outras *piconets* a que pertencem. Estas informações permitem ao mestre otimizar a alocação de novos pontos de encontro. Os autores não deixam claro como essas informações são passadas ao mestre. O algoritmo exige que muitas mensagens sejam trocadas para manter os mestres atualizados.

A proposta de escalonamento em topologia estruturada em árvore *Tree Scatternet Scheduling* (TSS) [9] foi elaborada para funcionar de forma integrada com o mecanismo de formação *Tree Scatternet Formation*, dos mesmos autores. O emprego da topologia em árvore facilita o escalonador e torna mais fácil a obtenção de uma coordenação global entre as *piconets*. Por outro lado, a aplicabilidade do algoritmo fica restrita a algumas topologias específicas, sem ciclos.

Quanto às propostas com decisão isolada, Racz *et al.* apresentam o algoritmo *Pseudo-Random Coordinated Scatternet Scheduling* (PCSS) [10]. Cada estação escolhe, pseudo-aleatoriamente, um ponto de encontro baseado no relógio do mestre e no endereço do escravo. Cada par mestre-escravo obterá os mesmos pontos de encontro, diferentes dos outros pares. A principal vantagem do PCSS é a coordenação entre as estações sem a necessidade de sinalização explícita. Entretanto, com o aumento da quantidade de estações, os pontos de encontro começam a colidir.

Har-Shai *et al.* propõem o *Load Adaptive Algorithm* (LAA) [11] para ser executado apenas nas pontes. Nesse algoritmo, a ponte se adapta a variações no tráfego observando suas filas de saída e recebendo a informação das filas dos mestres em sua direção. Para que o mestre passe a informação de sua fila em direção à ponte, é necessário um campo adicional nos pacotes. O trabalho não deixa claro como essa informação é passada. No estágio atual, o LAA é aplicável somente a pequenas *scatternets*, nas quais a ponte é conectada apenas a dois mestres.

Em resumo, a maioria das propostas apresenta as seguintes limitações. A ponte pode pertencer somente a duas *piconets*, funcionando sempre como escravo. A quantidade de estações de uma *scatternet* é limitada. As propostas avaliam as métricas de vazão agregada das *scatternet* e de retardo dos pacotes, porém, apenas a proposta PCSS [10] tece considerações sobre a eficiência da ponte em relação ao consumo de energia.

4. Algoritmo de Escalonamento Interpiconet Proposto: AISA

O AISA (*Adaptive Interpiconet Scheduling Algorithm*) é um algoritmo de escalonamento adaptativo e parametrizável que se diferencia das demais propostas de escalonamento *interpiconet*, por permitir a escolha da métrica de desempenho que se deseja otimizar. Mais especificamente, dependendo da parametrização do algoritmo, pode-se otimizar uma das seguintes métricas: a vazão agregada do tráfego, o retardo dos pacotes e o consumo de energia das pontes. Além disso, o escalonador mantém a justiça na ocupação do enlace entre fluxos de características similares.

O algoritmo AISA é executado somente nas pontes, às quais cabem as decisões sobre a forma de alternância entre as *piconets*. Dessa forma, evita-se a criação de pacotes de sinalização específicos para o escalonamento, possibilitando minimizar as modificações na especificação Bluetooth. De acordo com a classificação proposta na seção anterior, o AISA

encontra-se na categoria dos mecanismos de escalonamento de decisão isolada. O restante desta seção explica o funcionamento do AISA em termos de seus parâmetros.

A ponte ¹ escalona as suas *piconets* de uma forma similar ao *weighted round robin* (WRR) [15]. Ela divide o tempo em turnos de duração fixa (**turn_sz** na Figura 2). Em cada turno, a ponte passa exatamente uma vez em todas as *piconets* de que participa, permanecendo um tempo pré-determinado em cada uma delas (duração do encontro na Figura 2). O limite inferior da duração de um encontro é chamado **min_dur** e o limite superior, **max_dur**.

Ao sair de uma *piconet*, a ponte calcula a duração do encontro com essa *piconet* para o próximo turno, baseado na fórmula $avg_util = \alpha \times avg_util_ant + (1 - \alpha) \times curr_util$. Nessa fórmula, a ocupação do enlace entre a ponte e a *piconet* (**avg_util** na Figura 2) depende do percentual de *slots* ocupados por pacotes de dados no encontro atual (**curr_util**) e da média dos turnos anteriores (**avg_util_ant**). O parâmetro α é um valor entre 0 e 1, representando a influência do passado no cálculo da média atual. Dependendo da variação de **avg_util**, a duração do próximo encontro da ponte com a *piconet* pode aumentar ou diminuir.

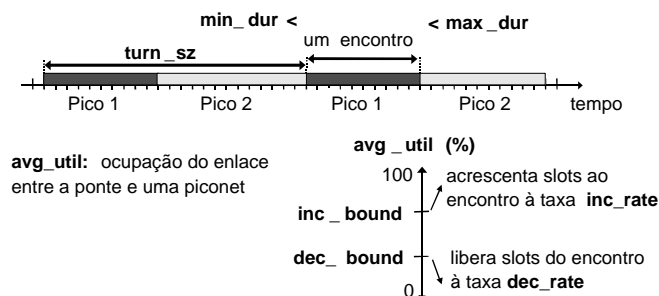


Figura 2: Principais parâmetros do AISA.

Se **avg_util** decrescer ao longo do tempo, atingindo valores abaixo do limite **dec_bound**, a ponte tentará reduzir a duração do encontro, liberando *slots* à taxa **dec_rate** (ver Figura 2). Por outro lado, se **avg_util** exceder o limite **inc_bound**, a ponte tentará aumentar a duração do próximo encontro, acrescentando *slots* à taxa **inc_rate**. Note que a duração do turno (**turn_sz**) mantém-se fixa ao longo de toda a operação da ponte.

É possível que a ponte tente aumentar a duração do encontro com uma *piconet*, mas não haja *slots* livres suficientes. Nesse caso, a ponte tenta retirar *slots* da *piconet* com o encontro mais longo. Entretanto, o encontro do qual se retiram *slots* não pode ficar com a duração menor do que o solicitante. Essa condição visa garantir a justiça na distribuição do turno entre todas as *piconets*.

As pontes se comunicam com o mestre de cada *piconet* através da sinalização LMP prevista para o modo de baixo consumo HOLD. Quando a ponte está prestes a deixar a *piconet* atual, ela envia um pacote LMP_Hold_Req para o mestre da rede, informando o instante em que ela entrará em modo HOLD e a duração deste estado. Assim, o mestre fica ciente da ausência da ponte e retira-a do escalonamento *intrapiconet* durante o período combinado.

AISA possui três maneiras de economizar energia, com a entrada das pontes em mo-

¹Todas as explicações desta seção consideram a ponte como um escravo nas *piconets* em que participa. Caso a ponte funcione como mestre, ela tem o controle da *piconet* e não precisa sinalizar sua saída.

do HOLD. Na primeira, conforme o tráfego das *piconets* com a ponte diminui, *slots* são liberados. Nos *slots* que sobram ao final de cada turno, a ponte entra em modo de baixo consumo HOLD. A segunda maneira é caracterizada pela saída antecipada da ponte de uma *piconet*, o que ocorre quando não há mais dados a transmitir no enlace. Como são perdidos alguns *slots* com a negociação do HOLD, a saída antecipada da ponte só é compensatória se o tempo restante na *piconet* for maior que um determinado limite (**early_exit**). Finalmente, a terceira maneira ocorre quando a ponte, mesmo permanecendo o tempo mínimo permitido em uma *piconet* (**min_dur**), não tem dados a transmitir ou receber. Então, a ponte pode deixar de escalonar esta *piconet* no próximo turno, voltando a escaloná-la no turno seguinte ao próximo. Para permitir esta forma de economia de energia, é necessário que o parâmetro booleano **skip_pico** esteja habilitado.

Em resumo, os seguintes parâmetros foram introduzidos: **turn_sz**, **min_dur**, **max_dur**, **inc_bound**, **dec_bound**, **inc_rate**, **dec_rate**, **early_exit** e **skip_pico**. Dependendo de suas configurações, uma métrica de desempenho pode ser beneficiada ou não. A próxima seção avalia o efeito dos parâmetros no desempenho das scatternets.

5. Simulações e Resultados

Esta seção apresenta os resultados do funcionamento do AISA para três cenários de simulações. Cada cenário prioriza uma métrica de desempenho distinta, onde as métricas são: vazão agregada, retardo dos pacotes e consumo de energia. Dependendo da métrica avaliada, alguns parâmetros são mantidos fixos e outros são variáveis, buscando-se a configuração que otimiza a referida métrica.

Todas as simulações foram realizadas com o BlueNetS (*Bluetooth Network Simulator*), uma extensão para o simulador de redes ns-2 [17], apresentada pelos autores em [12, 16]. BlueNetS implementa as principais funcionalidades das camadas física e de enlace Bluetooth, necessárias à execução dessas simulações. Quanto às camadas superiores, foram utilizados os protocolos TCP, UDP e IP e as aplicações existentes no próprio simulador ns-2.

5.1. Cenário 1 - Métrica: Vazão

No Cenário 1, os parâmetros do AISA foram ajustados, buscando maximizar a vazão dos fluxos entre *piconets*. Foi criada uma *scatternet*, composta por três *piconets*, compartilhando uma estação ponte. A Figura 3 ilustra o cenário.

Existem fluxos de transferência de arquivos, através de FTP, de 0 para 3 e de 2 para 1. Após a transmissão de cada arquivo, ambas as fontes FTP aguardam um intervalo, antes de iniciarem a transmissão do arquivo seguinte. O protocolo TCP empregado é não-persistente, pois a transmissão de cada novo arquivo é iniciada com o TCP em *slow start*. Tanto o tamanho do arquivo quanto o intervalo entre arquivos seguem distribuições exponenciais. No tráfego entre 0 e 3, as médias dessas distribuições são, respectivamente, 30 Kbytes e 1 segundo e entre 2 e 1, 40 Kbytes e 1 segundo, respectivamente. Esse modelo de tráfego pode representar, por exemplo, a transfêrencia de fotografias de uma câmera digital para um *laptop* ou pequenos trabalhos de impressão.

A Tabela 2 contém os valores dos parâmetros do AISA que foram mantidos fixos ao longo das simulações do Cenário 1. Os parâmetros foram escolhidos visando a rápida adaptação às mudanças no tráfego, maximizando a vazão média agregada. A permanência mínima da ponte em uma *piconet* (**min_dur**) deve ser a menor possível, liberando

o máximo de *slots* para as *piconets* com mais tráfego. Quanto à taxa de acréscimo de *slots* (**inc_rate**), ela é limitada pela quantidade de *slots* livres, restringindo o efeito da diferença entre os valores de **inc_rate**. Finalmente, **skip_pico** e **early_exit** visam, exclusivamente, economizar energia e foram configurados de forma a desativar essa funcionalidade.

| min_dur | max_dur | inc_rate | dec_rate | skip_pico | early_exit |
|-----------------|----------------------|----------|----------|-----------|----------------------|
| 20 <i>slots</i> | turn_sz <i>slots</i> | 20% | 20% | 0 (bool) | turn_sz <i>slots</i> |

Tabela 1: Parâmetros do AISA adotados no Cenário 1.

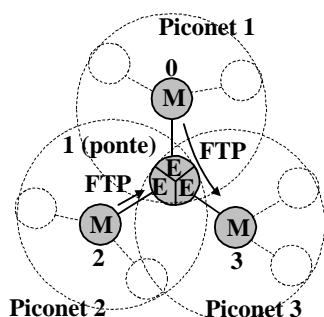


Figura 3: Cenário 1.

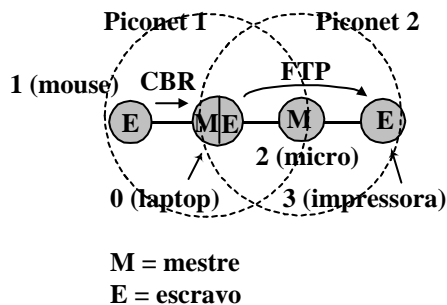


Figura 4: Cenário 2.

5.2. Resultados do Cenário 1

A simulação deste cenário mostra o resultado comparativo entre o AISA e o *round robin* (RR). O AISA foi executado com duas configurações distintas. Na configuração AISA 1, os limites a partir dos quais pode haver acréscimo ou liberação de *slots* (respectivamente, **inc_bound** e **dec_bound**) foram definidos como 80% e 60%, respectivamente. Na configuração AISA 2, esses parâmetros assumiram, respectivamente, os valores 90% e 50%. Foram executadas dez rodadas de simulação, cada uma com a duração de 120 segundos (simulações finitas independentes [18]) e os resultados das médias são expressos com intervalo de confiança de 95%.

Como a ponte perde até dois *slots* a cada troca de *piconet*, quanto menor a duração do turno, menor será a permanência da ponte em cada *piconet* e maior será a taxa de *slots* perdidos nessa troca. Nesta simulação, a duração do turno foi configurada para 240 *slots*², o que equivale a 80 *slots/piconet* no início da simulação.

A Figura 5 (a) mostra a vazão agregada das duas fontes FTP, medida a cada segundo, para as configurações AISA 1 e *round robin* (RR). A grande oscilação da vazão ao longo do tempo é provocada pela aleatoriedade do tráfego FTP/TCP não-persistente. O algoritmo AISA permite que o tráfego alcance vários picos de vazão acima de 500 kbps, representados, na figura, pelas barras verticais. Estes picos são causados pela redistribuição dinâmica de *slots* propiciada pelo AISA. Já no caso do algoritmo RR, apenas um ponto encontra-se acima de 400 kbps.

A Figura 5 (b) mostra a vazão média agregada dos fluxos FTP, computada ao longo de toda a simulação. A configuração AISA 1 apresentou a vazão média próxima de 360 kbps, contra menos de 320 kbps do RR, propiciando um ganho em torno de 15% no desempenho. Essa diferença poderia ter sido ainda maior se o tamanho médio dos arquivos fosse maior, pois os períodos de pico de transmissão seriam mais longos.

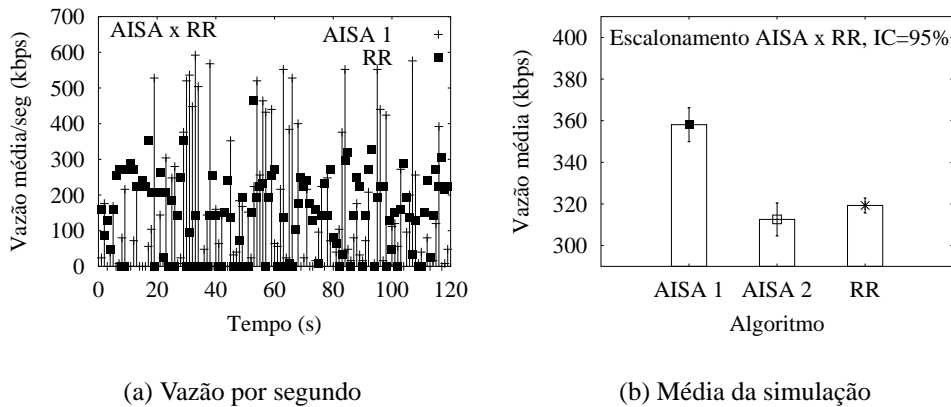


Figura 5: Resultados comparativos entre os mecanismos AISA e RR.

O desempenho da configuração AISA 2 foi bem inferior ao da AISA 1. O motivo é que, conforme o limite **inc_bound** aproxima-se dos 100%, a utilização média (**avg_util**) demora mais a ultrapassá-lo, retardando o ganho de *slots* para uma *piconet*. Analogamente, quanto maior for o parâmetro *dec_bound*, mais rapidamente uma *piconet* pode liberar *slots*. A configuração AISA 2 torna a duração dos encontros da ponte com as *piconets* mais estável, diminuindo a adaptatividade do algoritmo.

Com o auxílio dos resultados, fica claro que alguns parâmetros influem diretamente na vazão agregada da *scatternet*. As seguintes sugestões ajudam a aumentar a vazão. A duração do turno (**turn_sz**) deve ser grande, garantindo entre 80 e 100 *slots/piconet* no início da simulação. A duração mínima da ponte em uma *piconet* (**min_dur**) deve ser reduzida, ao passo que não se deve restringir a duração máxima (**max_dur**). Valores próximos de 100% para o limite **inc_bound** dificultam o ganho de *slots*. Analogamente, valores próximos de zero para o limite **dec_bound** dificultam a liberação de *slots*. Com os valores de 100% e zero para **inc_bound** e **dec_bound**, o funcionamento do AISA torna-se similar ao *round robin*.

5.3. Cenário 2 - Métrica: Retardo

Este cenário é composto por duas *piconets*. Em certo momento, o *laptop*, mestre da *piconet* 1, conecta-se ao microcomputador, para imprimir um arquivo. Na *piconet* 2, o *laptop* comporta-se como escravo, caracterizando uma ponte mestre/escravo. A Figura 4 ilustra o cenário.

O *mouse* é um dispositivo que exige interatividade, encaixando-se na categoria dos dispositivos de tráfego com requisito de retardo limitado. O *mouse* gera um pacote de 16 bytes a cada 65 ms (valor próximo ao adotado por Racz *et al.* [10]). O tráfego de fundo é composto pelo fluxo de impressão. A impressão foi modelada como um tráfego FTP.

A Tabela 2 mostra os valores dos parâmetros do AISA que foram mantidos fixos ao longo de todas as simulações do Cenário 2. Como a taxa de geração de pacotes do tráfego do *mouse* é baixa, os parâmetros que permitem ganhar ou liberar *slots* mais rapidamente (**inc_bound**, **dec_bound**, **inc_rate** e **dec_rate**) têm pouca influência no seu desempenho, mas podem, sem prejuízo ao *mouse*, melhorar o desempenho do tráfego de fundo. Por isso,

²No trabalho [12], mostrou-se que, a partir de 240 *slots*, o ganho de vazão provocado pela quantidade de *slots* é muito pequeno.

foram adotados os mesmos valores testados na configuração AISA 1 do Cenário 1. Novamente, **skip_pico** e **early_exit** foram configurados de modo a desativar as funcionalidades de consumo de energia.

| inc_bound | dec_bound | inc_rate | dec_rate | max_dur | skip_pico | early_exit |
|-----------|-----------|----------|----------|---------------|-----------|---------------|
| 80% | 60% | 20% | 20% | turn_sz slots | 0 (bool) | turn_sz slots |

Tabela 2: Parâmetros do AISA adotados no Cenário 2.

5.4. Resultados do Cenário 2

O tráfego do *mouse* para o *laptop* foi testado na presença do tráfego de impressão, originado na ponte e destinado à impressora. As simulações tiveram a duração de 120 segundos e foram comparadas três configurações diferentes de algoritmo de escalonamento *interpiconet* na ponte: AISA 1, com duração mínima da ponte em uma *piconet* (**min_dur**) igual a 20 slots, AISA 2, com **min_dur** igual a 50 slots e *round robin* (RR). Para AISA 1 e RR, variou-se a duração do turno (**turn_sz**) de 60 a 200 slots. Para AISA 2, simulações foram iniciadas em 120 slots.

A Figura 6 (a) mostra o 95-ésimo percentil do retardo para as três configurações acima, variando-se a duração do turno (**turn_sz**). Como o tráfego do *mouse* foi modelado gerando um pacote a cada 65 ms, considerou-se esse o valor máximo aceitável para o retardo dos pacotes.

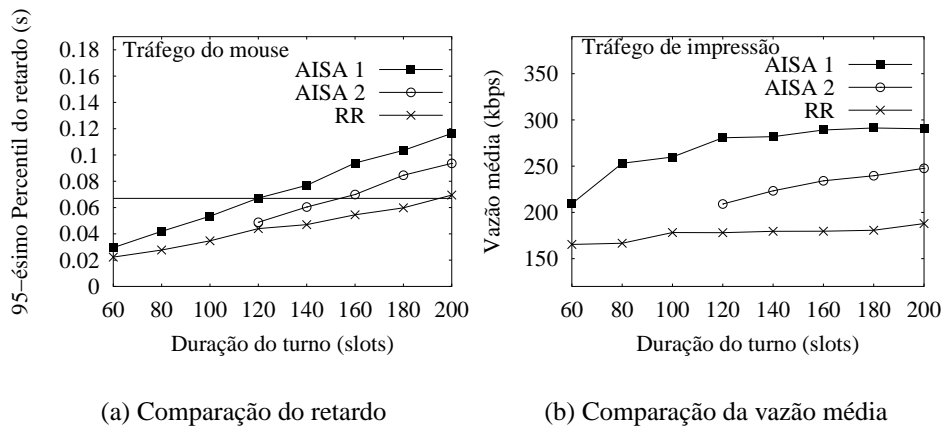


Figura 6: Resultados comparativos entre AISA e RR, com duas piconets.

A configuração AISA 1 ultrapassa o limite de retardo estipulado próximo dos 120 slots. Já com o aumento da duração mínima (AISA 2), o limite de retardo só é ultrapassado próximo de 150 slots. Este fato ocorre porque aumentando-se a duração mínima (**min_dur**), diminui-se automaticamente a duração máxima (**max_dur**) da outra *piconet*.

Como a vazão do *mouse* é baixa, a ponte dedica o tempo mínimo (**min_dur**) à sua *piconet*. Quando **min_dur** é 20 e a duração do turno (**turn_sz**) é 120, 100 slots (62,5 ms) destinam-se à *piconet* com o tráfego de impressão. Por isso, esse valor está próximo do limite superior do retardo aceitável para o *mouse*.

O algoritmo *round robin* (RR) fornece retardos menores que o AISA para uma mesma duração de turno, mas essa redução vem acompanhada de uma diminuição na vazão agregada do tráfego de fundo. Esse efeito é representado na Figura 6 (b). A vazão média do tráfego usando o RR fica abaixo de 180 kbps para todos os valores de turno testados.

Por sua vez, o AISA 1 obtém uma vazão média, com o turno de 120 *slots*, em torno de 280 kbps para o tráfego de impressão, mantendo o retardo dentro do limite estabelecido.

Convém, também, analisar o comportamento do retardo dos pacotes do *mouse*, quando a ponte participa de mais de duas *piconets*. Variou-se o número de *piconets* conectadas à ponte de três até sete. O teste foi executado com turno de 140 *slots*, permitindo que, mesmo na configuração com sete *piconets*, a duração mínima (**min_dur**) de 20 *slots* do AISA 1 seja respeitada para todas as *piconets*. Além disso, com sete *piconets*, o algoritmo AISA deve se comportar como o RR, pois todas as *piconets* permanecem com a mesma parcela do turno.

Cada *piconet* adicionada ao cenário mantém um fluxo de transferência de arquivos (FTP) sobre TCP não-persistente, análogo ao explicado no Cenário 1. Os resultados do 95-ésimo percentil do retardo dos pacotes do *mouse* e da vazão agregada de todo o tráfego são mostrados na Figura 7.

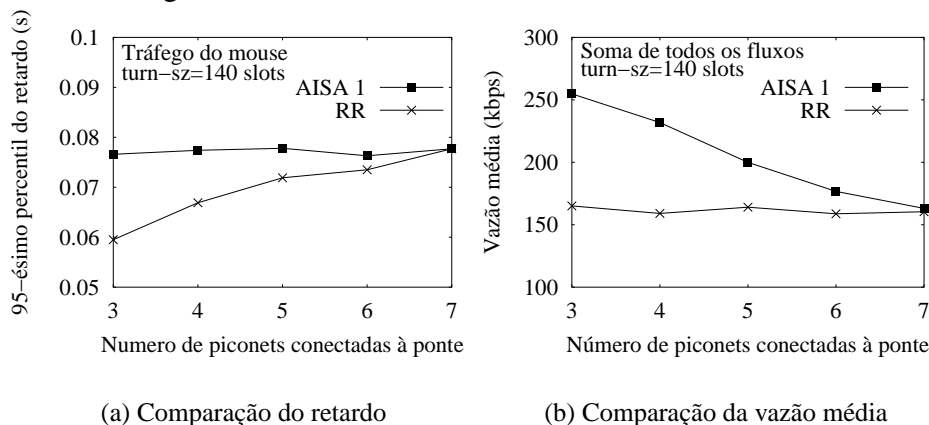


Figura 7: Resultados de AISA e RR, variando-se o número de piconets.

O perfil do retardo para o AISA 1 se mantém estável, independente do número de *piconets*, pois a *piconet* associada ao *mouse* geralmente ocupa a duração mínima permitida. A ponte divide o tempo restante entre as outras *piconets*. Já o *round robin* (RR) reparte o turno igualmente entre todas as redes. Por isso, com o RR, o retardo cresce com o aumento do número de *piconets* associadas à ponte. Apesar do AISA manter os valores de retardo maiores do que o RR, o AISA consegue, através da parametrização, obter retardos bem definidos em todos os casos.

Em termos de vazão, o *round robin* (RR) mantém a curva estável. Já a vazão agregada obtida com o AISA 1 diminui com o aumento do número de *piconets*. Conforme são acrescentadas novas redes, diminui-se a permanência na *piconet* da impressora. Além disso, há a perda de *slots* com a troca de *piconets* realizada pela ponte. Ainda assim, na configuração com seis *piconets*, a vazão agregada com o AISA é cerca de 10% maior do que com o algoritmo RR.

Dos resultados expostos, conclui-se que o AISA pode ser configurado para manter o retardo limitado para determinados fluxos, aumentando, concomitantemente, a vazão dos fluxos restantes. Sugere-se a configuração da duração do turno (**turn_sz**) de forma que a ponte não esteja ausente de uma *piconet* por mais tempo do que o estipulado para o retardo. Por exemplo, para o retardo máximo de 65 ms (equivalente a 104 *slots*), pode-se usar uma configuração com turno (**turn_sz**) e duração mínima (**min_dur**) iguais a 120 e 20 *slots*, respectivamente, ou 140 e 40 *slots*, respectivamente.

5.5. Cenário 3 - Métrica: Consumo de Energia

O objetivo do Cenário 3 é mostrar que, configurando-se corretamente os parâmetros do AISA, pode-se reduzir o consumo de energia nas pontes e, conseqüentemente, na *scatternet* como um todo. Esse tipo de avaliação pode servir para uma possível aplicação na área de redes de sensores.

A topologia do cenário é uma *scatternet*, composta por nove *piconets*, conforme ilustra a Figura 8. Somente as *piconets* que se encontram nas extremidades geram dados. As fontes são divididos em três categorias e cada *piconet* possui uma fonte de cada categoria. A estação central na *scatternet* é o ponto de acesso ao qual se destina todo o tráfego de dados. Todas as fontes geram pacotes à taxa constante de 3 kbps. As Categorias 1, 2 e 3 transmitem pacotes de 300, 100 e 20 bytes, respectivamente, resultando em intervalos entre pacotes diferentes.

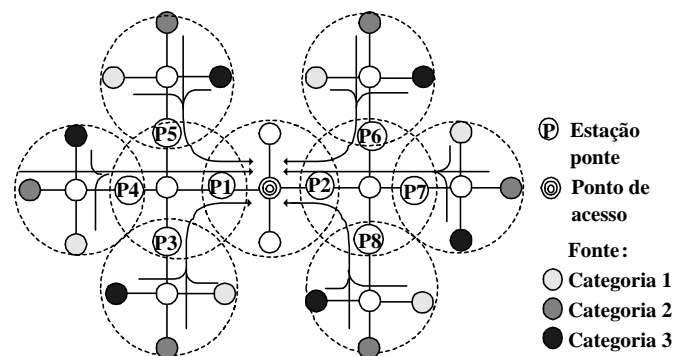


Figura 8: Topologia do Cenário 3.

Neste cenário, apenas os parâmetros contidos na Tabela 3 foram mantidos fixos em todas as simulações. A duração mínima (**min_dur**) deve ser pequena, permitindo que a ponte permaneça pouco tempo em *piconets* com poucos dados a transmitir. Os parâmetros **skip_pico** e **early_exit** possibilitam a economia de energia.

| min_dur | dec_rate | skip_pico | early_exit |
|----------|----------|-----------|------------|
| 20 slots | 20% | 1 (bool) | 15 slots |

Tabela 3: Parâmetros do AISA adotados no Cenário 3.

No Bluetooth, as correntes consumidas na transmissão e recepção de um pacote são da mesma ordem de grandeza ([19, 20]). Definiu-se, portanto, uma unidade de energia (1 u.e.) como sendo a potência dissipada na transmissão (ou recepção) do pacote de um *slot*. Todos os demais valores (por exemplo, potência dissipada transmissão de 3 slots, na recepção de 5 slots, etc.) foram calculados relativamente a essa unidade ³.

5.6. Resultados do Cenário 3

O cenário foi testado com duas configurações do AISA e com o *round robin* (RR). O algoritmo RR serve como parâmetro de comparação em termos de retardo dos pacotes. A métrica de retardo não é o foco principal deste cenário, mas a economia de energia não deve trazer como conseqüência um retardo excessivo no tráfego. Os parâmetros das duas configurações do AISA, denominadas AISA 1 e AISA 2, são expostos na Tabela 4.

³Maiores detalhes sobre o cálculo dos níveis de energia podem ser obtidos no trabalho [12].

Em uma primeira etapa, as simulações tiveram por objetivo medir o consumo de energia das pontes nas configurações do AISA e do RR. Foram executadas dez rodadas de simulações, com a duração de 120 segundos cada. Em cada uma das dez rodadas, variou-se a duração do turno de 60 até 140 *slots*. Para cada ponte, foi tirada a média do consumo de energia das dez rodadas, com intervalo de confiança de 95%.

| | inc_bound | dec_bound | inc_rate | max_dur |
|--------|-----------|-----------|----------|------------------------|
| AISA 1 | 80% | 60% | 20% | turn_sz slots |
| AISA 2 | 90% | 70% | 10% | (60% de turn_sz) slots |

Tabela 4: Configurações AISA 1 e AISA 2.

Existem oito pontes neste cenário. As pontes podem ser divididas em dois grupos, em relação à sua posição na topologia: o Grupo 1, contendo as pontes 1 e 2 e o Grupo 2, contendo as demais pontes. Devido às características das fontes e sua posição nas *piconets*, os resultados apresentados pelo Grupo 1 são aproximadamente iguais entre si. Da mesma forma, os resultados do Grupo 2 são iguais entre si. Por isso, os resultados são apresentados em termos da média obtida para cada grupo. A Figura 9 apresenta a energia consumida pelos dois grupos na simulação ⁴.

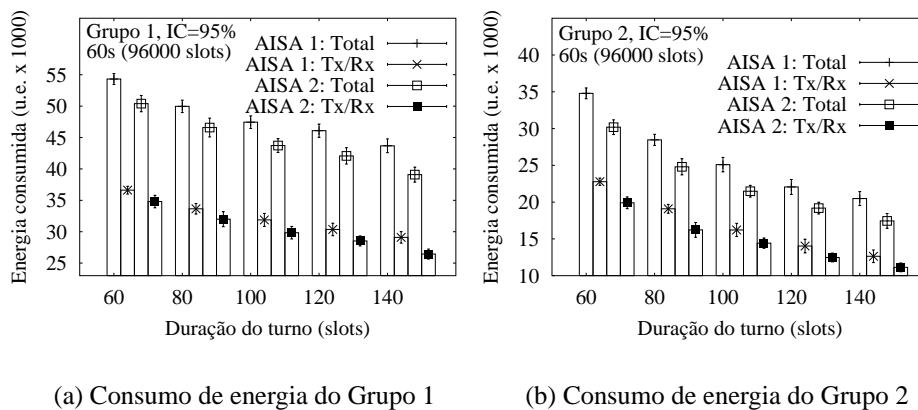


Figura 9: Consumo de energia das pontes dos Grupos 1 e 2, para AISA 1 e AISA 2.

Como esperado, os resultados mostram que, em todas as configurações de turno, as pontes do Grupo 1 consomem mais energia do que as pontes do Grupo 2, pois todo o tráfego destinado ao ponto de acesso passa pelo Grupo 1. Certamente, a energia do Grupo 1 acabará antes da energia do Grupo 2. Esse efeito é analisado adiante nesta seção.

Em todos os casos, o aumento do turno provoca a redução do consumo de energia nas pontes. Essa redução ocorre porque, como as pontes têm poucos dados a transmitir, elas permanecem, normalmente, o tempo mínimo (**min_dur**) em cada *piconet*. O aumento da duração do turno possibilita que a ponte, após escalonar todas as suas *piconets*, fique um período maior em modo HOLD, até o fim do turno. Essa situação se reflete na diminuição do consumo de energia.

Comparando os resultados das configurações AISA 1 e AISA 2, nota-se que AISA 2 consome menos energia que AISA 1, tanto para as pontes do Grupo 1 quanto para as

⁴Os resultados do RR não foram colocados na figura por motivo de legibilidade. O RR não economiza energia, por não trabalhar com o modo HOLD. Por exemplo, para o Grupo 1, com turno de 60 *slots*, o consumo médio de energia com o RR foi 70461 ± 1023 u.e., ou seja, cerca de 30% maior do que usando o AISA 1.

do Grupo 2 (Figura 9). Na configuração AISA 2, a ponte tem mais facilidade para liberar *slots* (pois $\text{dec_bound}_{\text{AISA2}} > \text{dec_bound}_{\text{AISA1}}$ na Tabela 4) e mais dificuldade para aumentar sua duração ($\text{inc_bound}_{\text{AISA2}} > \text{inc_bound}_{\text{AISA1}}$) do que na AISA 1. Quando a utilização média do enlace (**avg_util**) ultrapassa o limite que permite o acréscimo de *slots* (**inc_bound**), a taxa de aumento com AISA 2 é menor do que com AISA 1 ($\text{inc_rate}_{\text{AISA1}} < \text{inc_rate}_{\text{AISA2}}$). Finalmente, em AISA 1, a ponte pode ocupar todos os *slots* livres no encontro com uma *piconet*, enquanto em AISA 2, a duração de um encontro é limitada por **max_dur**.

A maior economia de energia propiciada pela configuração AISA 2 traz como contrapartida o aumento do retardo dos pacotes. No estudo do retardo, incluiu-se o algoritmo *round robin* (RR) como parâmetro de comparação. Para cada rodada, o 95-ésimo percentil do retardo dos pacotes foi avaliado. Ao final das dez rodadas, foi calculada a média do 95-ésimo percentil do retardo para cada fonte, com intervalo de confiança de 95%. A Figura 10 apresenta o resultado de retardo médio da categoria 1. As outras duas categorias apresentaram curvas semelhantes.

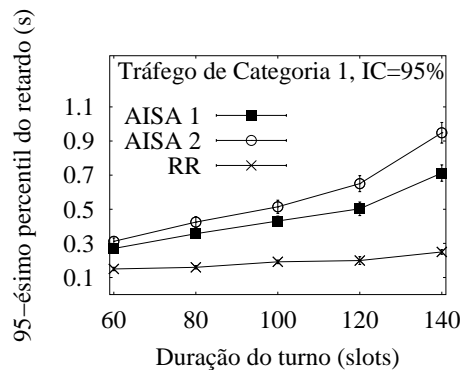


Figura 10: Retardo dos pacotes da Categoria 1, usando AISA 1, AISA 2 e RR.

O algoritmo RR propicia os menores retardos, porém, sem economia de energia. A configuração AISA 2 apresenta retardos ligeiramente maiores que AISA 1 até o valor de turno de 100 *slots*. A partir daí, a diferença entre as duas configurações aumenta em proporções maiores. Portanto, existe um compromisso entre a economia de energia desejada e o retardo aceitável para o tráfego.

Devido ao posicionamento das pontes na topologia, as pontes 1 e 2 consomem mais energia do que as pertencentes ao Grupo 2. Supondo que todas as pontes comecem a funcionar com a mesma carga de bateria, as pontes 1 e 2 deixarão de funcionar antes das restantes, não havendo mais rota para o ponto de acesso. Portanto, em alguns casos, devem-se buscar configurações diferenciadas para os grupos de pontes, na tentativa de que todas esgotem sua energia ao mesmo tempo. A nova configuração não deve aumentar excessivamente o retardo dos pacotes.

Buscando equalizar o consumo de energia, o Cenário 3 foi testado com uma parametrização diferente para cada grupo de pontes. A Tabela 5 resume as duas configurações. Note que o Grupo 1 foi configurado de forma similar à parametrização AISA 2 anterior, buscando minimizar o consumo de energia desse grupo. Entretanto, o retardo dos pacotes aumenta com essa configuração. Para contrabalançar o aumento do retardo, mesmo causando aumento no consumo de energia, o Grupo 2 foi configurado de forma similar à parametrização AISA 1 anterior.

Foram calculados o consumo médio de energia para os dois grupos de pontes e a média do 95-ésimo percentil do retardo dos pacotes, ambos com intervalo de confiança de 95%. A Tabela 6 contém as informações relativas ao consumo de energia nesta simulação, enquanto a Tabela 7 mostra o resultado de retardo para as três categorias de sensores.

| | turn_sz | inc_bound | dec_bound | inc_rate | max_dur |
|---------|-----------|-----------|-----------|----------|---------------|
| Grupo 1 | 140 slots | 90% | 70% | 10% | 80 slots |
| Grupo 2 | 60 slots | 80% | 60% | 20% | turn_sz slots |

Tabela 5: Parâmetros empregados nas pontes, visando equalizar o consumo.

| | Grupo 1 | Grupo 2 |
|---|-------------|-------------|
| Energia total consumida (u.e.) | 35134 ± 775 | 34436 ± 591 |
| Energia consumida na transmissão e recepção de pacotes (u.e.) | 23136 ± 534 | 22952 ± 467 |

Tabela 6: Energia consumida, com configurações diferentes para Grupos 1 e 2.

| | Categoria 1 | Categoria 2 | Categoria 3 |
|----------------------|---------------|--------------|--------------|
| Média do retardo (s) | 0,627 ± 0,015 | 0,57 ± 0,032 | 0,58 ± 0,024 |

Tabela 7: Retardo dos pacotes, com configurações diferentes para Grupos 1 e 2.

A diferença entre a energia consumida pelas pontes dos Grupos 1 e 2 foi menor que 3%. O mais importante é que o consumo das pontes 1 e 2 foi cerca de 20% menor do que o valor obtido com AISA 2 para os mesmos resultados de retardo dos pacotes (AISA 2 com turno de 120 slots na Figura 9 (a)), mostrando que é possível obter uma configuração para as pontes que prolongue o tempo de funcionamento de toda a rede.

Em resumo, para se maximizar a economia de energia, devem-se aumentar a duração do turno (**turn_sz**) e os limites **inc_bound** e **dec_bound**. Deve-se limitar a duração máxima da ponte em uma *piconet* (**max_dur**), e a taxa de acréscimo de slots (**inc_rate**) deve ser baixa. O efeito negativo dessa configuração é o aumento do retardo dos pacotes.

6. Conclusões e Trabalhos Futuros

Este trabalho apresentou um algoritmo de escalonamento *interpiconet*, denominado *Adaptive Interpiconet Scheduling Algorithm* (AISA). O AISA permite a adaptação da ponte às condições do tráfego e é parametrizável, possibilitando que uma determinada métrica de desempenho seja priorizada. As métricas testadas foram: a vazão agregada do tráfego, o retardo dos pacotes e o consumo de energia. O algoritmo AISA atua somente nas pontes, evitando-se a criação de sinalização adicional específica para o escalonamento. Dessa forma, sua implementação torna-se mais simples e com mais chance de ser incorporada à especificação.

Para testar o algoritmo AISA, foram definidos três cenários, explorando as três métricas de desempenho citadas. As simulações dos três cenários foram realizadas com o BlueNetS, uma extensão ao simulador ns-2 que permite simular diversas funcionalidades das camadas física e de enlace Bluetooth. Após diversas rodadas de simulações, foi possível isolar os parâmetros do AISA de maior influência em cada métrica de desempenho. A configuração adequada destes parâmetros permitiu a obtenção de bons resultados em todos os cenários avaliados. Por bons resultados, entende-se a otimização do desempenho da métrica definida como prioritária em um cenário, com o mínimo de prejuízo para as demais métricas.

Como continuação deste trabalho, pretende-se testar o algoritmo AISA com outros modelos de tráfego, incluindo tráfego de vídeo e navegação Web. Estas categorias de tráfego multimídia introduzem outras métricas de desempenho como, por exemplo, variação do retardo e tempo de resposta de uma solicitação.

Referências

- [1] “Bluetooth Special Interest Group,” <http://www.bluetooth.com>. Consultado em Nov. 2002.
- [2] J. Haartsen, “The Bluetooth Radio System,” *IEEE Personal Communications*, vol. 7, pp. 28–36, Fev. 2000.
- [3] T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaire, “Distributed Topology Construction of Bluetooth Personal Area Networks,” *IEEE Infocom*, Abr. 2001.
- [4] C. Law, A. Mehta, and K.-Y. Siu, “Performance of a New Bluetooth Scatternet Formation Protocol,” *Proc. 2001 ACM MobiHoc*, Out. 2001.
- [5] P. Bhagwat and A. Segall, “A Routing Vector Method (RVM) for Routing in Bluetooth Scatternets,” *The 6th IEEE MOMUC*, Nov. 1999.
- [6] M. Sun, C. Chang, and T. Lai, “A Self-routing Topology for Bluetooth Scatternets,” *Proc. I-SPAN 2002*, Mai. 2002.
- [7] N. Johansson, F. Alriksson, and U. Jönsson, “Jump Mode - A Dynamic Window-based Scheduling Framework for Bluetooth Scatternets,” *Proc. 2001 ACM MobiHoc*, Out. 2001.
- [8] P. Johansson, R. Kapoor, M. Kazantzidis, and M. Gerla, “Rendezvous Scheduling for Bluetooth Scatternets,” *Proc. ICC 2002*, Abr. 2002.
- [9] G. Tan, “Self-organizing Bluetooth Scatternets,” *Master Thesis*, Massachusetts Institute of Technology, Jan. 2002.
- [10] A. Racz, G. Miklos, F. Kubinszky, and A. Valko, “A Pseudo Random Coordinated Scheduling Algorithm for Bluetooth Scatternets,” *Proc. 2001 ACM MobiHoc*, Out. 2001.
- [11] L. Har-Shai, R. Kofman, G. Zussman, and A. Segall, “Interpiconet Scheduling in Bluetooth Scatternets,” *Proc. OPNETWORK 2002 Conference*, Ago. 2002.
- [12] W. Priess, “Mecanismos de Escalonamento com Qualidade de Serviço em Redes com Tecnologia Bluetooth,” *Tese de Mestrado, NCE/UFRJ*, Jan. 2003.
- [13] P. Johansson, N. Johansson, U. Körner, J. Elgg, and G. Svernar, “Short Range Radio Based Ad Hoc Networking: Performance and Properties,” *Proc. ICC 1999*, Jun. 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*, Out. 2001.
- [15] E. L. Hahne, “Round Robin Scheduling for Fair Flow Control in Data Communication Networks,” Massachusetts Institute of Technology, *Phd Thesis*, Dez. 1986.
- [16] W. Priess, J. F. de Rezende, L. Pirmez, and L. F. R. C. Carmo, “Bluenets: Uma Ferramenta para Estudo do Desempenho do Bluetooth,” *IV WCSF*, Out. 2002.
- [17] “The Network Simulator (ns-2),” <http://www.isi.edu/nsnam/ns>, Consultado em Jun. 2002.
- [18] K. Pawlikowski, H. -D. J. Jeong, and J. S. R. Lee, “On Credibility of Simulation Studies of Telecommunication Networks” *IEEE Communications Magazine*, pp. 132–139, Jan. 2002.
- [19] Philips Semiconductors, “UAA3558 Bluetooth RF Transceiver”.
www.semiconductors.philips.com/technologies/bluetooth, Consultado em Set. 2002.
- [20] Ericsson Microeletronics, “PBA 313 01/3 Bluetooth Radio”.
www.ericsson.com/microe/products/bluetooth_solutions, Consultado em Set. 2002.