

JOÃO HENRIQUE KLEINSCHMIDT

**REDES BLUETOOTH: MODELAGEM,
DESEMPENHO E APLICAÇÕES**

Dissertação apresentada ao Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática Aplicada.

CURITIBA

2004

JOÃO HENRIQUE KLEINSCHMIDT

**REDES BLUETOOTH: MODELAGEM,
DESEMPENHO E APLICAÇÕES**

Dissertação apresentada ao Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática Aplicada.

Área de Concentração: *Sistemas Distribuídos*

Orientador: Prof. Dr. Marcelo Eduardo Pellenz

Kleinschmidt, João Henrique

Redes Bluetooth: Modelagem, Desempenho e Aplicações. Curitiba, 2004. 116p.

Dissertação(Mestrado) – Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática Aplicada.

1. Bluetooth 2. Comunicações sem fio 3. Redes de sensores 4. Escalonamento I. Pontifícia Universidade Católica do Paraná. Centro de Ciências Exatas e de Tecnologia. Programa de Pós-Graduação em Informática Aplicada II-t

Esta página deve ser reservada à ata de defesa e termo de aprovação que serão fornecidos pela secretaria após a defesa da dissertação e efetuadas as correções solicitadas.

Agradecimentos

Aos meus pais, Edino e Vilma, e minha irmã, Ana Cristina, pelo apoio e compreensão durante toda a minha vida.

Ao orientador e amigo, Prof. Marcelo Eduardo Pellenz, pela paciência e dedicação nas longas discussões sobre os rumos deste trabalho.

Aos amigos Christiane e Guilherme, e demais amigos que de alguma maneira colaboraram na realização desta dissertação.

Agradeço também a CAPES pelo apoio financeiro, e finalmente, aos colegas e professores do PPGIA que me acompanharam nestes dois últimos anos.

Sumário

Agradecimentos	v
Sumário	vii
Lista de Figuras	xi
Lista de Tabelas	xiii
Lista de Símbolos	xv
Lista de Abreviaturas	xvii
Resumo	xxi
Abstract	xxiii
Capítulo 1	
Introdução	1
1.1. Desafio.....	1
1.2. Motivação.....	4
1.3. Proposta.....	5
1.4. Contribuição.....	5
1.5. Organização.....	6
Capítulo 2	
Comunicações sem Fio	7
2.1. Redes Móveis sem Fio.....	7
2.2.1. Redes Infra-estruturadas.....	8
2.2.2. Redes <i>Ad Hoc</i>	8
2.2.3. Redes de Sensores.....	10
2.2. Propagação de Ondas de Rádio.....	11

2.3.1. Reflexão.....	12
2.3.2. Difração.....	12
2.3.3. Espalhamento.....	12
2.3.4. Desvanecimento.....	12
2.3. Tecnologia Bluetooth.....	13
2.3.1. Camada de Rádio.....	14
2.3.2. Camada de Banda Base.....	15
2.3.3. Protocolo Gerenciador de Enlace – LMP.....	21
2.3.4. Interface de Controle do <i>Host</i> – HCI.....	21
2.3.5. Protocolo de Adaptação e Controle de Enlace Lógico – L2CAP.....	22
2.3.6. Camadas Superiores.....	22
2.3.7. Perfis.....	23
2.4. Outras Tecnologias para Redes sem Fio.....	23
2.5. Conclusão.....	25

Capítulo 3

Modelagem de Enlaces Bluetooth em Canais com Desvanecimento Nakagami-<i>m</i>	27
3.1. Análise de Vazão.....	27
3.2. Desempenho em Canais com Desvanecimento Nakagami- <i>m</i>	31
3.3. Modelo de Perda de Percurso.....	36
3.4. Conclusão.....	40

Capítulo 4

Simulação de Redes Bluetooth	41
4.1. Ferramentas de Simulação Existentes.....	41
4.2. Extensões para <i>BlueHoc</i> e <i>Blueware</i>	43
4.3. Desenvolvimento de Novos Simuladores.....	47
4.3.1. Simulador de Escalonamento de <i>Piconets</i>	47
4.3.2. Simulador do Processo de Conexão.....	48
4.4. Conclusão.....	50

Capítulo 5

Escalonamento de <i>Piconets</i> Bluetooth	51
5.1. Trabalhos Relacionados sobre Escalonamento.....	52
5.2. Informação de Estado de Canal.....	54
5.3. Algoritmo de Escalonamento usando Estimaco de Estado de Canal.....	55
5.4. Resultados de Simulao.....	57
5.4.1. Tráfego de Poisson.....	57
5.4.2. Tráfego HTTP.....	61
5.5. Concluso.....	63

Capítulo 6

Redes de Sensores usando Bluetooth	65
6.1. Formao e Roteamento de <i>Scatternets</i>	65
6.2. Uma Estrutura de Redes Bluetooth para Aplicaes de Sensores.....	68
6.3. Resultados de Simulao.....	71
6.4. Estudo de Caso de uma Aplicao de Sensoriamento usando Bluetooth.....	73
6.5. Concluso.....	76

Capítulo 7

Concluso	79
------------------	----

Referncias Bibliogrficas	83
-----------------------------------	----

Lista de Figuras

Figura 2.1	Rede móvel infra-estruturada [LOU02].....	8
Figura 2.2	Rede <i>ad hoc</i>	10
Figura 2.3	Pilha de protocolos Bluetooth.....	14
Figura 2.4	<i>Piconet</i> Bluetooth (adaptado de [BHU02]).....	16
Figura 2.5	TDD (adaptado de [BLU01a]).....	16
Figura 2.6	<i>Scatternet</i> Bluetooth.....	17
Figura 2.7	Formato de um pacote.....	17
Figura 2.8	<i>Payload</i> de pacotes assíncronos.....	18
Figura 2.9	Transições de estado do processo de <i>inquiry</i>	20
Figura 2.10	Transições de estado do processo de <i>page</i>	20
Figura 3.1	Desempenho do pacote DM1.....	32
Figura 3.2	Desempenho do pacote DM3.....	33
Figura 3.3	Desempenho do pacote DM5.....	33
Figura 3.4	Desempenho do pacote DH1.....	33
Figura 3.5	Desempenho do pacote DH3.....	34
Figura 3.6	Desempenho do pacote DH5.....	34
Figura 3.7	Vazão $\times PER_f$ para $m = 0,5$	34
Figura 3.8	Vazão $\times PER_r$ para $m = 0,5$	35
Figura 3.9	Vazão $\times PER_f$ para $m = 1$	35
Figura 3.10	Vazão $\times PER_r$ para $m = 1$	35
Figura 3.11	Vazão $\times PER_f$ para $m = 1,5$	36
Figura 3.12	Vazão $\times PER_r$ para $m = 1,5$	36
Figura 3.13	SNR \times distância.....	38
Figura 3.14	PER \times distância, $m = 0,5$, $\eta = 3$	38
Figura 3.15	PER \times distância, $m = 0,5$, $\eta = 4$	39

Figura 3.16	PER _x distância, $m = 1,5$, $\eta = 3$	39
Figura 3.17	PER _x distância, $m = 1,5$, $\eta = 4$	39
Figura 4.1	Procedimento de entrega de pacote.....	44
Figura 4.2	Primeiro cenário de simulação com quatro dispositivos.....	45
Figura 4.3	Topologia da <i>scatternet</i> para o primeiro exemplo.....	45
Figura 4.4	Segundo cenário de simulação com dispositivos de diferentes classes.....	46
Figura 4.5	Topologia da <i>scatternet</i> para o segundo exemplo.....	48
Figura 4.6	Entidades do simulador de escalonamento.....	49
Figura 4.7	Diagrama de estados do Bluetooth [BLU01a].....	50
Figura 5.1	Algoritmo BCSS.....	57
Figura 5.2	Primeiro cenário de <i>piconet</i>	58
Figura 5.3	Atraso médio para diferentes valores de m usando <i>round robin</i>	58
Figura 5.4	Segundo cenário de <i>piconet</i>	59
Figura 5.5	Atraso médio para $d = 7m$	59
Figura 5.6	Atraso médio para $d = 10m$	60
Figura 5.7	Atraso médio para diferentes cenários de tráfego.....	60
Figura 5.8	<i>Piconet</i> com um <i>gateway</i> Bluetooth.....	62
Figura 5.9	Tempo médio por página com 4 escravos.....	62
Figura 5.10	Tempo médio por página com 7 escravos.....	62
Figura 6.1	Fluxogramas do processo de descoberta.....	69
Figura 6.2	Distribuição dos nós.....	72
Figura 6.3	Tempo médio de descoberta.....	72
Figura 6.4	Exemplo de rotas descobertas para o cenário 1 com dez nós.....	73
Figura 6.5	Exemplo de rotas descobertas para o cenário 3 com dez nós.....	73
Figura 6.6	Cenário de aplicação.....	74
Figura 6.7	Fluxograma da descoberta dos nós móveis.....	75

Lista de Tabelas

Tabela 2.1	Classes de potência.....	15
Tabela 2.2	Pacotes assíncronos.....	19
Tabela 4.1	Vazão – Primeiro cenário.....	46
Tabela 4.2	Vazão – Segundo cenário.....	46
Tabela 5.1	Condições de tráfego para enlaces mestre-escravo.....	60
Tabela 5.2	Modelo de tráfego HTTP.....	61

Lista de Símbolos

b	tamanho de um pacote em bits
B	largura de banda
c	velocidade da luz
d	distância
d_o	distância de referência
D	número de <i>time slots</i> ocupados
$f(\gamma)$	função de densidade de probabilidade
f	frequência da portadora
h	índice de modulação
K	número de bits de dados de um pacote
m	parâmetro de desvanecimento de Nakagami
η	expoente de perda de percurso
N	número total de vezes que um pacote é transmitido
N_0	densidade espectral de potência unilateral de ruído
$P[.]$	probabilidade de ocorrência do evento
$P(n)$	probabilidade de n chegadas num tempo t
$p_N[n]$	probabilidade de um pacote ser transmitido n vezes
$\bar{P}_r(\bar{\gamma})$	probabilidade média de retransmissão
$p(\gamma)$	probabilidade de erro de símbolo
$\bar{P}L$	perda de percurso
$\bar{P}L(d_o)$	perda de percurso de referência
P_r	potência recebida
P_t	potência transmitida
PER_f	taxa de erro de pacote no canal direto

PER_r	taxa de erro de pacote no canal reverso
R	taxa de transmissão
R_{med}	vazão média
r_n	símbolo demodulado recebido
T	limiar de sincronização
T_d	tempo médio de descoberta dos nós móveis
T_s	tempo médio de envio de dados para um sensor
α_n	desvanecimento aleatório experimentado pelo n -ésimo símbolo recebido
γ_f	relação sinal-ruído instantânea do canal direto
γ_r	relação sinal-ruído instantânea do canal reverso
$\bar{\gamma}$	relação sinal-ruído média recebida
λ	taxa média de chegada de pacotes por <i>time slot</i>
Ω	segundo momento das amostras do sinal recebido

Lista de Abreviaturas

ACK	<i>Acknowledge</i>
ACL	<i>Asynchronous Connectionless</i>
AISA	<i>Adaptive Interpiconet Scheduling Algorithm</i>
AM_ADDR	<i>Active Member Address</i>
AODV	<i>Ad Hoc On-Demand Distance Vector</i>
ARQ	<i>Automatic Retransmission Query</i>
AWGN	<i>Additive White Gaussian Noise</i>
BAN	<i>Body Area Network</i>
BCSS	<i>Bluetooth Channel State Scheduling</i>
BD_ADDR	<i>Bluetooth Device Address</i>
BIAS	<i>Bluetooth Interference Aware Scheduling</i>
BSFA	<i>Bluetooth Scatternet Formation Algorithm</i>
BTCP	<i>Bluetooth Topology Construction Protocol</i>
CAC	<i>Channel Access Code</i>
CBR	<i>Constant Bit Rate</i>
CRC	<i>Cyclic Redundancy Check</i>
DAC	<i>Device Access Code</i>
DMx	<i>Data Medium-rate</i>
DHx	<i>Data High-rate</i>
DSDV	<i>Destination-Sequenced Distance-Vector</i>
DSR	<i>Dynamic Source Routing</i>
ERR	<i>Exhaustive Round Robin</i>
ETSI	<i>European Telecommunications Standards Institute</i>
FEC	<i>Forward Error Correction</i>
FHSS	<i>Frequency Hopping Spread Spectrum</i>

FTP	<i>File Transfer Protocol</i>
GFSK	<i>Gaussian Frequency Shift Keying</i>
GSFA	<i>Group-Scatternet Formation Algorithm</i>
HCI	<i>Host Controller Interface</i>
HEC	<i>Header Error Check</i>
HOL-KFP	<i>Head-of-the-Line K-Fairness Policy</i>
HTTP	<i>Hyper Text Transfer Protocol</i>
IAC	<i>Inquiry Access Code</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IP	<i>Internet Protocol</i>
IrDA	<i>Infrared Data Association</i>
ISI	<i>Information Sciences Institute</i>
ISM	<i>Industrial-Scientific-Medical</i>
LAA	<i>Load Adaptive Algorithm</i>
LAN	<i>Local Area Network</i>
LAR	<i>Location-Aided Routing</i>
LCS	<i>Locally Coordinated Scheduling</i>
LMP	<i>Link Manager Protocol</i>
L2CAP	<i>Logical Link and Control Adaptation Protocol</i>
LRR	<i>Limited Round Robin</i>
LWRR	<i>Limited and Weighted Round Robin</i>
MAC	<i>Medium Access Control</i>
MIT	<i>Massachusetts Institute of Technology</i>
NAK	<i>Negative Acknowledge</i>
NS	<i>Network Simulator</i>
OBEX	<i>Object Exchange</i>
PA	<i>Ponto de acesso</i>
PCSS	<i>Pseudo-Random Coordinated Scheduling Algorithm</i>
PDA	<i>Personal Digital Assistant</i>
PER	<i>Packet Error Rate</i>
PPP	<i>Point-to-Point Protocol</i>
QoS	<i>Qualidade de Serviço</i>

RFCOMM	<i>Radio Frequency Communications</i>
RR	<i>Round Robin</i>
SCO	<i>Synchronous Connection Oriented</i>
SDP	<i>Service Discovery Protocol</i>
SIG	<i>Special Interest Group</i>
SNR	<i>Signal-to-Noise Ratio</i>
TCP	<i>Transmission Control Protocol</i>
TCS	<i>Telephony Control Specification</i>
TDD	<i>Time-Division Duplex</i>
TORA	<i>Temporally Ordered Routing Algorithm</i>
TSF	<i>Tree Scatternet Formation</i>
WAKFP	<i>Wireless Adapted KFP</i>
WAP	<i>Wireless Application Protocol</i>
Wi-Fi	<i>Wireless Fidelity</i>
WLAN	<i>Wireless Local Area Network</i>
WPAN	<i>Wireless Personal Area Network</i>
WRP	<i>Wireless Routing Protocol</i>
WWW	<i>World Wide Web</i>
ZRP	<i>Zone Routing Protocol</i>

Resumo

Bluetooth é uma tecnologia de comunicação sem fio por ondas de rádio, de baixo custo e baixa potência. Esta tecnologia foi criada para eliminar fios e cabos entre dispositivos móveis e fixos para pequenas distâncias, formando redes pessoais sem fio (WPANs). Uma rede de até oito dispositivos que compartilham um canal comum é chamada de *piconet*, que opera num sistema mestre-escravo. Duas ou mais *piconets* podem se comunicar entre si formando uma rede de maior abrangência, denominada *scatternet*. O objetivo do padrão Bluetooth não é competir com as tecnologias de redes sem fio (WLANs), mas sim ampliar seus cenários de uso. Nesta dissertação são abordados quatro aspectos relativos a tecnologia Bluetooth: a modelagem eficiente do canal de rádio, o cálculo de desempenho de enlaces, estratégia de escalonamento para *piconets* e a aplicação em redes de sensoriamento. É proposta a utilização da distribuição de Nakagami- m para modelar o desvanecimento do canal de rádio, pois esta distribuição permite uma melhor caracterização de canais reais. Através do parâmetro " m " pode-se modelar diferentes condições de desvanecimento. O desempenho de enlaces Bluetooth para canais com desvanecimento Nakagami é derivado analiticamente, utilizando os pacotes assíncronos. Baseado no modelo de Nakagami é proposto um algoritmo de escalonamento de *piconets* denominado BCSS, que usa o parâmetro de desvanecimento " m " como estimativa do estado de canal para melhorar o desempenho de redes Bluetooth. Os nós priorizam a transmissão de dados quando as condições do canal forem boas, evitando transmissões em condições ruins. O desempenho do algoritmo de escalonamento foi avaliado através de simulação. Esta estratégia demonstrou bons resultados para condições de tráfego altas. Também é realizado um estudo das *scatternets* e sua utilização em redes de sensores, enfocando algoritmos para formação da rede, esquema de roteamento e possíveis cenários de aplicação. Embora a tecnologia Bluetooth não tenha sido projetada especificamente para redes de sensores, a estrutura de rede proposta, com um nó de gerência e pontos de acesso, pode ser utilizada em diversas aplicações. Na estrutura da rede nenhuma *scatternet* é formada, pois os dispositivos só se conectam em *piconets* temporárias quando precisam enviar ou receber

dados, sem a necessidade de manter conexões ativas. Isto diminui a sobrecarga de informação na rede devido à gerência das conexões e elimina o uso de algoritmos de escalonamento.

Palavras-Chave: Bluetooth, redes sem fio, escalonamento, redes de sensores.

Abstract

Bluetooth is a wireless technology that uses radio waves, with low cost and low power. It was designed to eliminate wires and cables between fixed and mobile devices for short distances, forming wireless personal area networks (WPANs). A network with up to eight devices that share a common channel is called piconet, which operates in a master-slave scheme. Two or more piconets can communicate forming a larger network, called scatternet. The objective of the Bluetooth standard is not to compete with the technologies for wireless networks (WLANs), but extends their usage models. In this work four aspects of Bluetooth technology are discussed: the efficient modeling of the radio channel, the evaluation of links performance, strategies for piconet scheduling and the application in sensor networks. It is proposed the use of the Nakagami- m distribution to model the fading of the radio channel, because this distribution gives a better characterization of real channels. Through the m parameter different fading conditions can be modeled. The performance of Bluetooth links for Nakagami fading channels is derived analytically, using the asynchronous packets. Based in the Nakagami model is proposed a piconet scheduling algorithm denominated BCSS, which utilizes the fading parameter m as an estimation of the channel state to improve the performance of Bluetooth piconets. The nodes give priority to data transmissions when the channel conditions are good, avoiding transmissions in bad conditions. The performance of the scheduling algorithm was evaluated by computer simulations. This strategy demonstrated good results for high traffic conditions. It is also realized a scatternet study and its application in sensor networks, with algorithms for network formation, routing scheme and possible application scenarios. Although the Bluetooth technology was not designed specifically for sensor networks, the network structure proposed, with a manager node and access points, can be utilized in several applications. In this structure the scatternet is not formed, because the devices only connect in temporary piconets when they require to send or receive data, without the need of maintaining active connections. This fact reduces the overload information in the network due to connections management and eliminates the use of scheduling algorithms.

Keywords: Bluetooth, wireless networks, scheduling, sensor networks.

Capítulo 1

Introdução

1.1. Desafio

A convergência das comunicações e da computação vem acontecendo há vários anos. Em 1991 o pesquisador Mark Weiser introduziu o termo computação ubíqua [WEI91], onde vários dispositivos com capacidade de computação e comunicação estão inseridos no ambiente de forma imperceptível para as pessoas. Nesta concepção, um computador tem a capacidade de obter informações do ambiente no qual ele está inserido e utilizá-las para melhor atender as necessidades do dispositivo ou usuário.

Weiser afirma que as mais profundas tecnologias são aquelas que desaparecem. O principal exemplo é a escrita, que pode ser considerada a primeira tecnologia da informação. A técnica de capturar a linguagem falada e transformá-la numa representação simbólica liberou a informação dos limites da memória individual de cada pessoa. A escrita está presente não só em livros, revistas e jornais, mas também em outros objetos, como embalagens, eletrodomésticos e placas de rua. O uso desta “tecnologia da escrita” não requer nenhum tipo de atenção especial por parte das pessoas, a informação faz parte do ambiente e pode ser usada sempre que necessário. Já a tecnologia da informação baseada na computação, permanece num mundo próprio e distante do ambiente real, como observa Weiser:

“Ela é acessível apenas através de um jargão complexo que não tem nada a ver com as tarefas para as quais as pessoas usam os computadores. O estágio de evolução talvez seja análogo ao período em que os escribas tinham de saber tanto sobre a obtenção da tinta e o cozimento da argila como sobre a escrita”.

Nesta visão, o computador pessoal pode ser visto apenas como uma etapa de transição em direção ao verdadeiro potencial da tecnologia da informação: que esteja tão

imperceptivelmente incorporada às nossas vidas diárias, que propicie todos os seus benefícios sem que sequer notemos a sua presença. Quando um motorista olha para uma placa na estrada para ver qual direção seguir para chegar ao seu destino, por exemplo, ele adquire esta informação sem tomar consciência do ato de ler. A leitura foi usada sem pensar, e só assim o motorista pode lançar seu foco além da leitura para alcançar seus objetivos, no caso escolher o caminho que o levará a seu destino. A computação ubíqua traz uma nova abordagem, que leva em conta o mundo humano e permita que os próprios computadores “desapareçam”. Se os computadores desaparecerem e forem usados sem pensar, as pessoas poderão se concentrar além dele em direção as suas verdadeiras metas e desejos.

A comunicação dos sistemas computacionais com o homem deve ser simples e eficiente, envolvendo reconhecimento de voz, interpretação de gestos ou até mesmo por ondas cerebrais [WWR01]. Assim, recursos computacionais e de comunicação podem estar presentes nos mais variados tipos de objetos: telefones celulares, câmeras digitais, etiquetas de roupa, embalagens de alimentos, canetas, eletrodomésticos, relógios, sensores, etc. A comunicação entre os objetos deve ser feita sem nenhum tipo de cabo e deve permitir a formação de redes de forma instantânea, sem a intervenção do usuário. Dentre as principais tecnologias de comunicação, o Bluetooth [BLU03] mostra-se como a mais promissora para estes novos conceitos de redes e computação ubíqua. É uma tecnologia de comunicação sem fio por ondas de rádio, de baixo custo e baixa potência.

A idéia que resultou na tecnologia Bluetooth nasceu em 1994 quando se verificou a necessidade de desenvolver uma interface de rádio para conectar telefones celulares e seus acessórios. Mais tarde esta tecnologia se mostrou promissora para se tornar um padrão universal para substituição de cabos entre os mais diversos dispositivos. Um grupo de interesse especial (SIG) foi formado no começo de 1998 pela Ericsson, IBM, Intel, Nokia e Toshiba [BHA01]. Mais tarde, Lucent, Microsoft, Motorola e 3Com se juntaram a elas no Bluetooth SIG como empresas promotoras da tecnologia. Atualmente centenas de outras empresas do mundo todo também fazem parte como membros do Bluetooth SIG.

A primeira versão da especificação foi lançada em 1999 e foi recebida com grande interesse pelos mais diversos setores. A divulgação pela mídia de uma tecnologia revolucionária que permitiria a comunicação entre qualquer tipo de dispositivo trouxe grandes expectativas, logo seguidas de uma certa frustração. O Bluetooth começou com muitas promessas, mas seu uso ficou limitado a alguns poucos produtos comerciais. Na realidade

nem poderia ser diferente, pois a própria especificação do Bluetooth deixa várias questões em aberto, que devem ser ainda pesquisadas pela comunidade científica e pela própria indústria. Hoje, sua popularidade está crescendo e já estão disponíveis diversos equipamentos integrados à tecnologia de rádio Bluetooth, como telefones celulares, fones de ouvido, computadores portáteis, PDAs, impressoras, teclados, câmeras digitais, entre outros.

O nome Bluetooth veio de um rei viking chamado Harald Bluetooth, que governou a Dinamarca entre 940 e 981 [MIL01]. Durante seu reinado a Dinamarca e a Noruega foram unificadas. Foi também o responsável em levar o cristianismo para o povo viking, que antes eram devotos do deus nórdico Odin. Muitos historiadores acreditam que Harald fez isto apenas como uma jogada política para proteger seu reino de uma futura invasão alemã. De qualquer forma, este ato o ajudou a entrar para a história. Na cidade dinamarquesa de Jelling, o rei Harald ergueu uma enorme pedra com símbolos rúnicos que ainda permanece em sua posição original. Ela tem uma imagem de Cristo com a seguinte inscrição: “Rei Harald ergue este monumento em memória de seu pai Gorm e sua mãe Thyre, o mesmo Harald que conquistou toda a Dinamarca e Noruega e tornou os dinamarqueses cristãos”.

Como curiosidade, o nome Bluetooth (ou *Blåtand*, em dinamarquês) não tem nada a ver com a cor dos dentes (*blue*=azul; *tooth*=dente). Na verdade, a palavra *Blåtand* é aparentemente derivada das palavras “*blå*”, que quer dizer pele escura, e “*tan*”, significando grande homem. Nas terras nórdicas onde as pessoas eram loiras e de pele clara, a cor do rei Harald se destacava.

A habilidade do rei em unir os povos inspirou a tecnologia Bluetooth, que pretende fazer com que os mais diversos tipos de dispositivos se comuniquem entre si. Espera-se que o Bluetooth possa não só eliminar cabos, mas que represente um passo adiante para a computação ubíqua. Usando as palavras de Mark Weiser,

“Computadores ubíquos ajudarão a resolver o problema de sobrecarga de informação. Há mais informação disponível ao nosso alcance durante uma caminhada num bosque do que em qualquer sistema de computação, ainda que as pessoas achem que uma caminhada entre as árvores é relaxante e usar os computadores pode ser muito frustrante. Máquinas que se adaptem ao ambiente humano, ao invés de forçar os seres humanos entrarem o delas, farão o uso de computadores tão agradável como uma caminhada no bosque”.

1.2. Motivação

Uma rede em que os dispositivos se comunicam automaticamente, conforme seu interesse, sem a necessidade de uma infra-estrutura de rede ou alguma estação de rede fixa, é chamada de rede *ad hoc*. As redes *ad hoc* tornam possíveis muitas aplicações do paradigma da computação ubíqua, pelo menos no que diz respeito à comunicação. Também podem ser usadas como solução de emergência em locais onde a infra-estrutura de comunicações tenha sido atingida por desastres naturais.

Uma aplicação recente e que tem recebido grande atenção nos últimos anos são as redes de sensores sem fio, que são compostas de algumas dezenas até milhares de pequenos dispositivos sensores, de baixa potência, com a capacidade de monitorar um ambiente ou equipamento e comunicar-se com outros elementos da rede. Ligado aos conceitos de computação ubíqua e redes *ad hoc* estão as redes pessoais sem fio (WPANs – *Wireless Personal Area Networks*), que são redes de curto alcance e com dispositivos usualmente associados a uma pessoa. A tecnologia Bluetooth possibilita a formação destes tipos de redes, ainda que muita pesquisa precisa ser feita em questões como modelagem do canal, técnicas para melhora de desempenho, segurança, escalonamento, roteamento e topologias de redes.

A estrutura básica de redes Bluetooth é chamada de *piconet* (ou pico-rede), composta por até oito dispositivos. Um deles assume o papel de mestre da rede e todos os outros (os escravos) se comunicam somente com o mestre. Esse número de oito dispositivos é suficiente para agrupar um microcomputador com todos os seus periféricos ou os dispositivos Bluetooth carregados por uma pessoa. Para aumentar a quantidade de nós da rede e também o seu alcance, foi criada uma outra estrutura de rede, chamada de *scatternet* (ou redes de dispersão). Numa *scatternet*, várias *piconets* podem ser interligadas através de nós ponte pertencentes a duas ou mais *piconets*.

Diferente das redes com fio tradicionais, o canal sem fio de um enlace Bluetooth é mais difícil de analisar, pois as ondas de rádio sofrem desvanecimento e estão sujeitas a várias fontes de interferência. Logo, uma modelagem adequada do canal de rádio precisa ser feita para análise de desempenho em diferentes ambientes e topologias de rede.

Questões relacionadas ao escalonamento em *piconets* e *scatternets* não foram especificadas. O *intrapiconet* é o escalonamento dos escravos, realizado pelo mestre dentro de uma *piconet*. O escalonamento *interpiconet* é o modo como um nó pertencente a múltiplas *piconets* divide seu tempo entre elas. Vários algoritmos de escalonamento têm sido propostos,

mas o problema continua em aberto. As *scatternets* foram definidas, mas os protocolos necessários para sua formação e roteamento de pacotes também não foram especificados. Algumas topologias e esquemas de roteamento também têm sido propostos na literatura para possibilitar a formação e manutenção de redes *ad hoc* e de sensores. Estas questões em aberto têm proporcionado inúmeras oportunidades de pesquisa.

1.3. Proposta

Esta dissertação tem como objetivos analisar o comportamento de enlaces Bluetooth em diferentes condições de canal, em termos de vazão e taxa de erro de pacotes. Baseado nestes resultados, propor uma política de escalonamento que use o estado do canal para melhorar o desempenho de redes Bluetooth. Também é realizado o estudo das *scatternets* e sua utilização em redes de sensores, propondo algoritmos para formação da rede, esquema de roteamento e possíveis cenários de aplicação. Os resultados são obtidos através de modelagens analíticas e de simulação.

1.4. Contribuições

Parte das análises e resultados apresentados na dissertação foram publicados em [KLE03], [KLE04a] e [KLE04b]. As contribuições trazidas por este trabalho de pesquisa são:

- Modelagem e análise de desempenho de enlaces Bluetooth em canais com desvanecimento Nakagami- m ;
- Implementação de novas características nas ferramentas de simulação *BlueHoc* [IBM00] e *Blueware* [TAN02c] para incluir modelos de erro para o canal com desvanecimento Nakagami e para diferentes classes de potência dos dispositivos Bluetooth;
- Desenvolvimento em linguagem C++ de um simulador orientado a eventos para estudo de técnicas de escalonamento e processo de formação de redes Bluetooth;
- Proposta de um algoritmo de escalonamento de *piconets* usando como métrica a estimação do estado de canal;
- Proposta de uma estratégia de formação e roteamento de redes Bluetooth para aplicações de sensoriamento;
- Estudo de caso de uma aplicação de redes de sensores usando Bluetooth.

1.5. Organização

A dissertação está estruturada em sete capítulos, organizados da seguinte forma:

- Capítulo 1 – Introdução: Faz uma apresentação geral do contexto deste trabalho.
- Capítulo 2 – Comunicações sem Fio: Descreve os conceitos básicos necessários para o entendimento da dissertação. São abordadas as redes *ad hoc* e de sensores, propagação de ondas de rádio e a tecnologia Bluetooth.
- Capítulo 3 – Modelagem de Enlaces Bluetooth em Canais com Desvanecimento Nakagami-*m*: Apresenta uma modelagem analítica para cálculo da vazão em enlaces Bluetooth e um modelo de perda de percurso. São apresentados resultados para canais com desvanecimento Nakagami-*m*.
- Capítulo 4 – Simulação de Redes Bluetooth: Discute os principais simuladores Bluetooth existentes, a implementação de novas funcionalidades e o desenvolvimento de novos simuladores. São apresentados resultados de simulação de *scatternets* com desvanecimento Nakagami.
- Capítulo 5 – Escalonamento de *Piconets* Bluetooth: Apresenta os principais esquemas de escalonamento de *piconets* e *scatternets*. Propõe um novo algoritmo baseado na estimação da qualidade do canal, mostrando resultados de simulação para diferentes condições de tráfego.
- Capítulo 6 – Redes de Sensores usando Bluetooth: Descreve os principais algoritmo de formação e roteamento de *scatternets* e seu uso em redes de sensores. São propostos algoritmos para formação e roteamento em redes de sensores com a tecnologia Bluetooth.
- Capítulo 7 – Conclusão: São apresentadas as conclusões e propostas de trabalhos futuros.

Capítulo 2

Comunicações sem Fio

Este capítulo apresenta alguns conceitos importantes para o entendimento da dissertação. A Seção 2.1 apresenta as redes móveis sem fio e as classifica em 3 categorias: redes infra-estruturadas, *ad hoc* e de sensores. A comunicação nestas redes é feita através de técnicas de rádio-frequência e a seção seguinte apresenta os conceitos básicos de propagação de ondas de rádio. A Seção 2.3 descreve com mais detalhes a tecnologia Bluetooth e sua pilha de protocolos, que constituem a base deste trabalho. Outras tecnologias para redes sem fio são discutidas na Seção 2.4. Finalmente, a Seção 2.5 faz a conclusão do capítulo.

2.1. Redes Móveis sem Fio

Uma rede de computadores é formada por um conjunto de dispositivos capazes de trocar informações e compartilhar recursos, interligados por um sistema de comunicação. Nas redes com fio o meio de transmissão utilizado é constituído de cabos, como fibra ótica, cabos coaxiais ou de par trançado. Em redes sem fio, o meio físico de transmissão é o ar, usando ondas de rádio ou luz infravermelha. Para que estas redes sem fio sejam consideradas móveis, os dispositivos devem ser capazes de se deslocar durante a comunicação sem que esta seja interrompida. Com o surgimento e a popularização de equipamentos portáteis e de comunicação sem fio como telefones celulares, computadores portáteis e PDAs (*Personal Digital Assistants*) novas aplicações estão surgindo. As redes móveis sem fio podem ser divididas em redes infra-estruturadas e *ad hoc*. As redes de sensores podem ser vistas como um caso especial das redes *ad hoc*.

2.1.1. Redes Infra-estruturadas

Nas redes infra-estruturadas existe pelo menos um equipamento fixo, chamado de ponto de acesso ou estação base. Qualquer comunicação entre os nós móveis será intermediada pelo ponto de acesso, como mostra a Figura 2.1. Quando uma estação móvel precisa enviar alguma informação para outra estação móvel, os dados são transmitidos a um ponto de acesso que irá reenviar os dados para a estação de destino. Como os pontos de acesso têm alcance limitado, normalmente existem vários deles interligados. Os pontos de acesso podem estar conectados a outras redes, como a Internet ou uma rede local. O principal exemplo de redes móveis sem fio infra-estruturadas são as redes de telefonia celular. Uma estação rádio-base possui uma área de cobertura chamada célula. Quando um usuário está usando seu telefone celular, a informação é transmitida para a estação rádio-base de sua célula, que verifica se o destino se encontra nesta mesma célula, em outra célula ou na parte fixa da rede de telefonia. Este tipo de rede não é abordado neste trabalho.

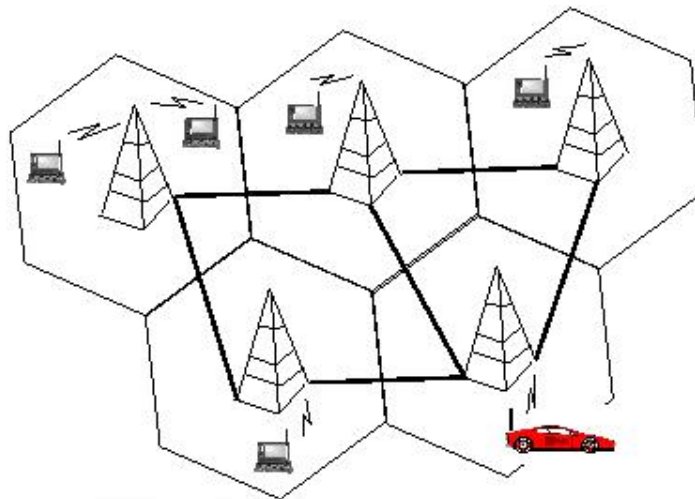


Figura 2.1: Rede móvel infra-estruturada [LOU03]

2.1.2. Redes *Ad Hoc*

Em uma rede *ad hoc* todos os nós móveis são capazes de se comunicar diretamente entre si, sem a necessidade de pontos de acesso para suporte à mobilidade, conforme Figura 2.2. Como os nós podem se mover arbitrariamente, a topologia da rede muda frequentemente e de forma imprevisível. Qualquer aplicação comum pode ser implementada num ambiente *ad hoc*, como transferência de arquivos ou correio eletrônico. Porém novas aplicações surgiram como específicas das redes *ad hoc* [COR02]:

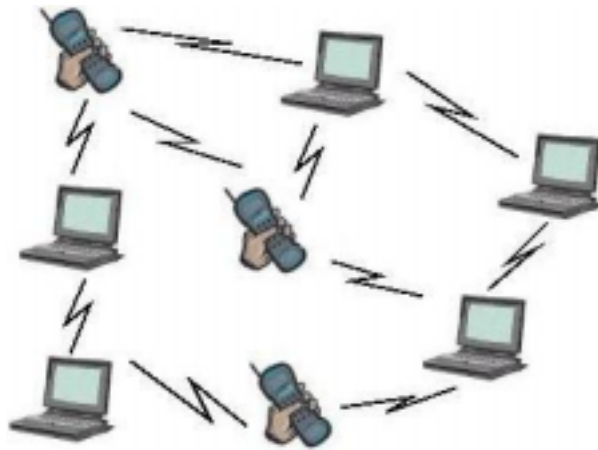
- trabalho cooperativo: em reuniões de trabalho ou conferências os participantes podem trocar informações no mesmo instante, estabelecendo redes temporárias com seus *notebooks* e PDAs;

- aplicações para gerenciamento de crises: o restabelecimento das comunicações em casos de desastres naturais como furacões, terremotos, incêndios ou enchentes pode ser feito de forma muito mais rápida com redes *ad hoc* do que com redes cabeadas, que podem levar dias ou até semanas para serem restabelecidas.

- redes pessoais sem fio (WPANs – *Wireless Personal Area Networks*): uma rede pessoal é uma rede de curto alcance que pode conectar os mais variados tipos de dispositivos eletrônicos, como computadores portáteis, câmeras digitais, telefones celulares, impressoras e outros. O principal padrão para WPANs é o Bluetooth, discutido na seção 2.3.

Os nós numa rede *ad hoc* podem ter diferentes alcances, capacidades de processamento e velocidades, mas devem ser capazes de fazer o roteamento de pacotes na rede. O roteamento em redes *ad hoc* é um desafio, pois como a conectividade dos nós muda constantemente, as rotas devem ser permanentemente adaptadas e reconfiguradas. Os protocolos de roteamento *ad hoc* podem ser classificados em pró-ativos (ou orientados a tabelas) e reativos (por demanda) [COR02]. Nos protocolos pró-ativos os nós devem manter uma tabela de roteamento para todos os possíveis destinos. Assim, quando um pacote precisa ser enviado a rota já é conhecida e pode ser imediatamente usada. Os principais protocolos que usam tabelas são o DSDV (*Destination-Sequence Distance-Vector*) e o WRP (*Wireless Routing Protocol*).

Nos protocolos reativos os nós descobrem as rotas para um destino apenas quando é necessário, sendo os principais o DSR (*Dynamic Source Routing*), AODV (*Ad Hoc On-Demand Distance Vector*), TORA (*Temporally Ordered Routing Algorithm*) e LAR (*Location-Aided Routing*). Protocolos pró-ativos têm a vantagem de apresentarem um atraso mínimo na descoberta da rota, pois esta é selecionada diretamente de uma tabela. Porém uma capacidade considerável da rede é usada para manter as informações de roteamento atualizadas. Os protocolos reativos consomem menos largura de banda, mas o atraso para descoberta de rota é bem maior. Para aproveitar as vantagens das duas abordagens, alguns protocolos híbridos têm sido propostos, como o ZRP (*Zone Routing Protocol*).

Figura 2.2: Rede *ad hoc*

2.1.3. Redes de Sensores

O avanço tecnológico recente nas áreas de sensores, circuitos integrados e comunicação sem fio permitiram a criação de redes de sensores sem fio [LOU03]. As redes de sensores são aplicações das redes *ad hoc* e devem ter um grande crescimento nos próximos anos, pois tem recebido muita atenção de empresas, dos militares e da comunidade científica. Este tipo de rede pode ser usada no monitoramento, rastreamento, coordenação e processamento em diferentes aplicações. Cada nó da rede pode ser equipado com uma variedade de sensores, tais como de temperatura, sísmico, pressão, acústico, de imagem, etc. Entre as áreas que as redes podem ser aplicadas pode-se citar o monitoramento das condições ambientais numa floresta, oceano, deserto e vulcões ou ainda em locais internos como prédios e residências. Num ambiente industrial, sensores podem ser colocados numa linha de montagem para fazer testes no processo de manufatura. Podem ainda ser usados para monitorar tráfego de veículos em rodovias, prover segurança em centros comerciais e em aplicações médicas, como monitoramento de condições físicas de um paciente. Isto sem considerar as aplicações militares, que foram os primeiros cenários de redes de sensores que deram um grande impulso no seu desenvolvimento, com um grande financiamento do departamento de defesa norte-americano.

Embora existam muitos protocolos e algoritmos para as redes *ad hoc*, eles não podem ser aplicados diretamente nas redes de sensores, pois estas apresentam algumas características únicas. Em geral, as redes de sensores diferem das redes *ad hoc* em alguns aspectos [AKY02]:

- o número de nós sensores pode ser muito grande (chegando até centenas ou milhares em casos de ampla abrangência geográfica);

- os sensores são mais suscetíveis à falhas;
- a topologia da rede pode mudar ainda mais rapidamente;
- a comunicação é feita geralmente através de *broadcast*, enquanto que na maioria das redes *ad hoc* a comunicação é ponto-a-ponto;
- os nós sensores são limitados em capacidade computacional, memória e principalmente, energia.

Algumas estratégias de roteamento, economia de energia, protocolos MAC (*Medium Access Control*) e de gerenciamento são apresentadas em [AKY02], [COR02] e [LOU03]. As redes de sensores sem fio apresentam desafios novos em relação aos sistemas distribuídos tradicionais e têm trazido inúmeras oportunidades de pesquisa.

2.2. Propagação de Ondas de Rádio

As redes apresentadas na Seção 2.1 utilizam ondas de rádio para transmissão de informação. Ao contrário dos canais com cabos que são estacionários e relativamente previsíveis, os canais de rádio são extremamente aleatórios, limitando o desempenho de sistemas de comunicação sem fio e dificultando sua análise [RAP99]. O transmissor e o receptor podem estar próximos, com linha de visada, ou estarem obstruídos por paredes, construções, montanhas e até mesmo pessoas ou outros objetos. A modelagem do canal de rádio é uma das partes mais difíceis no projeto de um sistema de rádio e deve ser feita de uma maneira estatística, baseada em medidas feitas especificamente para um sistema de comunicação, alocação de espectro e ambiente onde o sistema vai operar [RAP99].

Os modelos de propagação tentam prever a potência média do sinal recebido numa dada distância do transmissor, assim como a variabilidade da potência do sinal. Modelos de propagação que prevêm a potência média do sinal para uma separação transmissor-receptor qualquer são chamados de modelos de propagação de larga escala, pois caracterizam a intensidade do sinal em grandes distâncias transmissor-receptor. Os modelos de propagação que caracterizam as flutuações rápidas da potência do sinal recebido em distâncias muito pequenas (poucos comprimento de onda) ou durações muito curtas (poucos segundos) são chamados de pequena escala ou modelos de desvanecimento. Os mecanismos básicos de propagação de ondas eletromagnéticas são reflexão, difração e espalhamento. Os modelos de propagação de larga escala e de desvanecimento são baseados na física destes três mecanismos de propagação.

2.2.1. Reflexão

A reflexão ocorre quando uma onda eletromagnética atinge um objeto que tem dimensões muito grandes quando comparado ao comprimento de onda do sinal que se propaga. Reflexões ocorrem na superfície da Terra, em paredes e construções como casas e edifícios. Quando uma onda de rádio que se propaga num meio atinge um outro meio com diferentes propriedades elétricas, a onda é parcialmente refletida e parcialmente transmitida. A intensidade do campo elétrico das ondas refletidas e transmitidas podem ser relacionadas à onda incidente no meio de origem através do coeficiente de reflexão de Fresnel [RAP99]. Este coeficiente de reflexão depende das propriedades do material e também da polarização da onda, frequência e ângulo de incidência.

2.2.2. Difração

A difração permite que um sinal de rádio se propague pela superfície curvada da Terra e através de obstáculos. Embora a intensidade de campo recebida caia rapidamente quando um receptor se move mais para trás numa região obstruída, o campo de difração ainda existe e frequentemente tem força suficiente para produzir um sinal útil. O fenômeno da difração é explicado pelo princípio de Huygens, que diz que todos os pontos de uma onda podem ser considerados como fontes para a produção de novas formas de onda, que se combinam e produzem uma nova forma de onda na direção de propagação [RAP99].

2.2.3. Espalhamento

O espalhamento pode ser descrito como uma reflexão desorganizada de uma superfície rugosa, pequenos objetos ou outras irregularidades do canal. Ocorre quando o meio em que a onda se propaga tem objetos com dimensões pequenas se comparados ao comprimento de onda e o número de obstáculos é grande.

2.2.4. Desvanecimento

O desvanecimento é usado para descrever a rápida flutuação da amplitude do sinal de rádio num curto período de tempo ou pequenas distâncias. É causado pela interferência entre duas ou mais versões do sinal transmitido que chegam no receptor em tempos ligeiramente diferentes. Estas ondas são chamadas de ondas multipercurso e são combinadas na antena receptora resultando num sinal que pode variar em amplitude e fase, dependendo da

distribuição da intensidade e do tempo de propagação relativa das ondas e da largura de banda do sinal transmitido.

Devido ao movimento relativo entre o nó móvel de origem e destino, cada onda multipercurso experimenta um deslocamento na frequência. Esse deslocamento na frequência do sinal recebido devido ao movimento é chamado de deslocamento Doppler, sendo diretamente proporcional à velocidade e a direção do movimento do nó móvel em relação à direção de chegada da onda multipercurso recebida. Os objetos no canal de rádio também podem estar em movimento, gerando também o deslocamento Doppler. Se os objetos se movem numa taxa superior ao do nó móvel, o desvanecimento será dominado pelo movimento destes objetos. Caso contrário, o movimento dos objetos do canal podem ser ignorados e apenas o movimento do nó móvel pode ser considerado.

2.3. Tecnologia Bluetooth

Bluetooth [BLU01a] é uma tecnologia de comunicação sem fio por ondas de rádio, de baixo custo e baixa potência. Foi criada para eliminar fios e cabos entre dispositivos móveis e fixos para pequenas distâncias formando redes pessoais sem fio (WPANs). Um chip Bluetooth pode ser facilmente integrado em qualquer dispositivo eletrônico, possibilitando a criação de redes *ad hoc* entre vários tipos de dispositivos.

A especificação Bluetooth é dividida em duas partes: *Core* [BLU01a] e Perfis [BLU01b]. A especificação *Core* define todas as camadas da pilha de protocolos Bluetooth. Baseadas nessa pilha de protocolos, várias aplicações podem ser criadas. A especificação Perfis define 13 perfis de aplicação. Com o surgimento de novas aplicações, mais perfis devem ser adicionados a especificação. A versão descrita nesta seção e que foi considerada em toda a dissertação é a versão Bluetooth 1.1 [BLU01a], de fevereiro de 2001. Em novembro de 2003 foi lançada a versão 1.2 [BLU03]. Esta nova versão apresenta algumas novidades em relação à versão 1.1, mas é totalmente compatível. As principais são novos enlaces síncronos para transporte de áudio a taxas mais altas, mais opções de QoS (qualidade de serviço) e um esquema de salto de frequências adaptativo para reduzir interferência. Maiores mudanças são esperadas na versão 2.0, que ainda está em estudo e só deve ser lançada quando a tecnologia estiver mais madura. A pilha de protocolos Bluetooth é mostrada na Figura 2.3 e comentada nas próximas seções. As camadas de rádio e banda base são as

mais importantes para o entendimento da dissertação e por isso são explicadas com mais detalhes.

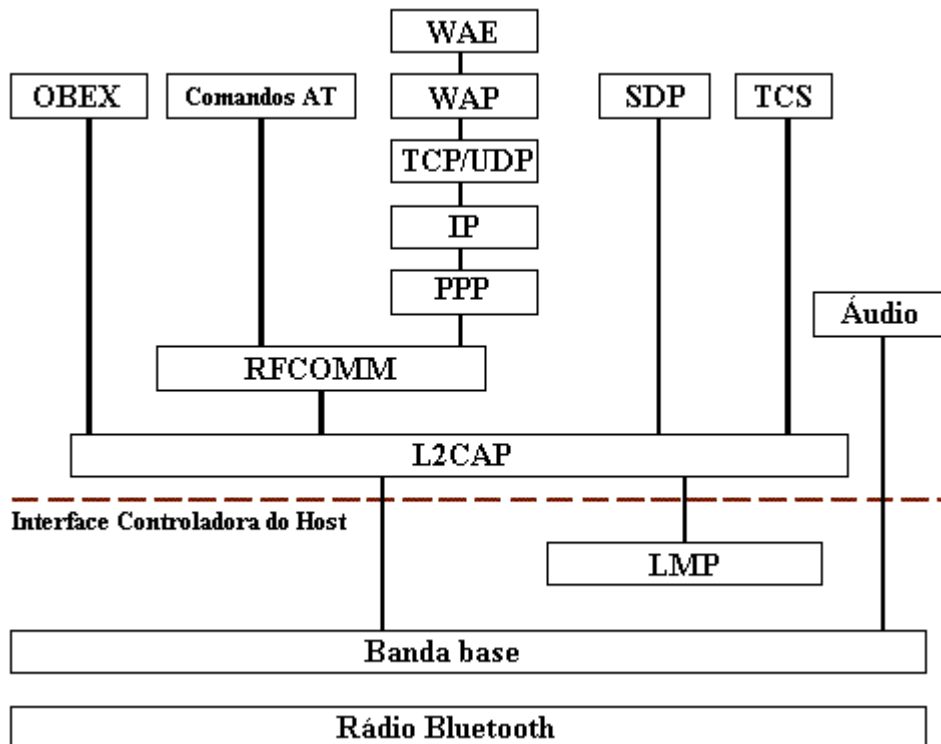


Figura 2.3: Pilha de protocolos Bluetooth

2.3.1. Camada de Rádio

Bluetooth opera na faixa de frequência de 2,4 GHz ISM (*Industrial, Scientific, Medical*). A banda ISM de 2,4 GHz está disponível para uso livre globalmente. Brasil, Europa e Estados Unidos alocam 83,5 MHz para esta banda, mas alguns países como Espanha, França e Japão alocam menos banda. Para suportar estas diferenças, 79 canais de 1 MHz são alocados para EUA, Brasil e Europa e 23 canais de 1 MHz para França, Espanha e Japão.

Bluetooth utiliza espalhamento espectral com saltos de frequência (*Frequency Hopping Spread Spectrum*- FHSS). São utilizados saltos de frequência pelos 79 ou 23 canais, a uma taxa de 1600 saltos por segundo para diminuir interferência com outras fontes que utilizem a mesma banda. A velocidade atingida chega a 1 Mbps, usando modulação GFSK (*Gaussian Frequency Shift Keying*), com o produto tempo-largura de banda igual a 0,5. O índice de modulação h deve estar entre 0,28 e 0,35. Um valor típico para o índice de modulação é de 0,32. Os dispositivos transmissores são divididos em 3 classes de potência,

conforme tabela 2.1. A classe 3 é a mais comum e tem um alcance aproximado de 10 metros. A classe 1 possui a maior potência de transmissão e atinge até 100 metros.

Tabela 2.1: Classes de potência

Classe de Potência	Potência Máxima	Potência Mínima
1	100 mW (20 dBm)	1 mW (0 dBm)
2	2,5 mW (4 dBm)	0,25 mW (-6 dBm)
3	1 mW (0 dBm)	-

2.3.2. Camada de Banda Base

A banda base é a camada física da pilha de protocolos. É responsável, entre outras funções, pela criação das *piconets* (ou pico-redes) e dos enlaces. Uma *piconet* é um conjunto de dispositivos Bluetooth compartilhando um canal comum. Uma *piconet* possui uma configuração estrela na qual o dispositivo no centro age como um mestre e os outros operam como escravos (Figura 2.4). Até sete escravos podem ser servidos simultaneamente por um mestre. Se um mestre precisa se comunicar com mais que sete dispositivos, ele pode instruir um ou mais escravos ativos a entrarem num modo de economia de energia, como o modo *park*.

O mestre de uma *piconet* é quem dita a seqüência de saltos e a fase nesta seqüência, através do seu endereço e do seu relógio. Assim que uma *piconet* é formada, a comunicação entre o mestre e o escravo pode começar. O canal é dividido em intervalos de 625 μ s, chamados *time slots*, onde um diferente salto de freqüência é usado para cada *slot*. O canal é compartilhado entre o mestre e o escravo usando salto de freqüências e duplexação por divisão de tempo (TDD - *Time-Division Duplex*) para transmissão bidirecional. O mestre deve começar a transmissão apenas em *time slots* pares e os dispositivos escravos em *time slots* ímpares (Figura 2.5).



Figura 2.4: *Piconet* Bluetooth (adaptado de [BUH02])

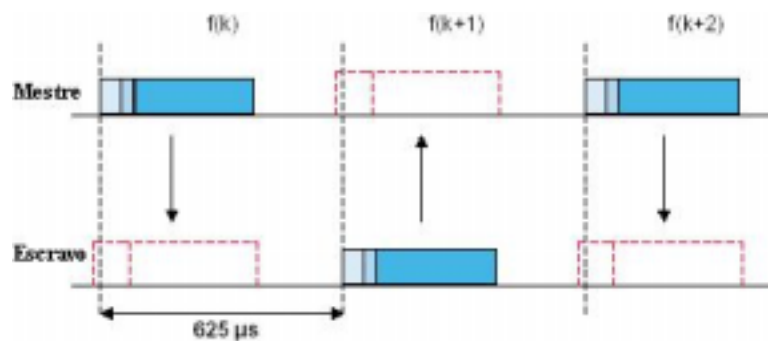


Figura 2.5: TDD (adaptado de [BLU01a])

Em algumas aplicações, diferentes dispositivos em diferentes *piconets* podem querer se comunicar uns com os outros. É definida então uma estrutura chamada *scatternet* (ou redes de dispersão) que facilita a comunicação entre diferentes *piconets*. As conexões são formadas por nós ponte, que fazem parte de duas ou mais *piconets*. Um nó ponte fica numa *piconet* por algum tempo e pode mudar para outra trocando sua seqüência de saltos. Um nó ponte pode ser um escravo em ambas *piconets* ou escravo em uma e mestre em outra. Na Figura 2.6 é mostrado um exemplo de uma *scatternet* formada por duas *piconets*. O nó ponte age como escravo na *piconet* 1 e como mestre na *piconet* 2. Entretanto, a especificação Bluetooth não define como as *scatternets* devem se organizar. Formação de *scatternets*, roteamento de pacotes e escalonamento do nó ponte são tópicos recentes de pesquisa em redes Bluetooth.

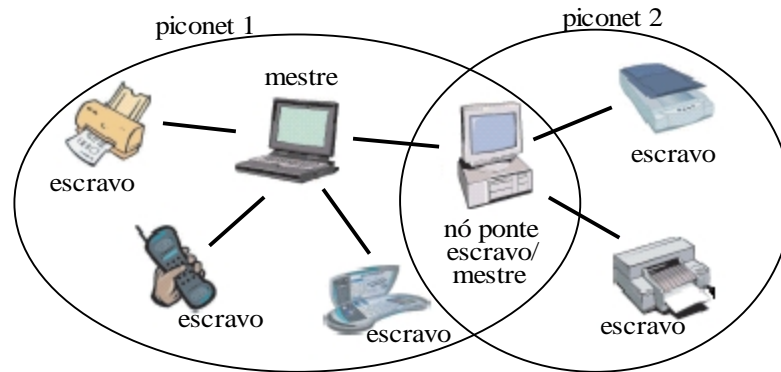


Figura 2.6: *Scatternet* Bluetooth

Há dois tipos de enlaces físicos possíveis em uma *piconet*: síncrono, orientado a conexão (SCO - *Synchronous Connection-Oriented*) e assíncrono, sem conexão (ACL - *Asynchronous Connectionless*). O SCO é um enlace ponto-a-ponto, simétrico, entre o mestre e um escravo, mantido pelo mestre em *slots* reservados a intervalos regulares. Sua aplicação principal é o tráfego de voz. O ACL é um enlace ponto-a-multiponto entre o mestre e todos os escravos da *piconet*, principalmente para tráfego de dados.

Todos os dados no canal são transportados em pacotes. São definidos 15 tipos de pacotes para a camada banda base. Os pacotes são ID, NULL, POLL, FHS e DM1, definidos para enlaces SCO e ACL. Os pacotes DH1, AUX1, DM3, DH3, DM5 e DH5 são definidos apenas para enlaces ACL. Os pacotes HV1, HV2, HV3 e DV são definidos apenas para enlaces SCO. Cada pacote consiste de três campos: o código de acesso (68/72 bits), o cabeçalho (54 bits) e o *payload* (0-2745 bits). A Figura 2.7 mostra a estrutura de um pacote:

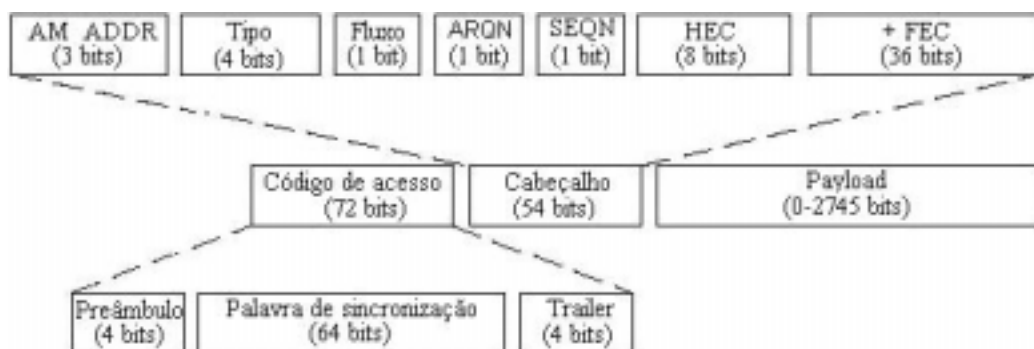


Figura 2.7: Formato de um pacote

O código de acesso é usado para sincronização e compensação de *offset*. Há três tipos de código de acesso: código de acesso de canal (CAC), código de acesso de dispositivo

(DAC) e código de acesso de inquiry (IAC). O cabeçalho contém informação sobre o endereço de escravo ativo da *piconet* (AM_ADDR), tipo de pacote, controle de fluxo, reconhecimento de pacotes (ARQN), numeração de pacotes para reordenação de pacotes fora de ordem (SEQN) e verificador de erro para o cabeçalho (HEC). O campo FEC (*Forward Error Correction*) é um código de repetição de 3 bits usado para proteger o cabeçalho. O código de repetição é implementado repetindo cada bit do cabeçalho 3 vezes. O *payload* pode conter campos de dados, voz ou ambos. O formato do *payload* depende do tipo de pacote. O formato básico para os pacotes assíncronos é mostrado na Figura 2.8:



Figura 2.8: *Payload* de pacotes assíncronos

O cabeçalho especifica o canal lógico, controla o fluxo entre os canais e possui um indicador do tamanho do *payload*. Dependendo do tipo de pacote, o cabeçalho possui 1 ou 2 bytes. O número de bytes de dados também é variável de acordo com o pacote. A maioria dos pacotes é protegida por um esquema de retransmissão ARQ (*Automatic Retransmission Query*) baseado num campo CRC (*Cyclic Redundancy Check*). Se algum erro é detectado a unidade receptora indica isso no próximo pacote retornado. Alguns pacotes possuem detecção e correção de erros FEC com um algoritmo de Hamming (15,10). A tabela 2.2 mostra estas informações para cada tipo de pacote assíncrono.

Os pacotes ID, NULL, POLL e FHS são pacotes especiais definidos pela especificação. O pacote ID tem tamanho fixo de 68 bits e é usado nas rotinas de *inquiry* e *paging* para fornecer o DAC ou o IAC. O pacote NULL tem tamanho fixo de 126 bits. É usado para retornar informações do destino para a fonte, como indicar o sucesso de uma transmissão. O pacote POLL é usado pelo mestre de uma *piconet* para fazer a varredura dos escravos, que devem responder mesmo que não tenham informações para transmitir. O pacote FHS é um pacote especial de controle que dá informações como o endereço Bluetooth (BD_ADDR) e o relógio do transmissor.

Tabela 2.2: Pacotes assíncronos

Pacote	Time slots	Dados (bytes)	FEC	CRC e ARQ	Taxa Máxima Simétrica (kbps)	Taxa Máxima Assimétrica	
						Direto (kbps)	Reverso (kbps)
DM1	1	0-17	Sim	Sim	108,8	108,8	108,8
DH1	1	0-27	Não	Sim	172,8	172,8	172,8
DM3	3	0-121	Sim	Sim	258,1	387,2	54,4
DH3	3	0-183	Não	Sim	390,4	585,6	86,4
DM5	5	0-224	Sim	Sim	286,7	477,8	36,3
DH5	5	0-339	Não	Sim	433,9	723,2	57,6
AUX1	1	0-29	Não	Não	185,6	185,6	185,6

O processo de formação de um enlace Bluetooth consiste de dois processos: *inquiry* (investigação) e *page* (negociação da conexão). O objetivo do *inquiry* é fazer com que o mestre descubra dispositivos vizinhos e colete informações destes nós (endereço e relógio). O objetivo do *page* é usar a informação obtida no processo de *inquiry* para estabelecer um canal de comunicação.

Durante o processo de *inquiry*, um dispositivo entra no estado de *inquiry* e outro no estado *inquiry scan*. O nó no estado de *inquiry* alterna repetidamente entre a transmissão de pacotes ID contendo um IAC e a escuta por respostas. Um dispositivo em *inquiry scan* escuta por pacotes ID dos dispositivos no estado de *inquiry* e responde com um pacote FHS (que contém o endereço e relógio). No processo de *inquiry* os nós saltam em 32 frequências. Como a fase de cada nó é determinada pelo relógio local, os nós geralmente estão fora de fase. Para facilitar a frequência de sincronização dentro de um tempo razoável, um nó no estado de *inquiry* salta bem mais rápido que um nó em *inquiry scan*. Vários nós em *inquiry scan* podem receber simultaneamente mensagens de *inquiry* de um mesmo nó. Para evitar que respondam ao mesmo tempo, cada nó escolhe um intervalo de *backoff* aleatório, entre 0 e 1023 *time slots*. O processo de *inquiry* é mostrado na Figura 2.9.

O processo de *page* é parecido com o de *inquiry*. Um dispositivo no estado de *page* usa a informação obtida durante o *inquiry* e envia trens de pacotes ID baseados no endereço

do dispositivo descoberto. Quando o dispositivo no estado *page scan* responde, o nó no estado de *page* envia um pacote FHS com a informação de sincronização para estabelecer uma conexão mestre-escravo e entrar no estado conectado. O dispositivo no estado de *page* se torna o mestre e o dispositivo em *page scan* se torna escravo. A Figura 2.10 mostra este processo.

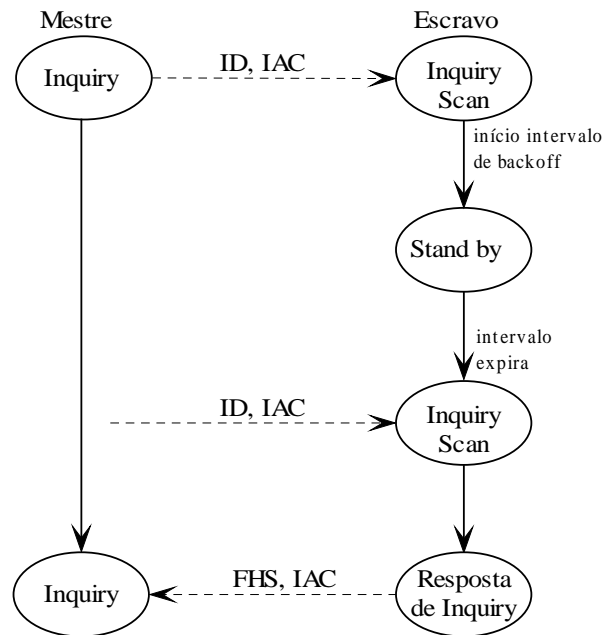


Figura 2.9: Transições de estado do processo de *inquiry*

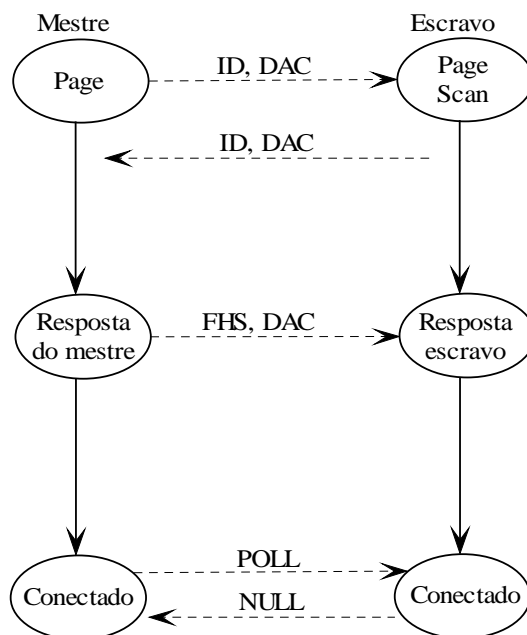


Figura 2.10: Transições de estado do processo de *page*

Como um dos objetivos do Bluetooth é a economia de energia, existem quatro modos de operação para os dispositivos no estado conectado: ativo, *sniff*, *hold* e *park*. No modo ativo, a comunicação ocorre normalmente. Os outros caracterizam modos de baixo consumo de energia. 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 baixo consumo por um intervalo de tempo fixo, após o qual volta a atividade normal. O modo *park* é similar ao *sniff*, mas o escravo libera seu endereço AM_ADDR, permanecendo sincronizado a *piconet*. Um dispositivo em modo *park* não participa da comunicação de dados na *piconet*.

2.3.3. Protocolo Gerenciador de Enlace - LMP

O LMP (*Link Manager Protocol*) situa-se na camada de enlace de dados e é responsável pela configuração e gerenciamento das conexões banda base. Procedimentos de autenticação e criptografia e os modos de operação de baixo consumo de energia são definidos na banda base, mas são configurados através de transações LMP entre dois nós. Parâmetro de QoS também podem ser negociados em transações LMP. As mensagens LMP são sempre enviadas em pacotes de um *slot* e têm prioridade sobre os pacotes de dados das camadas superiores. As mensagens podem ser agrupadas em três funções: gerenciamento da *piconet*, configuração de enlace e segurança.

2.3.4. Interface Controladora do Host - HCI

A HCI (*Host Controller Interface*) fornece ao *host* uma interface de comandos para se fazer acesso ao gerenciador de enlace e ao controlador de banda base e para se conhecer o estado do hardware e dos registradores de controle. Essa interface provê um método uniforme de acesso às funcionalidades da banda base. Os protocolos Bluetooth foram divididos de forma que uma parte fosse implementada em software (*host*) e outra em hardware (dispositivo físico e *firmware*). Isso não impede a fabricação de dispositivos com a tecnologia Bluetooth implementada totalmente em hardware. A HCI é constituída de três partes: um módulo de *driver* no *host*, um de *firmware* e um canal de transporte entre esses dois.

2.3.5. Protocolo de Adaptação e Controle de Enlace Lógico - L2CAP

L2CAP (*Logical Link Control and Adaptation Protocol*) é o protocolo que fornece serviços para as camadas superiores e esconde dessas os detalhes de implementação das camadas de rádio e banda base. É definido apenas para enlaces assíncronos e usa o conceito de canais lógicos. Cada canal lógico é identificado por um CID (*Channel Identification*). Os canais podem ser sem conexão ou orientados a conexão. Nos orientados a conexão, para cada fluxo L2CAP é estabelecido um canal entre os dois pontos. Os canais sem conexão restringem o fluxo de dados em um único sentido. Pode haver vários canais lógicos entre dois nós, mas todos sobre um mesmo enlace ACL. L2CAP faz também a multiplexação de protocolos de camadas superiores e realiza a segmentação e remontagem de pacotes, necessária para permitir a utilização, por parte das camadas superiores, de pacotes maiores que o tamanho máximo aceito pela banda base.

2.3.6. Camadas Superiores

Os principais protocolos que se situam sobre a camada L2CAP são o protocolo de descoberta de serviços (SDP - *Service Discovery Protocol*), emulação de porta serial (RFCOMM) e o protocolo de controle de telefonia (TCS - *Telephony Control Specification*). O SDP é um padrão para que dispositivos Bluetooth possam prestar e descobrir serviços suportados por outros dispositivos. É um protocolo cliente-servidor. O servidor mantém uma lista dos serviços disponíveis e suas características. Um cliente pode ter acesso aos serviços disponíveis no servidor e obter informações específicas sobre determinado serviço. SDP define um método padrão para descrição dos atributos dos serviços. A especificação define os serviços mais comuns, mas desenvolvedores podem criar subclasses de serviços padrão ou serviços totalmente novos.

A camada RFCOMM realiza a emulação de portas seriais RS-232, para aplicações que necessitam desta funcionalidade. Foi desenvolvida para que aplicações que utilizem comunicação serial possam ser portadas para o Bluetooth com pouca ou nenhuma modificação. O TCS define a sinalização de controle de chamadas para o estabelecimento de chamadas de voz entre dispositivos Bluetooth.

O Bluetooth SIG também adotou alguns protocolos já existentes, como o OBEX (*Object Exchange*), desenvolvido pela IrDA (*Infrared Data Association*) para a troca de objetos. Suporta o transporte de alguns objetos pré-definidos, como o *vCard*, *vCalendar*,

vMessage e *vNote*. Os protocolos WAP e TCP/IP também podem ser usados no Bluetooth, usando PPP (*Point-to-Point Protocol*) sobre a camada RFCOMM. Outras soluções para usar TCP sobre Bluetooth são propostas nos perfis [BLU01b].

2.3.7. Perfis

Um perfil é a descrição técnica de como implementar uma determinada aplicação. Os perfis definem os procedimentos e protocolos necessários para implementar uma aplicação específica com base nos protocolos comuns do Bluetooth. Inicialmente foram definidos 13 perfis de aplicação [BLU01b]: perfil de acesso genérico, aplicação para descoberta de serviço, porta serial, troca genérica de objetos, telefonia sem fio, *intercom*, fone de ouvido, acesso a LAN, acesso discado à rede, fax, transferência de arquivos, sincronização e troca de objetos. Além destes, outros perfis estão sendo lançados pelo Bluetooth SIG, como perfil de impressão, imagem, controle remoto, acesso à rede digital de serviços integrados, distribuição de áudio e vídeo, entre outros. Existem vários grupos de trabalho pesquisando outras aplicações e que podem ser incorporadas em futuras versões da especificação.

2.4. Outras Tecnologias para Redes sem Fio

As tecnologias sem fio que são consideradas concorrentes e/ou complementares ao Bluetooth podem ser divididas de acordo com a sua aplicação: em WLANs ou WPANs. As WPANs são redes sem fio de pequeno alcance (poucos metros) e as WLANs são redes sem fio para transmissões de maior alcance (algumas dezenas de metros) e maior volume de dados. Podem ser consideradas redes ainda menores, as chamadas BANs (*Body Area Networks*) [WWR01], ou redes do corpo, onde o alcance da rede é apenas o corpo de uma pessoa. As WLANs e WPANs são complementares, permitindo aos usuários que selecionem aquela mais adequada aos seus propósitos.

As WLANs são normalmente usadas como uma rede infra-estruturada, assim como as redes celulares, usando pontos de acesso para conectar usuários sem fio à rede local ou à Internet. As principais tecnologias são os padrões IEEE 802.11 e HiperLAN [COR03]. Os padrões IEEE são o IEEE 802.11b (Wi-Fi), IEEE 802.11a e IEEE 802.11g. O mais utilizado é o Wi-Fi, que é o mais antigo e opera na mesma banda de 2,4 GHz do Bluetooth. Sua taxa atinge até 11 Mbps, existindo vários produtos comerciais disponíveis e um grande número de redes locais que utilizam este padrão. O IEEE 802.11a opera na faixa de 5 GHz com taxas de

até 54 Mbps e o IEEE 802.11g na banda ISM de 2,4 GHz com taxas de até 20 Mbps. Estas duas últimas ainda não são largamente utilizadas por serem mais recentes, mas devem ter um grande aumento nos próximos anos. O HiperLAN/2 é um padrão desenvolvido pelo ETSI (*European Telecommunications Standards Institute*) que opera na faixa de 5 GHz com velocidade de até 54 Mbps. É um concorrente direto dos padrões IEEE pelo mercado das WLANs. Estes padrões apresentados para WLANs têm um custo bem mais alto que os das WPANs, pois atingem taxas de transmissão bem mais altas e consomem mais energia.

As WPANs visam criar redes sem fio heterogêneas com equipamentos como computadores pessoais, telefones celulares, PDAs, impressoras e até eletrodomésticos. As principais tecnologias são o IrDA, HomeRF, Bluetooth e os padrões em desenvolvimento IEEE 802.15 [COR03]. O IrDA utiliza ondas de luz infravermelha para transmissão de dados, chegando a taxas de 4 Mbps. Tem um custo muito baixo, mas requer linha de visada entre os dispositivos para a comunicação, limitando o seu uso em muitas aplicações.

O HomeRF foi desenvolvido para formar redes sem fio entre computadores e outros dispositivos em residências. Opera na faixa de 2,4 GHz e atinge até 10 Mbps. Pode ser uma opção para formar WLANs em residências e pequenos escritórios, devido ao menor custo se comparado às redes IEEE 802.11. Porém o custo é alto em relação às tecnologias para WPANs, principal razão da pouca aceitação do padrão até o momento.

Os padrões IEEE 802.15 surgiram como necessidade de padronizar os diferentes tipos de redes pessoais sem fio, assim como aconteceu com as redes locais sem fio [COR03]. O IEEE 802.15.1 é derivado do Bluetooth, sendo que as camadas mais baixas do protocolo foram adotadas. Foi um passo importante para o Bluetooth, que se fixa definitivamente como um padrão para WPANs. O IEEE 802.15.2 são práticas recomendadas para facilitar a coexistência de dispositivos que usam a banda livre ISM, como Bluetooth, Wi-Fi e até fornos de microondas e equipamentos médicos. A interferência entre estes dispositivos pode causar sérios problemas, tanto que a versão 1.2 da especificação Bluetooth traz novas técnicas para minimizar a interferência.

O padrão IEEE 802.15.3 ainda está em desenvolvimento e será para WPANs de alta velocidade (20 Mbps ou mais), especialmente para aplicações multimídia. O IEEE 802.15.4 também está sendo desenvolvido, para WPANs de baixa velocidade (200 kbps ou menos) e baixa potência. Pode vir a ser uma das principais tecnologias para redes de sensores.

2.5. Conclusão

Neste capítulo foram apresentados os conceitos básicos de redes móveis sem fio e de propagação de ondas de rádio. Bluetooth é uma das tecnologias mais promissoras para as redes sem fio e foi a escolhida para o trabalho desta dissertação, sendo abordada com mais detalhes. As características do Bluetooth mais importantes para a compreensão dos próximos capítulos são as camadas de rádio e banda base, especialmente os enlaces e pacotes assíncronos, processos de *inquiry* e *page* e o conceito de *piconets* e *scatternets*. No próximo capítulo será apresentada uma modelagem para análise de desempenho de enlaces de dados Bluetooth.

Capítulo 3

Modelagem de Enlaces Bluetooth em Canais com Desvanecimento Nakagami- m

Um sinal que se propaga num canal sem fio é afetado por diversos fatores, como desvanecimento, perda de percurso e interferência co-canal. Dispositivos operando na mesma faixa de frequências podem causar interferência entre si. A mobilidade dos nós e a distância entre transmissor e receptor causa mudanças no desvanecimento experimentado pelo sinal de rádio. Assim, a qualidade do canal varia com o tempo e depende também da localização dos dispositivos. Estimações sobre o canal podem ser feitas usando modelos estatísticos. A vazão de enlaces Bluetooth usando pacotes assíncronos foi calculada em [VAL02a] para o canal Gaussiano (AWGN – *Additive White Gaussian Noise*) e um canal com desvanecimento Rayleigh. Neste capítulo é apresentada a modelagem analítica para cálculo da vazão e da probabilidade de erro de pacotes em enlaces Bluetooth para canais com desvanecimento Nakagami- m . A distribuição de Nakagami abrange uma ampla variedade de condições de desvanecimento através do parâmetro m . O parâmetro m de Nakagami indica a qualidade do canal, considerando desvanecimento e interferência. Também é utilizado um modelo logarítmico de perda de percurso para relacionar as probabilidades de erro de pacotes com a distância entre os nós.

3.1. Análise de Vazão

Para análise da vazão (do inglês: *throughput*) foi utilizada a modelagem descrita em [VAL02a]. A vazão pode ser calculada usando a probabilidade de retransmissão dos pacotes.

A retransmissão de um pacote de dados (assíncrono) ocorre quando qualquer um dos cinco eventos acontece:

A: o destino falha em sincronizar com o código de acesso do pacote enviado;

B: o cabeçalho do pacote enviado está corrompido (depois que o código de repetição é decodificado);

C: os dados do pacote enviado estão corrompidos depois que o código de Hamming é decodificado, causando a falha na verificação de CRC;

D: a fonte é incapaz de sincronizar com o código de acesso do pacote de retorno;

E: o cabeçalho do pacote de retorno está corrompido.

Um pacote é retransmitido se qualquer um dos eventos A-E ocorrerem. A probabilidade de retransmissão é dada por:

$$P_r(\gamma_f, \gamma_r) = 1 - P[\bar{A}]P[\bar{B}]P[\bar{C}]P[\bar{D}]P[\bar{E}], \quad (3.1)$$

onde γ_f denota a relação sinal-ruído (SNR) instantânea do canal direto e γ_r a SNR instantânea do canal reverso. O evento \bar{A} indica o complemento do evento A , $P[A]$ é a probabilidade de ocorrência do evento A e assim sucessivamente. Destes cinco eventos, três (A, B e C) se referem ao canal direto, isto é, o pacote de dados que foi enviado. Como um pacote ACL de dados tem três campos (código de acesso, cabeçalho e *payload*) com diferentes esquemas de correção de erros, três eventos de erro podem ocorrer. Para o canal reverso apenas dois eventos (D e E) podem ocorrer, pois o pacote de retorno NULL usado para indicar o sucesso (ACK) ou insucesso (NAK) de uma transmissão não possui dados, apenas os campos de código de acesso e cabeçalho.

A sincronização é feita correlacionando a saída do demodulador com uma cópia gravada do código de acesso. Um pacote é sincronizado se a saída do correlator excedeu um determinado limiar T . O quadro é sincronizado se pelo menos T dos 72 bits do código de acesso forem demodulados corretamente. Quanto menor o valor de T maior a probabilidade de o pacote ser sincronizado. A probabilidade de sincronização com o pacote enviado acontece se não há mais que $(72 - T)$ erros no código de acesso recebido:

$$P[\bar{A}] = \sum_{k=0}^{72-T} \binom{72}{k} \cdot [p(\gamma_f)]^k \cdot [1 - p(\gamma_f)]^{72-k}, \quad (3.2)$$

onde $p(\gamma_f)$ é a probabilidade de erro de símbolo do canal direto. Como o pacote de retorno também tem um código de acesso de 72 bits, o evento D tem a mesma forma,

$$P[\overline{D}] = \sum_{k=0}^{72-T} \binom{72}{k} \cdot [p(\gamma_r)]^k \cdot [1 - p(\gamma_r)]^{72-k}, \quad (3.3)$$

onde $p(\gamma_r)$ é a probabilidade de erro de símbolo do canal reverso. O evento B ou E ocorre se qualquer das oito triplas do código de repetição (3,1) for decodificada incorretamente,

$$P[\overline{B}] = \{3p(\gamma_f)[1 - p(\gamma_f)]^2 + [1 - p(\gamma_f)]^3\}^{18} \quad (3.4)$$

$$P[\overline{E}] = \{3p(\gamma_r)[1 - p(\gamma_r)]^2 + [1 - p(\gamma_r)]^3\}^{18} \quad (3.5)$$

O erro mais provável é do tipo C. Para os pacotes DHx ocorre quando qualquer bit dos dados for recebido com erro:

$$P[\overline{C}] = [1 - p(\gamma_f)]^b, \quad (3.6)$$

onde b é o tamanho do pacote em bits. Para DH1 tem-se que $b=240$, $b=1496$ para DH3 e $b=2744$ para DH5. Para os pacotes DMx os dados são protegidos por um código de Hamming, onde $b=16$ para DM1, $b=100$ para DM3 e $b=183$ para DM5. A probabilidade do evento C para os pacotes DMx é:

$$P[\overline{C}] = [15p(\gamma_f)[1 - p(\gamma_f)]^{14} + [1 - p(\gamma_f)]^{15}]^b \quad (3.7)$$

Bluetooth utiliza modulação GFSK com o produto tempo-largura de banda $BT = 0,5$ e índice de modulação h entre 0,28 e 0,35. A probabilidade de erro de símbolo $p(\gamma)$ para a modulação GFSK deve ser aplicada nas equações 3.2 a 3.6. Quando o índice de modulação do FSK é menor que $h = 0,5$, os sinais são correlacionados e a correlação é dada em [PRO01] por:

$$\rho = \frac{\text{sen}(2\pi h)}{2\pi h} \quad (3.8)$$

Sejam duas constantes a e b :

$$\begin{aligned} a &= \sqrt{\frac{\gamma}{2}(1 - \sqrt{1 - \rho^2})} \\ b &= \sqrt{\frac{\gamma}{2}(1 + \sqrt{1 - \rho^2})} \end{aligned} \quad (3.9)$$

A probabilidade de erro da modulação GFSK, onde $Q_1(a, b)$ é a função Q-Marcum e I_0 é a função de Bessel modificada de primeira ordem é dada em [PRO01]:

$$\begin{aligned} p(\gamma) &= Q_1(a, b) - \frac{1}{2} e^{(a^2+b^2)/2} I_0(ab) \\ &= e^{-\gamma/2} \left\{ \frac{1}{2} I_0(ab) + \sum_{k=1}^{\infty} \left(\frac{a}{b}\right)^k I_k(ab) \right\} \end{aligned} \quad (3.10)$$

Pode-se definir uma variável aleatória N como o número total de vezes que um pacote deve ser transmitido. A probabilidade que $N = n$ é o produto da probabilidade de falha durante as primeiras $n-1$ tentativas e a probabilidade de sucesso durante a n -ésima transmissão. Assumindo que o canal tem desvanecimento lento e considerando que a função de densidade de probabilidade (pdf) da relação sinal-ruído é a mesma de pacote para pacote, a probabilidade que um pacote seja transmitido n vezes é

$$p_N[n] = (1 - \bar{P}_r(\bar{\gamma})) (\bar{P}_r(\bar{\gamma}))^{n-1} \quad (3.11)$$

onde $\bar{\gamma}$ é a SNR média recebida e $\bar{P}_r(\bar{\gamma})$ é a probabilidade média de retransmissão. Para qualquer tipo de pacote, a taxa de dados R é calculada em função de N

$$R = \frac{K}{D \cdot N \cdot 625 \cdot 10^{-6}}, \quad (3.12)$$

onde D é o número de *time slots* ocupados incluindo o pacote de retorno (dois para Dx1, quatro para Dx3 e seis para Dx5) e K é o número de bits de dados do pacote (136 para DM1, 968 para DM3, 1792 para DM5, 216 para DH1, 1464 para DH3 e 2712 para DH5). A vazão média é encontrada pelo valor médio esperado de R em relação a N

$$R_{med} = E_N\{R\} = \sum_{n=1}^{\infty} p_N[n] \cdot R \quad (3.13)$$

A probabilidade média de retransmissão é dada por

$$\bar{P}_r(\bar{\gamma}) = 1 - \int_0^\infty f(\gamma_f) P[\bar{A}] P[\bar{B}] P[\bar{C}] d\gamma_f \cdot \int_0^\infty f(\gamma_r) P[\bar{D}] P[\bar{E}] d\gamma_r, \quad (3.14)$$

onde $f(\gamma_f)$ e $f(\gamma_r)$ são as funções de densidade de probabilidade do canal direto e reverso, respectivamente. Se o transmissor e o receptor têm o mesmo tipo de demodulador, $p_f(\gamma_f) = p_r(\gamma_r) = p(\gamma)$. Ainda, a reciprocidade dos canais implica que $f(\gamma_f) = f(\gamma_r) = f(\gamma)$. A taxa de erro de pacote (PER) do canal direto, PER_f , e reverso, PER_r , podem ser definidas por

$$PER_f = 1 - \int_0^\infty f(\gamma_f) P[\bar{A}] P[\bar{B}] P[\bar{C}] d\gamma_f \quad (3.15)$$

$$PER_r = 1 - \int_0^\infty f(\gamma_r) P[\bar{D}] P[\bar{E}] d\gamma_r \quad (3.16)$$

3.2. Desempenho em Canais com Desvanecimento Nakagami- m

A distribuição de Nakagami vem sendo muito usada para avaliar o desempenho de sistemas de comunicações móveis e sem fio, pois possibilita uma ampla variedade de condições de desvanecimento e tem uma boa aproximação com dados reais de medição. A modelagem do canal tem impacto direto na avaliação de desempenho de redes sem fio. A distribuição de Nakagami modela estatisticamente as variações rápidas do sinal, causadas por movimentações no ambiente ou dos próprios nós da rede. Ela pode ser reduzida para uma distribuição Rayleigh (quando $m = 1$) ou modelar condições de desvanecimento mais severas ($m < 1$) ou menos severas ($m > 1$) que Rayleigh. Quando $m \rightarrow \infty$, o desvanecimento de Nakagami converge para um canal AWGN. Para $m \geq 1$, um mapeamento entre o parâmetro m de desvanecimento e o fator K de Rician permite uma aproximação da distribuição Rice. A função de densidade de probabilidade de Nakagami [PRO01] é dada por

$$f(\gamma) = \frac{m^m \gamma^{m-1}}{\Gamma(m) \bar{\gamma}^m} \exp\left(-\frac{m\gamma}{\bar{\gamma}}\right), \quad m \geq 0,5 \quad (3.17)$$

onde Γ é a função gama. Substituindo-se a equação (3.17) em (3.14), pode-se calcular a vazão usando (3.13) para diferentes valores de m . Foi utilizado um índice de modulação $h = 0,32$ e um limiar $T = 65$. Estes são valores típicos do Bluetooth e foram usados em [VAL02a].

As Figuras 3.1 a 3.6 mostram o desempenho para cada tipo de pacote de dados. Os cálculos foram feitos com o software *Matlab*® [MAT00]. Foram implementadas funções no editor do *Matlab* para cálculo de todos os parâmetros necessários. Para $m = 1$, a vazão é a mesma do canal com desvanecimento Rayleigh. Os resultados mostram que o parâmetro m afeta diretamente o desempenho de enlaces Bluetooth.

Quanto maior o valor de m melhor é o desempenho, pois a condição do canal é menos severa. Quanto menor for o valor de m , indicando más condições do canal, menor será a vazão. Para valores altos da relação sinal-ruído a vazão converge para a taxa máxima de cada pacote. Para $m = 0,5$ o desvanecimento é mais severo e a taxa máxima de cada pacote só é atingida para uma relação sinal-ruído em torno de 45 dB. Para $m = 1,5$, o desvanecimento é menos severo e a vazão converge para a taxa máxima em valores mais baixos da relação sinal-ruído.

As Figuras 3.7 a 3.12 mostram a vazão em função da taxa de erro de pacote (PER) no canal direto, PER_f , e no canal reverso, PER_r . Para todos os pacotes, quanto maior a probabilidade de erro, menor a vazão. É importante notar que a PER_f é sempre maior que PER_r , pois o pacote de retorno é menor. Para uma determinada vazão média em que a PER_f já está alta, a PER_r ainda estará muito baixa. É por isso que a queda da vazão aparece mais acentuada nos gráficos de PER_r .

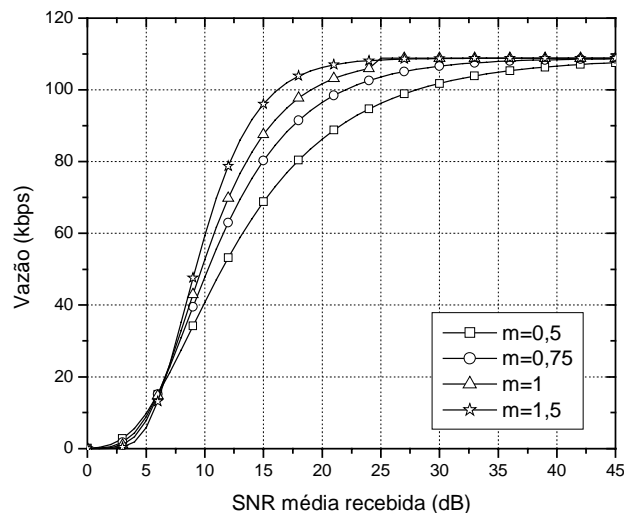


Figura 3.1: Desempenho do pacote DM1

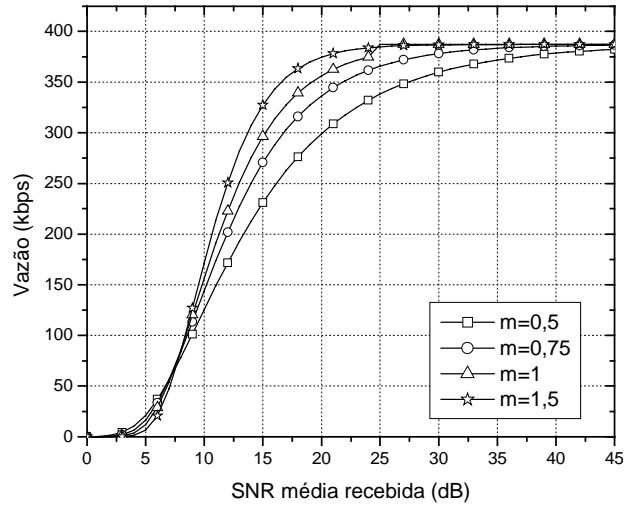


Figura 3.2: Desempenho do pacote DM3

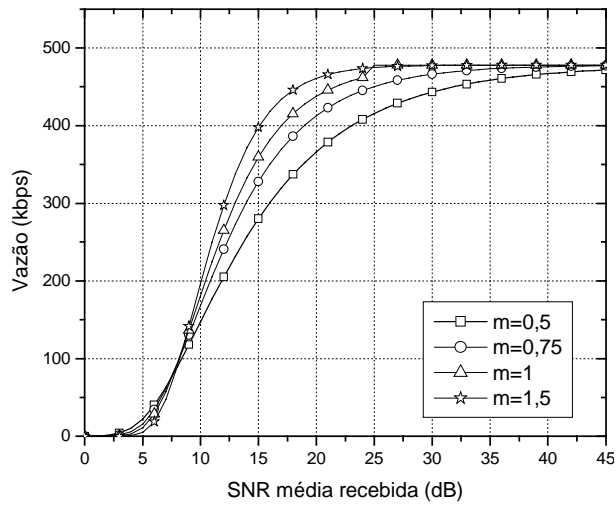


Figura 3.3: Desempenho do pacote DM5

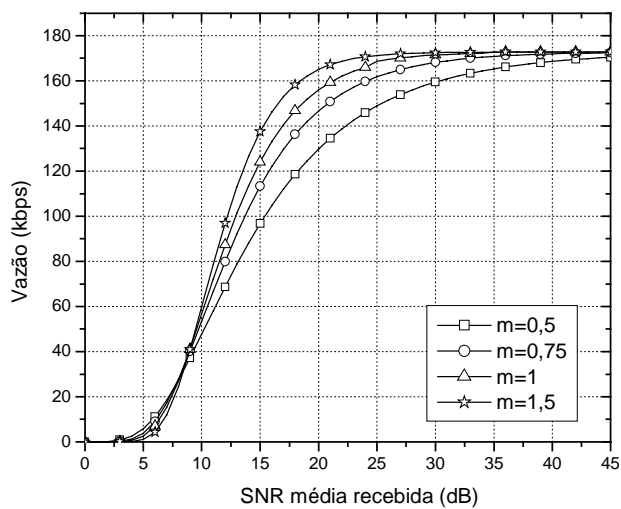


Figura 3.4: Desempenho do pacote DH1

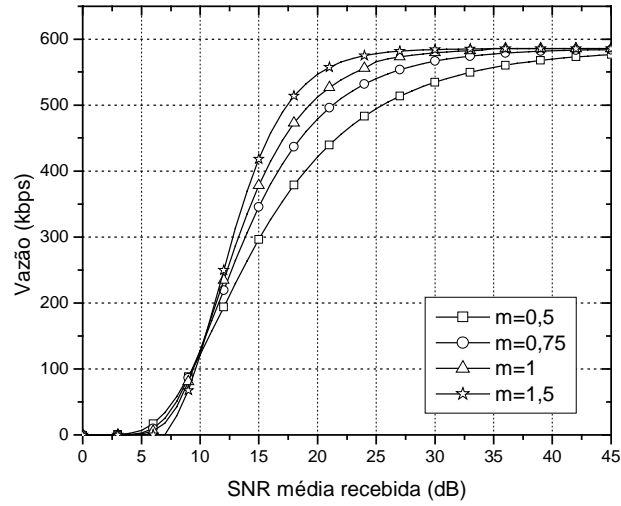


Figura 3.5: Desempenho do pacote DH3

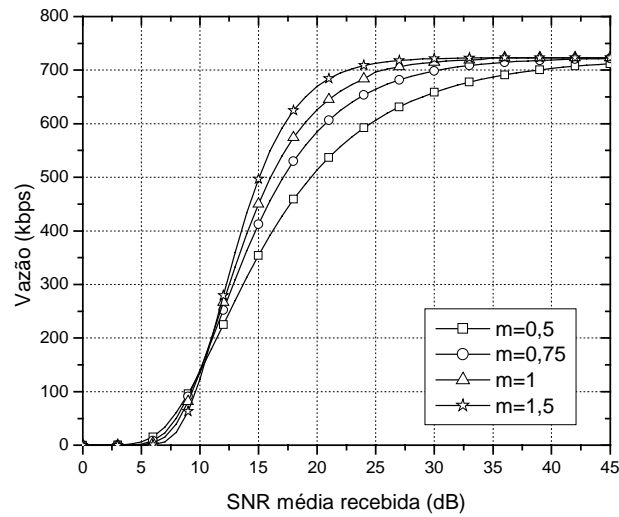


Figura 3.6: Desempenho do pacote DH5

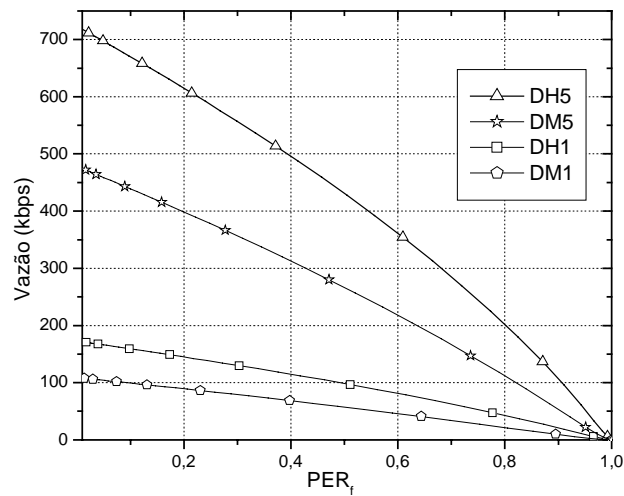


Figura 3.7: Vazão \times PER_f para $m = 0,5$

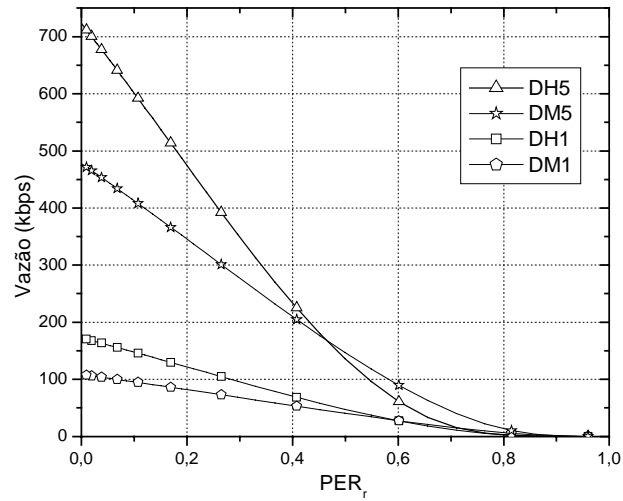


Figura 3.8: Vazão \times PER_r para $m = 0,5$

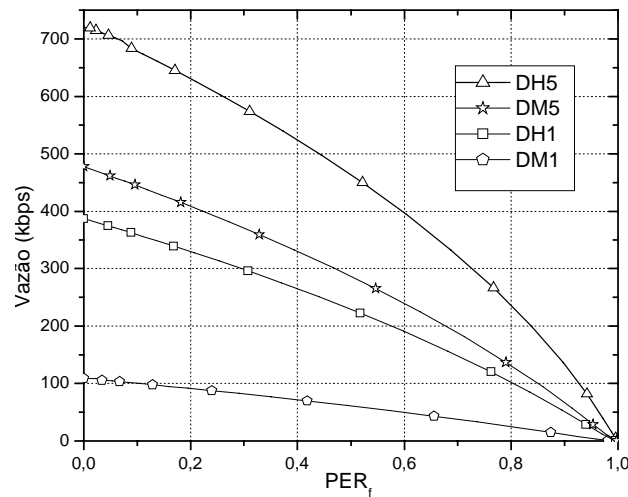


Figura 3.9: Vazão \times PER_f para $m = 1$

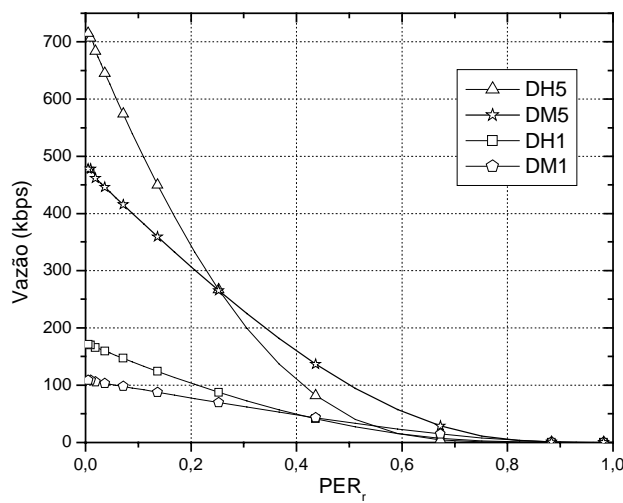


Figura 3.10: Vazão \times PER_r para $m = 1$

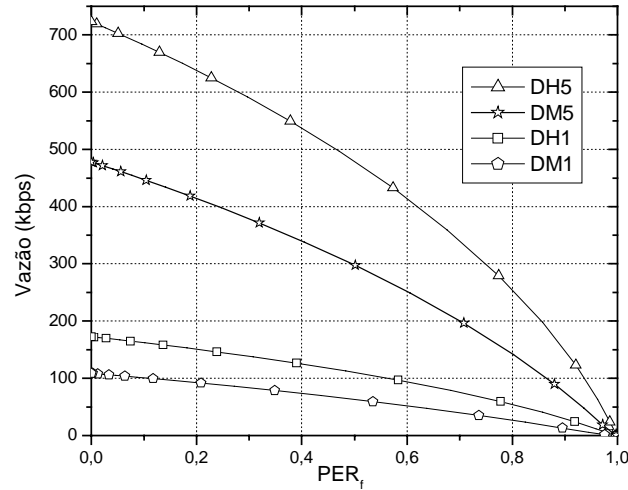


Figura 3.11: Vazão \times PER_f para $m = 1,5$

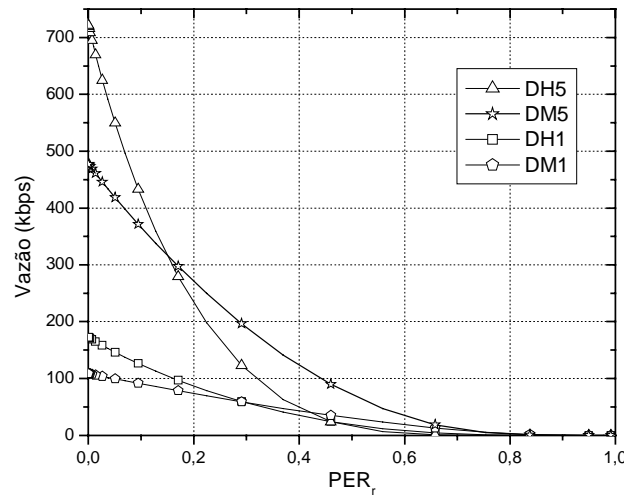


Figura 3.12: Vazão \times PER_r para $m = 1,5$

3.3. Modelo de Perda de Percurso

Para estimar a potência média recebida por dispositivos Bluetooth foi aplicado o modelo logarítmico de perda de percurso [RAP99]. Modelos de propagação teóricos e baseados em medição indicam que a potência média recebida do sinal diminui de forma logarítmica com a distância, tanto em canais de rádio dentro de construções (*indoor*) ou fora (*outdoor*). Este modelo tem sido usado extensivamente na literatura [RAP99]. No cálculo da perda de percurso foi sempre considerada a potência máxima transmitida para cada classe de

potência Bluetooth (ver Tabela 2.1). Um receptor Bluetooth requer uma potência mínima recebida de -70 dBm [BLUE01a]. O modelo logarítmico de perda de percurso é dado por

$$\overline{PL}(\text{dB}) = \overline{PL}(d_0) + 10 \cdot \eta \cdot \log\left(\frac{d}{d_0}\right), \quad (3.18)$$

onde η é o expoente de perda de percurso que indica a proporção em que a perda aumenta com a distância, d é a separação transmissor-receptor e d_0 é a distância de referência. O valor de n pode ser escolhido baseado no ambiente que os dispositivos irão operar. A perda de percurso de referência $\overline{PL}(d_0)$ é calculada usando o modelo de perda de percurso de espaço livre [RAP99],

$$P_r(d) = P_t \frac{\lambda^2}{(4\pi)^2 d^2}, \quad (3.19)$$

sendo P_t é a potência transmitida, d é a distância e λ é o comprimento de onda. O comprimento de onda pode ser obtido por $\lambda = c/f$, onde c é a velocidade da luz e f a frequência da portadora (2,4 GHz para Bluetooth). A potência recebida P_r é dada por

$$P_r = P_t - \overline{PL} \quad (3.20)$$

Assim, a relação sinal-ruído média recebida é

$$SNR = \frac{E_b}{N_0} = \frac{P_r}{R \cdot N_0}, \quad (3.21)$$

onde R é a taxa de transmissão (1 Mbps para Bluetooth) e N_0 é a densidade espectral de potência unilateral do ruído. O ruído térmico para uma largura de banda de 1 MHz é -114 dBm. A relação logarítmica entre a SNR de entrada e a SNR de saída de um circuito é definida como figura de ruído. Considerando um receptor Bluetooth com figura de ruído de 23 dB [KOU02] tem-se uma potência de ruído $N = N_0 B = -91$ dBm .

Para aplicar o modelo logarítmico de perda de percurso foram considerados dispositivos Bluetooth classe 1, que de acordo com a especificação possuem um alcance de 100 metros. Os resultados são mostrados apenas para a classe 1, pois as classes 2 e 3 terão o

mesmo comportamento, só que com um alcance menor. Por exemplo, a classe 3 terá um alcance aproximadamente 10 vezes menor que um dispositivo da classe 1.

Aplicando o modelo logarítmico de perda de percurso a SNR média recebida foi calculada para diferentes separações transmissor-receptor. Este resultado é mostrado na Figura 3.13 para um expoente de perda de percurso $\eta = 3$, $\eta = 3,5$ e $\eta = 4$. Uma linha indicando a potência mínima recebida de -70 dBm (sensibilidade do receptor) também é mostrada na Figura 3.13. A interseção desta linha com as curvas de SNR indicam o alcance máximo de transmissão. As Figuras 3.14 a 3.17 relacionam a distância transmissor-receptor e a taxa de erro de pacote dos pacotes assíncronos. Para o pacote de retorno (ACK) o eixo PER representa a probabilidade PER_r . Para os pacotes DM1, DM5, DH1 e DH5 o eixo PER representa a probabilidade PER_f .

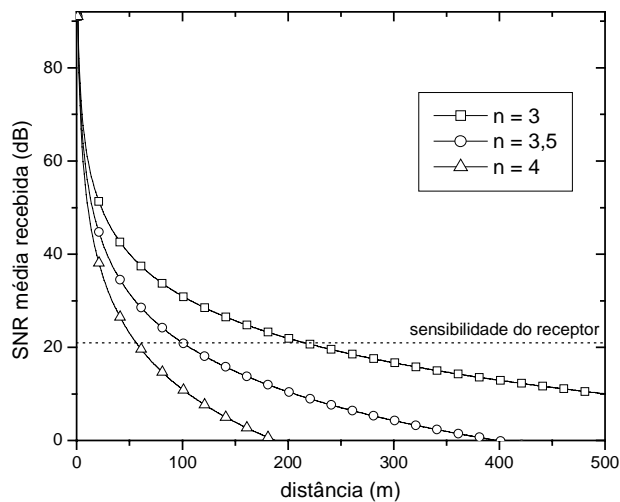


Figura 3.13: SNR \times distância

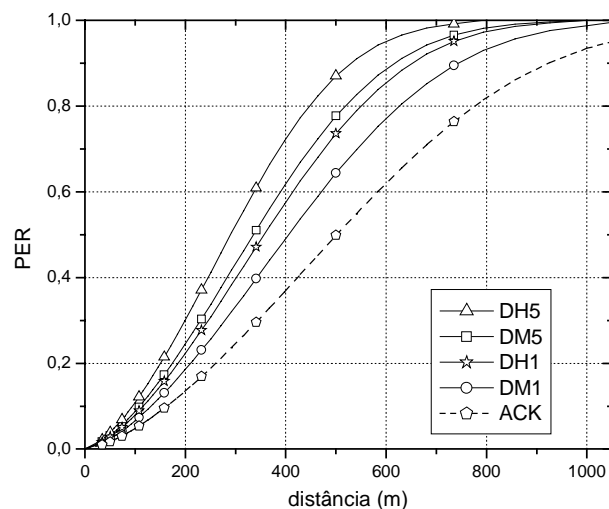


Figura 3.14: PER \times distância, $m = 0,5$, $\eta = 3$

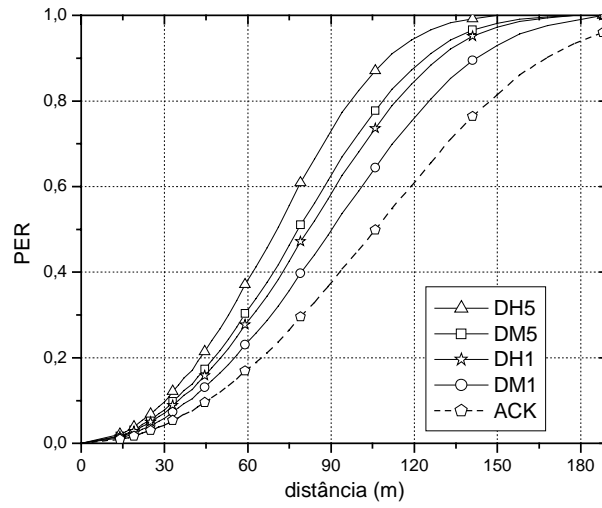


Figura 3.15: PER \times distância, $m = 0,5$, $\eta = 4$

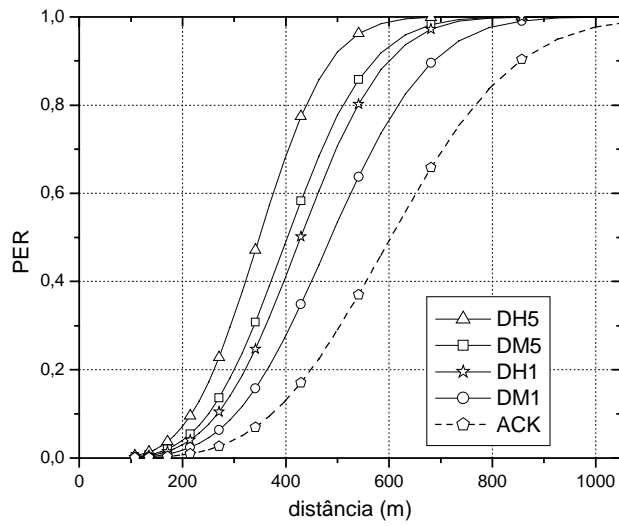


Figura 3.16: PER \times distância, $m = 1,5$, $\eta = 3$

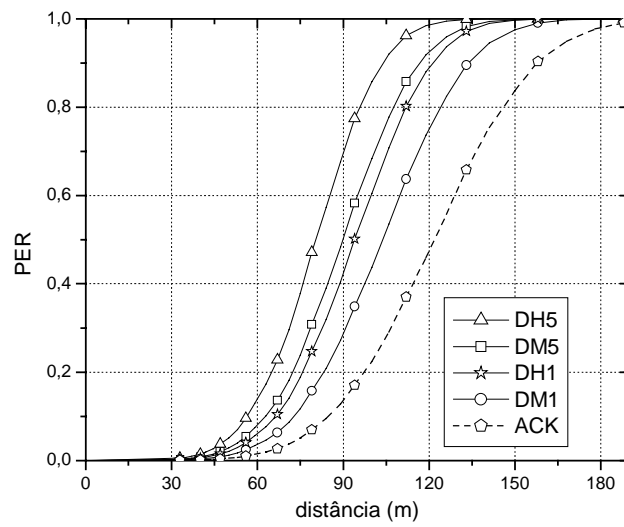


Figura 3.17: PER \times distância, $m = 1,5$, $\eta = 4$

3.4. Conclusão

Neste capítulo foram apresentadas expressões analíticas para cálculo da vazão e taxa de erro de pacotes para os seis pacotes assíncronos que usam retransmissão (ARQ). O desempenho destes pacotes foi analisado para canais com desvanecimento Nakagami- m . Com desvanecimento mais severo, os pacotes precisam de uma relação sinal-ruído muito alta para atingir a taxa máxima prevista na especificação [BLU01a]. Por exemplo, para uma vazão de 90% do total e uma condição de desvanecimento mais severa ($m = 0,5$) é necessária em média uma relação sinal-ruído adicional de 7,5 dB comparado com desvanecimento Rayleigh. O sétimo pacote assíncrono, AUX1, não utiliza retransmissão nem FEC e é entregue ao destino independente se contém erros ou não. Por isso, o pacote AUX1 pode ser usado para aplicar estratégias específicas de correção de erros, como é feito em [VAL02b] usando códigos turbo.

Uma relação entre taxa de erro de pacotes e a distância entre dispositivos foi apresentada para o canal direto e reverso, considerando um modelo logarítmico de perda de percurso. O alcance previsto na especificação do Bluetooth pode ser ainda maior em ambientes com boas condições de canal, porém em condições mais severas pode não atingir o alcance desejado. No próximo capítulo é abordada a simulação de redes Bluetooth, incluindo os softwares de simulação existentes e o desenvolvimento de novas ferramentas. Num simulador, a modelagem do canal de rádio é importante para indicar o sucesso ou insucesso da transmissão de um pacote. As taxas de erro de pacote são incorporadas em simuladores Bluetooth para que as condições do desvanecimento Nakagami possam ser simuladas em *piconets* e *scatternets*. Utilizando o desvanecimento Nakagami como modelo de erros, condições de canal mais realistas podem ser simuladas através do parâmetro m , desde condições boas de canal até condições mais severas.

Capítulo 4

Simulação de Redes Bluetooth

A simulação consiste na utilização de determinadas técnicas matemáticas, empregadas em computadores, as quais permitem imitar o funcionamento de, praticamente, qualquer tipo de operação ou processo (sistemas) do mundo real [FRE01]. É o processo de projetar um modelo computacional de um sistema real e conduzir experimentos com este modelo com o propósito de entender seu comportamento e/ou avaliar estratégias para sua operação. É largamente utilizada nas mais diversas áreas, como administração, economia, biologia e na própria ciência da computação.

Em redes de computadores a realização de simulações é o melhor caminho para obter boas estimativas do comportamento de um sistema após a alteração de um protocolo ou de uma topologia de rede. A tecnologia Bluetooth também se beneficia muito destas técnicas, especialmente por ser um padrão ainda em desenvolvimento e ter problemas totalmente em aberto, como no caso das *scatternets*. Neste capítulo serão apresentados os simuladores Bluetooth existentes e a incorporação do modelo de desvanecimento Nakagami nestes softwares. Também serão apresentados novos simuladores desenvolvidos para simular tarefas específicas de redes Bluetooth.

4.1. Ferramentas de Simulação Existentes

Para simular e avaliar o desempenho de redes Bluetooth existem dois simuladores desenvolvidos como uma extensão ao software *Network Simulator (ns-2)* [NS2]: *BlueHoc* [IBM00] e *Blueware* [TAN02c]. O *Network Simulator* é um simulador de redes baseado em eventos discretos e orientado a objetos escrito em linguagem C++. As simulações são desenvolvidas num interpretador na linguagem de script OTcl, que acessa as classes escritas

em C++. Modificações em protocolos ou adição de novas funcionalidades devem ser feitas em C++. O *ns* foi criado a partir de outros simuladores e hoje é mantido e distribuído gratuitamente pelo ISI (*Information Sciences Institute*), com versões para vários sistemas operacionais, como Linux, Unix e Windows. A versão utilizada neste trabalho foi o *ns 2.1b7* para Linux. Por ser gratuito e ter o código fonte aberto, é muito usado para pesquisas.

O *BlueHoc* [IBM00] foi desenvolvido pela IBM como uma extensão ao *ns* e implementa as principais características do Bluetooth, como rádio e banda base, camada L2CAP, protocolo LMP e HCI. A classe dos dispositivos utilizada tem alcance de dez metros (classe 3). Possui uma interface gráfica que permite a configuração de uma *piconet* Bluetooth com um mestre e até sete escravos (Figura 4.1).

O *BlueHoc* utiliza um modelo de erros para o canal sem fio, porém não há nenhuma descrição deste modelo. Pode ser configurado tráfego FTP ou *Telnet* entre os dispositivos para avaliar o desempenho da *piconet*. Porém o software não suporta *scatternets* Bluetooth, o que limita o seu uso. Há uma versão do *BlueHoc* chamada *Bluescat* que possui algumas modificações no código para suportar funcionalidades básicas de *scatternets*. Porém não possui nenhum algoritmo de formação de *scatternets* implementado e é bastante limitado. Cada simulação gera um arquivo de rastreamento (*trace*) com informações sobre os processo de conexão e pacotes trocados na rede, entre outros dados.

O simulador *Blueware* [TAN02c] foi desenvolvido em 2002 pelo laboratório de ciência da computação do Instituto de Tecnologia de Massachusetts (MIT). Foi feito a partir código do *BlueHoc*, com o objetivo de implementar também simulação de *scatternets* no simulador. Todas as funções do *BlueHoc* continuam presentes, exceto a interface gráfica. A configuração da simulação é feita no script OTcl. Também não possui nenhum modelo de erros para o canal de rádio, isto é, não há perda de pacotes.

O *Blueware* permite que sejam implementados vários protocolos para formação de *scatternets*, escalonamento e roteamento de pacotes. Isto é feito estendendo a classe *HCIEventsHandler*, que entrega os eventos relacionados à descoberta de dispositivos vizinhos e comunicação entre eles aos protocolos que se deseja implementar. No entanto, a implementação de novos algoritmos não é tão simples, pois a documentação do código nem sempre é clara e completa.

No simulador está implementado um esquema para formação de scatternet chamado TSF (*Tree Scatternet Formation*) [TAN02a]. Este algoritmo cria uma topologia em árvore e é

usado como o padrão nas simulações do *Blueware*. Suporta ambientes dinâmicos, onde nós podem entrar e sair da *scatternet* em diferentes instantes. Também está implementado no simulador um esquema de escalonamento de *scatternets*, chamado LCS (*Locally Coordinated Scheduling*) [TAN02b], que faz um escalonamento dinâmico baseado nas condições de tráfego da rede. O nó ponte fica mais tempo na *piconet* que tiver mais tráfego e menos tempo na *piconet* que tiver menos tráfego.

As opções de simulação incluem o número de nós na *scatternet*, distância entre nós (no máximo dez metros), tráfego CBR (*Constant Bit Rate*) ou TCP e duração da simulação. Modificações no código C++ podem ser feitas para melhorar ou mudar determinadas características do simulador. O *Blueware* ainda inclui vários arquivos de saída (*traces*) com os dados de cada simulação, indicando todo o processo de *inquiry* e *page*, formação da *scatternet*, tráfego e todos os pacotes Bluetooth trocados entre todos os dispositivos. Entre outras limitações, o *Blueware* não suporta conexões síncronas (SCO) e tem pouco suporte para QoS.

4.2. Extensões para *BlueHoc* e *Blueware*

Utilizando os resultados obtidos no Capítulo 3 foi implementado o desvanecimento Nakagami- m nos simuladores *BlueHoc* e *Blueware*. Isto foi feito usando as taxas de erros de pacotes da Seção 3.3 para diferentes distâncias. As taxas de erro de pacote (ou probabilidades de erro) dependem do parâmetro de desvanecimento m e do expoente de perda de percurso η . Os valores das taxas foram colocados nos simuladores para vários valores de m e η em formas de tabelas. Antes de um pacote ser entregue ao seu destino, é gerado um número aleatório entre 0 e 1, usando uma distribuição uniforme. Este número é comparado com a probabilidade de erro daquele tipo de pacote em função da distância entre o nó transmissor e o nó receptor. Se o número gerado é maior que a probabilidade de erro, o pacote é recebido. Caso contrário, um *flag* indicador de erro é ligado e o pacote é descartado. Este procedimento é mostrado na Figura 4.1.

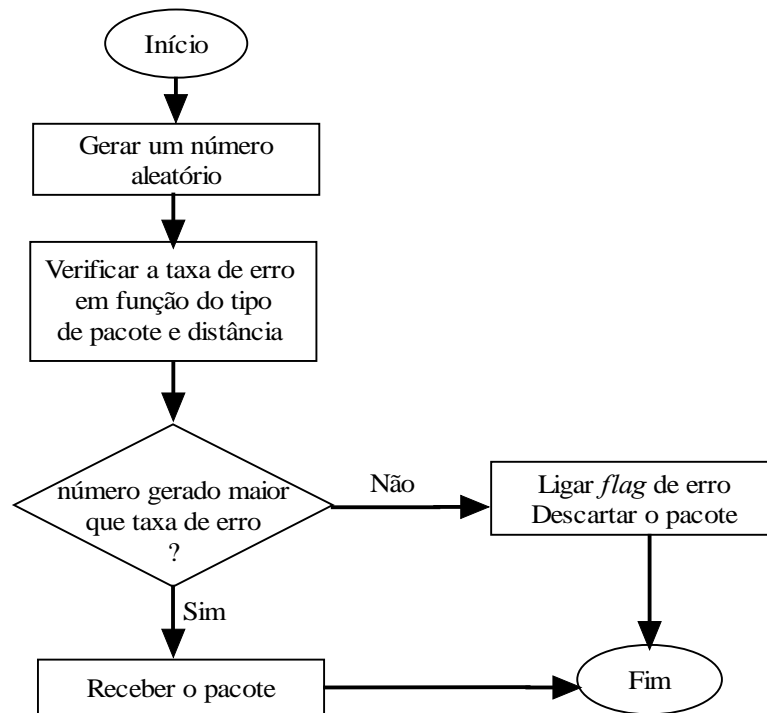


Figura 4.1: Procedimento de entrega de pacote

Outra funcionalidade também foi acrescentada ao *BlueHoc* e *Blueware*. Estas ferramentas suportam apenas dispositivos da classe 3, com alcance de 10 metros. Foi então implementado dispositivos classe 1, que alcançam distâncias maiores. Os dispositivos classe 1 podem ser usados em aplicações que tenham de cobrir grandes distâncias, como em vários cenários de redes de sensores. A desvantagem é que consomem mais energia. A implementação de uma nova classe é relativamente simples, bastando definir mais um parâmetro para cada nó da rede, indicando a classe do dispositivo. Dependendo da classe, o nó usa uma determinada tabela de erro de pacotes. O código acrescentado ao *BlueHoc* e *Blueware* é o mesmo, já que o *Blueware* reutilizou grande parte do código fonte do *BlueHoc*. Assim, podem ser simuladas *piconets* e *scatternets* com diferentes classes de dispositivos.

Como exemplo das novas características do simulador, são apresentados dois cenários de simulação usando o *Blueware*. O primeiro cenário consiste de quatro nós Bluetooth da classe 3 para verificar os efeitos do desvanecimento Nakagami. Esta configuração é mostrada na Figura 4.2, com as distâncias entre os nós e a qualidade do canal (m_1 ou m_2). O segundo cenário tem dispositivos de diferentes classes (Figura 4.4). Em geral, dispositivos mais próximos têm condições de canal melhores ($m=m_1$) que os dispositivos separados por distâncias maiores, que sofrerão um desvanecimento mais severo ($m=m_2$).

As *scatternets* são formadas usando o algoritmo TSF [TAN02a], já implementado no simulador. A topologia da *scatternet* formada para o primeiro e segundo cenário são mostradas nas Figuras 4.3 e 4.5, respectivamente. Na simulação foi usado tráfego CBR de 720 kbps e um expoente de perda de percurso $\eta=3$, um valor típico para ambientes obstruídos em edifícios ou fábricas. Há sempre um único fluxo de dados CBR na rede, usando pacotes DH5. Os resultados da simulação para diferentes valores de m_1 e m_2 são apresentados nas Tabelas 4.1 e 4.2.

O nó que gera o tráfego CBR é o nó fonte e o nó que recebe é o destino, como mostrado nas Tabelas 4.1 e 4.2. Como esperado, os enlaces com desvanecimento menos severo têm um desempenho melhor. Também é interessante notar que sempre que o tráfego passa por um nó ponte a vazão diminui significativamente, pois o nó ponte tem que dividir seu tempo entre duas *piconets*. Os resultados dependem também do esquema de escalonamento da *piconet* e da *scatternet*. O algoritmo usado foi o LCS [TAN02b], em que o nó ponte permanece mais tempo na *piconet* com maior tráfego e menos tempo na *piconet* com menor tráfego.

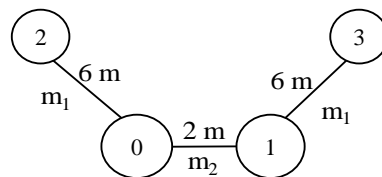


Figura 4.2: Primeiro cenário de simulação com quatro dispositivos

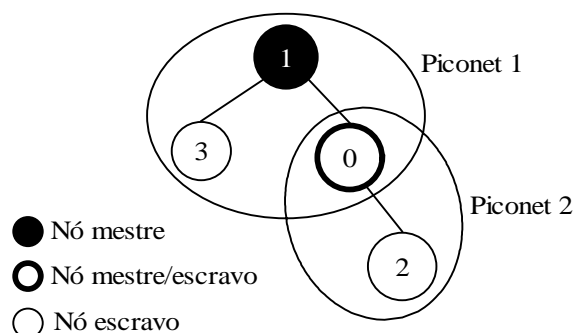


Figura 4.3: Topologia da *scatternet* para o primeiro exemplo

Tabela 4.1: Vazão – Primeiro cenário

Nós	Vazão (kbps)		
	$m_1 = 1$ $m_2 = 0,5$	$m_1 = 1$ $m_2 = 0,75$	$m_1 = 1$ $m_2 = 1$
1-0	333	505	656
3-1	658	658	658
3-0	195	263	335
3-2	192	250	318

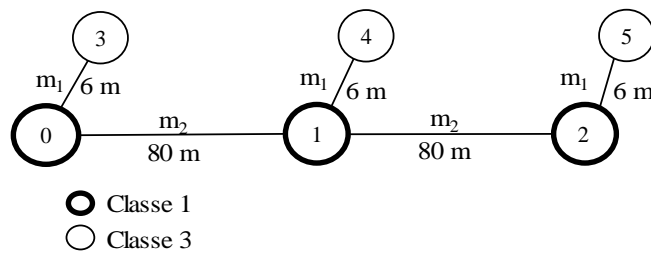


Figura 4.4: Segundo cenário de simulação com dispositivos de diferentes classes

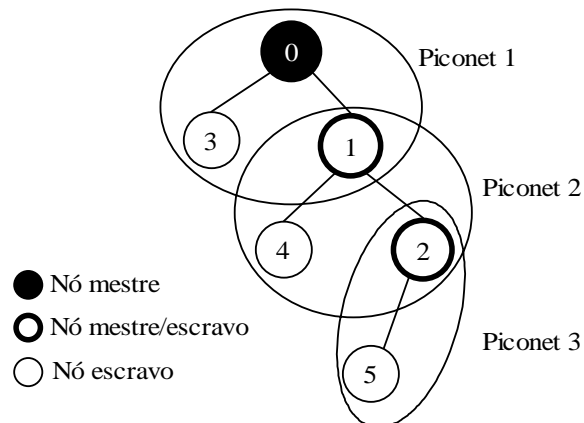
Figura 4.5: Topologia da *scatternet* para o segundo exemplo

Tabela 4.2: Vazão – Segundo cenário

Nós	Vazão (kbps)	
	$m_1 = 1$ $m_2 = 0,5$	$m_1 = 1,5$ $m_2 = 0,75$
3-0	648	668
3-1	197	262
3-4	188	243
3-2	40,9	214
3-5	32,1	209

4.3. Desenvolvimento de Novos Simuladores

Na Seção 4.1 foram apresentadas duas ferramentas de simulação Bluetooth que implementam as principais camadas do protocolo. Qualquer simulação feita terá todos os detalhes de camadas como banda base, LMP e L2CAP. No entanto, nem sempre é necessário tal nível de detalhamento. Para a implementação de um novo algoritmo de formação de *scatternets*, por exemplo, deve-se usar as classes e métodos da camada HCI do simulador.

A tarefa de identificar quais métodos usar, como funcionam e quais seus parâmetros de entrada e saída pode ser bastante trabalhosa. Se o algoritmo de formação criar topologias diferentes de árvore, o esquema de escalonamento de *scatternets* LCS também deve ser modificado, pois só funciona com topologias sem ciclos (como árvores). Os softwares não possuem nenhuma documentação adequada de como implementar novas funções, apenas de como usar as já existentes. Além disso, a falta de uma descrição maior de como as camadas estão implementadas pode levar a suposições errôneas e a resultados incorretos. A incorporação de novos algoritmos de escalonamento, formação ou roteamento deve ser feita levando em conta todos os detalhes das funcionalidades já implementadas.

Por isso, foram desenvolvidos dois novos simuladores em linguagem C++: um para simulação de escalonamento de *piconets* e outro para simulação do processo de conexão (*inquiry* e *page*) de nós Bluetooth. Estes simuladores são usados para avaliar estratégias de escalonamento e de formação de *scatternets* nos próximos capítulos. É importante ressaltar que esta escolha em desenvolver novos ambientes de simulação foi feita por ser mais simples e rápida que a implementação de novos algoritmos nas ferramentas existentes. Novas funcionalidades podem ser incorporadas ao *Blueware*, porém isto demandaria um maior tempo de desenvolvimento.

4.3.1. Simulador de Escalonamento de *Piconets*

Para o simulador de escalonamento de *piconets* foi usada uma modelagem que em [FRE01] é chamada de modelagem por processos. Nela, os sistemas são vistos principalmente do ponto de vista das entidades. As entidades estáticas que compõe o sistema são o mestre e os escravos e as entidades dinâmicas são os pacotes, como mostra a Figura 4.6. O mestre e os escravos (mínimo de um e máximo de sete) possuem os seguintes atributos:

- posição do nó (coordenada x,y);
- fila de pacotes para transmitir;

- fila de pacotes recebidos.

Os pacotes são as entidades dinâmicas porque se movem pelo sistema. Um pacote entra primeiro numa fila de transmissão esperando pelo instante de ser enviado. Depois é entregue a uma fila de pacotes recebidos. Os atributos de um pacote são os seguintes:

- número do pacote (identificador);
- tipo de pacote;
- tempo de chegada na fila de transmissão;
- tempo de chegada na fila de recebidos.

Os pacotes chegam nas filas de transmissão de acordo com um modelo de tráfego, como Poisson, HTTP ou FTP. Como a comunicação na *piconet* é controlada pelo mestre, os pacotes são transmitidos de acordo com determinadas políticas de escalonamento, que serão vistas no Capítulo 5. Com estas entidades (nó mestre, nós escravos e pacotes) pode-se implementar várias estratégias de escalonamento.

Também foi implementado neste simulador o modelo de Nakagami. Os valores das taxas de erro de pacote foram colocados em formas de tabelas para vários valores de m e η , em função da distância entre os nós e tipo de pacote. O processo de entrega de pacotes será o mesmo da Figura 4.1. Se o *flag* que indica erro no pacote estiver ligado, o pacote é descartado. Caso contrário, o pacote não possui erros e é recebido.

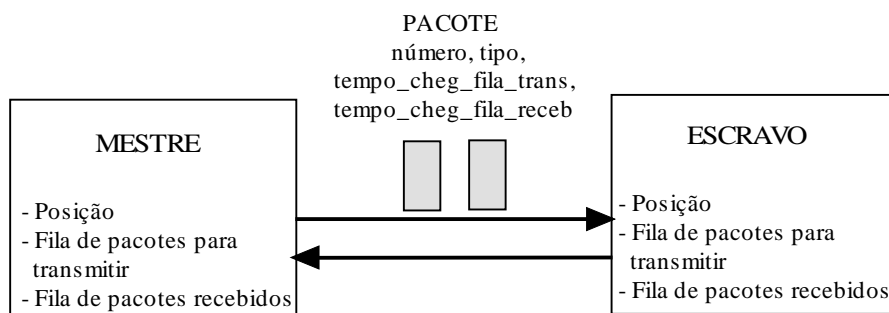


Figura 4.6: Entidades do simulador de escalonamento

4.3.2. Simulador do Processo de Conexão

Para o simulador do processo de conexão foi usada uma modelagem um pouco diferente, que em [FRE01] é chamada de modelagem por eventos. Embora na modelagem por processos também existam eventos (como transmissão de um pacote), estes são condicionais,

ou seja, dependem do tempo de simulação e de outras condições do sistema. Na modelagem por eventos, todos os eventos são incondicionais, isto é, serão executados quando da ocorrência de seu tempo programado no relógio do programa de simulação. Nesta abordagem, o sistema é modelado pela identificação dos eventos, descritos a seguir:

- Evento *Inicia Inquiry*: um nó Bluetooth começa a descobrir nós próximos (*inquiry*);
- Evento *Inicia Inquiry Scan*: o nó começa a escutar processos de *inquiry*;
- Evento *Resposta Inquiry*: o nó responde a um processo de *inquiry*;
- Evento *Fim Inquiry*: nó encerra o processo de descoberta de dispositivos;
- Evento *Inicia Page*: um nó Bluetooth começa o processo de conexão (*page*);
- Evento *Recebe Page*: o nó recebe uma requisição de conexão (*page*);

Além dos eventos, também são definidas as entidades, que são os nós Bluetooth. Os nós têm dois atributos: posição e estado do nó. O estado pode ser um dos seguintes: *standby* (desconectado), conectado, *inquiry*, *inquiry scan*, *page*, *page scan*, *master response*, *slave response* e *inquiry response*. Estes estados são os definidos pela especificação Bluetooth. As transições possíveis entre os estados são mostradas na Figura 4.7. As ações relacionadas aos eventos devem ser definidas num algoritmo de formação de *scatternets*, explorados no Capítulo 6. As transições de estado ocorrem baseadas nestas ações.

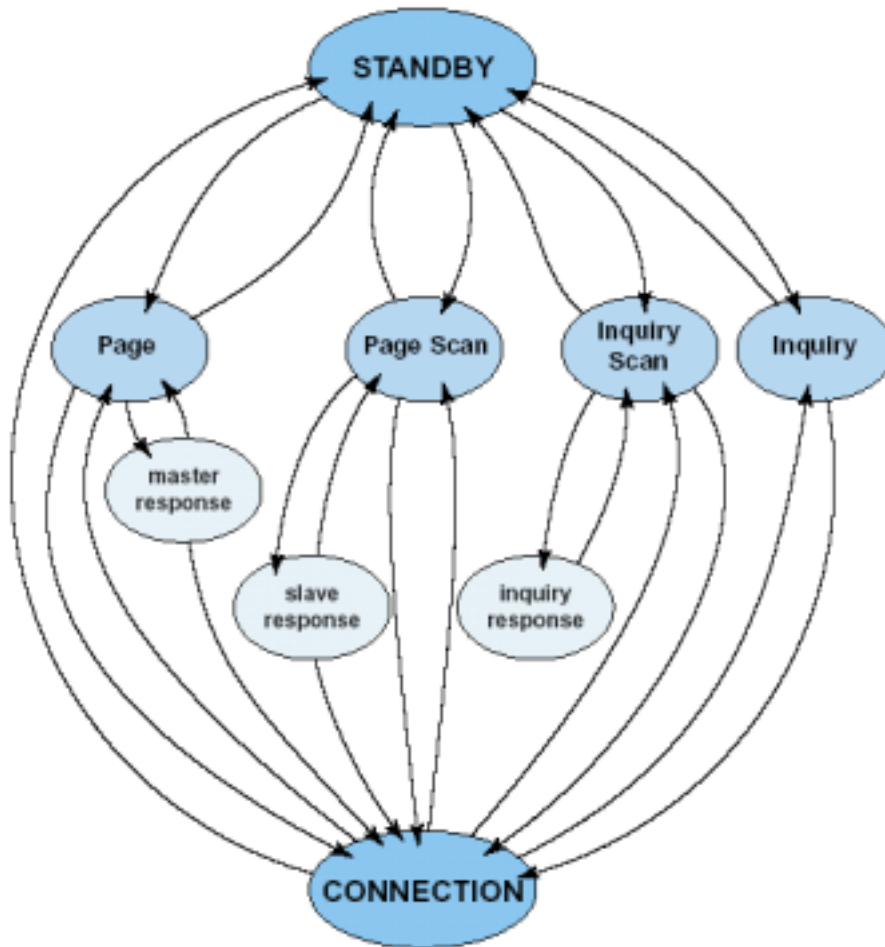


Figura 4.7: Diagrama de estados do Bluetooth [BLU01a]

4.4. Conclusão

A simulação é uma técnica amplamente utilizada em redes de computadores e telecomunicações e pode ser muito útil para redes Bluetooth. Neste capítulo foram apresentados os simuladores existentes, o desenvolvimento de novas ferramentas e resultados de vazão em *scatternets* com desvanecimento Nakagami. A qualidade do canal pode afetar significativamente o desempenho de redes Bluetooth. Quanto menor o valor de m , pior o desempenho. No próximo capítulo será abordado o problema do escalonamento de *piconets* e proposta uma técnica que utiliza o parâmetro m de desvanecimento para indicar a qualidade do canal e melhorar o desempenho da *piconet*.

Capítulo 5

Escalonamento de *Piconets* Bluetooth

Numa *piconet* o acesso ao canal é controlado pelo dispositivo mestre. Um dispositivo escravo pode enviar um pacote apenas se receber um pacote do mestre. O mestre transmite pacotes ao escravo em *time-slots* pares e recebe pacotes em *time-slots* ímpares. Portanto, como o Bluetooth é um padrão TDD controlado pelo mestre, os algoritmos de escalonamento devem minimizar o desperdício de quadros de tempo que ocorre quando apenas o mestre ou o escravo tiver dados para transmitir.

O compartilhamento da mesma banda de frequências entre os dispositivos IEEE 802.11b e os dispositivos Bluetooth causa interferência se eles estiverem operando próximos. Adicionalmente, a proximidade de operação de diferentes *piconets* também pode gerar interferência mútua. Em redes Bluetooth com mobilidade dos nós, como em potenciais aplicações de sensoriamento, os efeitos de desvanecimento do sinal de rádio podem degradar significativamente o desempenho dos enlaces. A utilização de estratégias que minimizem a transmissão em canais com grande interferência ou desvanecimento severo do sinal pode melhorar substancialmente o desempenho da *piconet*.

Vários esquemas são propostos na literatura para escalonamento de *piconets* e são apresentados neste capítulo. Um novo algoritmo de escalonamento que usa a informação de qualidade de canal na política de escalonamento é proposto. A qualidade do canal é baseada numa estimativa símbolo-a-símbolo da relação sinal-interferência recebida e do parâmetro m de Nakagami. São apresentados resultados de simulação que demonstram o desempenho do algoritmo para diferentes condições de tráfego, usando o simulador descrito na Seção 4.3.

5.1. Trabalhos Relacionados sobre Escalonamento

Como na especificação do Bluetooth não foi descrito como deve ser feito o escalonamento, várias estratégias foram propostas para escalonamento de *piconets* e *scatternets*. O escalonamento de *piconets*, ou *intrapiconet*, consiste de como o mestre fará a varredura (*polling*) dos escravos dentro da sua *piconet*, que determina como a capacidade da largura de banda será distribuída entre as unidades escravo. Numa *scatternet*, um novo problema de escalonamento surge, pois um nó ponte que participa de duas ou mais *piconets* só pode estar ativo em uma de cada vez. O escalonamento de *scatternets*, ou *interpiconet*, deve agendar a presença do nó ponte nas diferentes *piconets* em que participa para que o tráfego possa fluir da maneira mais eficiente possível. As estratégias de escalonamento *intrapiconet* e *interpiconet* podem ser combinadas para atingir uma melhor eficiência [JOH01a].

Em [CAP01], várias estratégias de escalonamento *intrapiconet* são comparadas. No esquema *round robin* (RR) é definida uma ordem cíclica fixa, com duração de um quadro de tempo. A técnica *round robin* exaustivo (ERR) também usa uma ordem fixa, mas o mestre não muda para próximo escravo até que as filas de ambos estejam vazias. A principal desvantagem do ERR é que o canal pode ser capturado pelos nós que geram um tráfego maior que a capacidade do sistema. Este efeito de captura do ERR pode ser contornado utilizando o esquema *round robin* limitado (LRR), que limita o número de transmissões.

Uma nova estratégia denominada *round robin* limitado e ponderado (LWRR), com pesos sendo atribuídos dinamicamente de acordo com o tamanho das filas, também é apresentada em [CAP01]. Outros trabalhos sobre escalonamento de *piconets* consideram qualidade de serviço (QoS) em redes Bluetooth, como [LIU02] e [LAP03]. Os resultados apresentados em [CAP01], [LIU02] e [LAP03] não consideram nenhum modelo de perda de pacotes ou qualidade para o canal de rádio.

Em [KAL99] é apresentada uma política de escalonamento baseada nas filas de transmissão do mestre e dos escravos. Os pares mestre-escravo são classificados em classes de acordo com o tamanho das filas e do desperdício de quadros. Esta informação é usada numa política denominada HOL-KFP (*Head-of-the-Line K-Fairness Policy*). Quando os autores introduziram erros no canal, o HOL-KFP teve seu desempenho reduzido. Uma extensão do HOL-KFP chamada de WAKFP (*Wireless Adapted KFP*) foi proposta em [KAL99] para melhorar o desempenho na presença de erros de canal. A estratégia utilizada no

WAKFP é baseada apenas na informação de existência ou não de comunicação entre o mestre e o escravo. O modelo de erro utilizado para o canal é uma cadeia de Markov com dois estados.

Em geral as estratégias de escalonamento não consideram um modelo de perda para o canal sem fio. Em [GOL02] um algoritmo chamado *BIAS* (*Bluetooth Interference Aware Scheduling*) usa um procedimento de estimação de canal para detectar a presença de outros dispositivos sem fio que operam na mesma banda (como outros dispositivos Bluetooth ou IEEE 802.11b). O algoritmo de escalonamento evita a transmissão em frequências que têm uma alta taxa de erro de bit, chamadas de frequências ruins. Isto reduz a perda de pacotes devido à interferência de outros dispositivos próximos. O algoritmo BIAS proposto em [GOL02] detecta canais problemáticos monitorando a taxa de erro de bit.

O escalonamento *interpiconet* também é abordado por vários autores. O ideal de um esquema *interpiconet* é que o nó ponte esteja presente na *piconet* no momento em que o mestre o escalonar e ainda que o mestre retire o nó ponte do processo de varredura quando este estiver ausente. Os algoritmos de escalonamento de *scatternets* podem ser divididos em mecanismos com decisão isolada e com decisão distribuída. Em mecanismos de decisão isolada o nó ponte é o único responsável em decidir sobre sua presença nas *piconets*, podendo ou não comunicar essa decisão aos mestres das *piconets* as quais pertence. Em mecanismos com decisão distribuída o nó ponte faz um acordo com os mestres das *piconets* que participa para decidir quando vai estar presente numa *piconet*. Estes acordos podem representar ganhos na vazão das *scatternets*, porém requerem novas mensagens de sinalização.

Em [JOH01a] é apresentado o conceito de ponto de encontro e janela de encontro. O ponto de encontro é o *slot* em que o nó ponte e o mestre decidem se comunicar, sendo a duração deste encontro chamada de janela de encontro. Em [JOH02] é apresentado um mecanismo de decisão distribuída que usa estes conceitos. O mestre deve manter uma lista com os pontos de encontro de seus nós ponte e uma outra lista dos pontos de encontro de seus nós ponte com as outras *piconets*.

Entre outras propostas com decisão distribuída, estão o LCS [TAN02b] e o escalonamento baseado num novo modo de funcionamento, o modo JUMP [JOH01b]. Uma estação em modo JUMP estaria por padrão ausente da rede, não sendo necessário ao nó ponte comunicar a sua saída da *piconet*. O LCS é o esquema implementado no simulador *Blueware*, sendo restrito a topologias sem ciclos, como árvores.

Os principais algoritmos de decisão isolada são o *Pseudo-Random Coordinated Scheduling* – PCSS [RAC01], *Load Adaptive Algorithm* – LAA [HAR02] e *Adaptive Interpiconet Scheduling Algorithm* – AISA [PRI30]. No PCSS cada nó ponte escolhe um ponto de encontro pseudo-aleatoriamente, baseado no relógio do mestre e no endereço do escravo. No LAA, o nó ponte monitora suas filas de saída e recebe informação das filas dos mestres para se adaptar a variações no tráfego. O AISA é um algoritmo adaptativo e parametrizável, permitindo a escolha da métrica que se deseja otimizar (vazão agregada do tráfego, retardo dos pacotes ou consumo de energia dos nós ponte).

Como visto nesta seção, poucos algoritmos de escalonamento *intrapiconet* consideram erros no canal de rádio, mesmo assim bastante limitados. Entre os esquemas *interpiconet* nenhum deles considera a qualidade do canal. Neste contexto é proposta uma estratégia de escalonamento *intrapiconet* baseada na estimação de qualidade do canal usando o modelo de desvanecimento de Nakagami.

5.2. Informação de Estado de Canal

Para obter a informação de estado de canal, foram considerados dois parâmetros que precisam ser estimados: a relação sinal-interferência (SIR) recebida e o parâmetro m da distribuição de Nakagami. O modelo de interferência é aditivo e não pode ser dissociado do ruído. Portanto ruído e interferência são modelados conjuntamente como ruído Gaussiano branco aditivo (AWGN). Esta relação também é expressa como relação sinal-ruído (SNR). Em [RAM01], é proposto um estimador *on-line* de SNR para canais com desvanecimento que não requer a transmissão de símbolos de treinamento. É baseado na observação de um bloco de símbolos demodulados, r_n . A relação sinal-ruído média estimada,

$$\bar{\gamma} = \frac{E_s}{N_0} E[\alpha_n^2], \quad (5.1)$$

é uma função da energia de símbolo, E_s , da densidade espectral de potência unilateral do ruído, N_0 , e do desvanecimento aleatório α_n experimentado pelo n -ésimo símbolo recebido, r_n . O estimador proposto em [RAM01] define um parâmetro z como sendo a razão de duas medidas estatísticas calculadas a partir da seqüência de símbolos observados, r_n ,

$$z = \frac{E[r_n^2]}{E[|r_n|^2]^2} \quad (5.2)$$

Através de um ajuste polinomial de curvas é obtida uma relação aproximada entre z e $\bar{\gamma}$ para $m=1$, $z = f(\gamma)$. Equações para estimar SNR para outros valores de m podem ser derivadas de forma análoga. A derivação de $f(\cdot)$ também depende do esquema de modulação empregado. O algoritmo proposto em [RAM01] apresenta uma boa convergência para os valores estimados de SNR, a partir da análise de uma seqüência de 1000 bits. A aplicação desta técnica para a estimação de SNR pode ser feita em enlaces Bluetooth, pois demanda baixo custo computacional. O valor estimado de SNR é refinado de maneira contínua em função dos pacotes recebidos, permitindo uma estimativa símbolo-a-símbolo.

A função densidade de probabilidade de Nakagami é uma distribuição de dois parâmetros (m, Ω) , onde $\Omega = E[r_n^2]$ é o segundo momento das amostras do sinal, r_n . O parâmetro de desvanecimento m é definido como

$$m = \frac{\Omega^2}{E[(r_n^2 - \Omega)^2]} \quad m \geq 0,5 \quad (5.3)$$

Pode-se estimar conjuntamente SNR e o parâmetro m . Muitos estimadores têm sido propostos na literatura para o parâmetro m . Em [CHE02], Cheng e Beaulieu propuseram uma nova família de estimadores baseados em momento inteiros e não-inteiros. Estes estimadores são eficientes para um número moderado de amostras e são adequados para implementações de baixa complexidade. Em [ZHA02] diferentes métodos para estimação de m são comparados, sendo que uma técnica clássica proposta por Greenwood e Durand foi considerada a melhor. Qualquer destas técnicas pode ser implementada num receptor Bluetooth sem grande aumento da complexidade computacional. As estimativas de m e SNR representam a informação de estado de canal, que é utilizada na estratégia de escalonamento proposta.

5.3. Algoritmo de Escalonamento *Intrapiconet* usando Estimação de Estado de Canal

A condição do canal de comunicação sem fio pode afetar o desempenho da *piconet* e da estratégia de *polling*. Em ambientes móveis, a qualidade do canal pode mudar muito rapidamente e isto significa que um desempenho melhor pode ser atingido se um nó é requisitado a transmitir no momento que tem uma boa condição de canal e não permitido de

transmitir quando as condições são ruins. Como Bluetooth é uma tecnologia desenvolvida para redes pessoais sem fio, erros de canal devido à mobilidade e interferência de outros dispositivos são muito comuns. Um bom algoritmo de escalonamento deve considerar estes dois fatores.

Nesta seção é proposto um algoritmo chamado *BCSS (Bluetooth Channel State Scheduling)* que usa a informação de estado de canal para o escalonamento de *piconets*. Os valores do parâmetro de desvanecimento m e da relação sinal-ruído podem ser facilmente estimados como descrito na Seção 5.2. Estes parâmetros indicam uma estimativa da qualidade do canal, levando em conta interferência, desvanecimento e distância entre os nós.

O mestre faz as estimações usando os pacotes de dados trocados com os escravos. Cada vez que o mestre recebe um pacote, os valores de m e SNR para aquele enlace são refinados. Como esta tarefa não requer que nenhuma informação adicional seja trocada entre o mestre e os escravos, nenhum tempo é adicionado à política de escalonamento. A precisão da estimação de m e da SNR será baixa no começo da transmissão, quando poucos pacotes foram trocados e a quantidade de bits a serem analisados não é significativa. Depois da transmissão de alguns pacotes, os valores estimados convergem para valores próximos aos reais.

Na nova política de escalonamento, o mestre irá fazer o *polling* somente dos escravos que estão acima de um certo limiar para m e SNR, indicando um bom estado de canal. Um escravo poderá transmitir apenas um pacote cada vez que é requisitado. Os escravos que estão abaixo do limiar, indicando uma baixa qualidade de canal, serão pulados por no máximo t_j vezes. O mestre deve armazenar o número t de vezes que cada escravo já foi pulado. Cada vez que t atingir o valor de t_j , t deve voltar a zero. Se a condição do canal for sempre boa (ou sempre ruim) para todos os enlaces o algoritmo funciona como uma estratégia *round robin* convencional. O procedimento do algoritmo é mostrado na Figura 5.1.

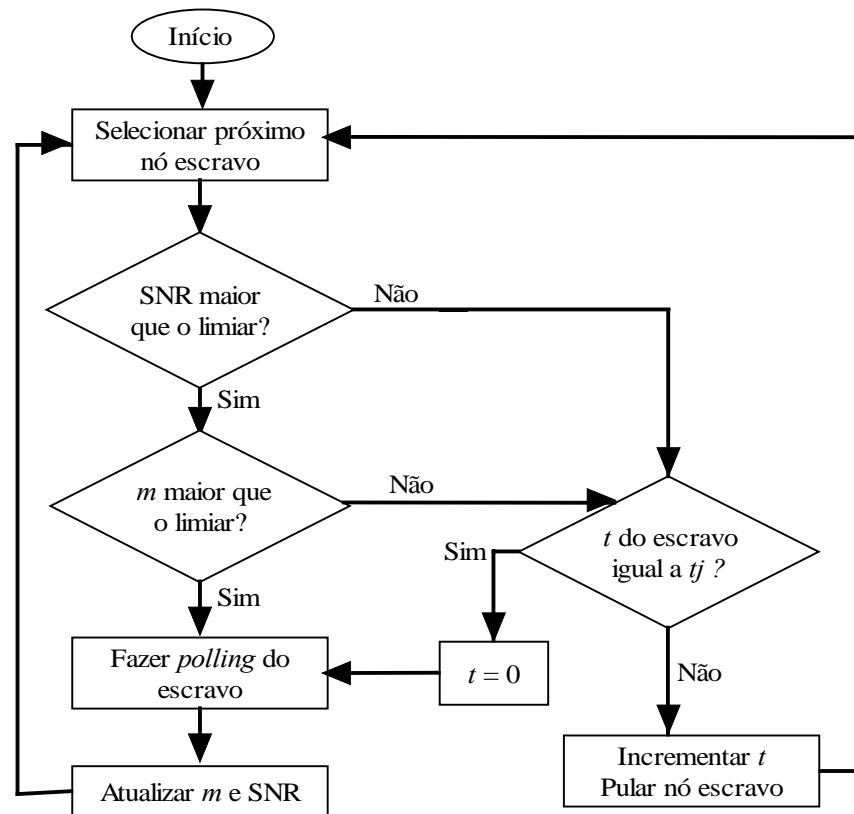


Figura 5.1: Algoritmo BCSS

5.4. Resultados de Simulação

Para verificar o desempenho do algoritmo BCSS foi usado o simulador de escalonamento descrito na Seção 4.3. Foram implementadas, além do BCSS, as estratégias *round robin* (RR) e *round robin* exaustivo (ERR) para comparar o desempenho entre elas. O efeito do desvanecimento Nakagami é simulado usando os resultados obtidos no Capítulo 2.

5.4.1. Tráfego de Poisson

Uma fonte de informação tipo Poisson foi considerada para geração de tráfego em cada nó da *piconet*. Este modelo pode simular várias aplicações do Bluetooth. A distribuição de Poisson é dada por:

$$P(n) = \frac{(\lambda t)^n}{n!} \exp(-\lambda t), \quad (5.4)$$

onde λ é a taxa média de chegada de pacotes por *time slot* e $P(n)$ é a probabilidade de n chegadas num tempo t .

No primeiro cenário de simulação foi investigada a influência do parâmetro m no escalonamento *round robin*. Como mostra a Figura 5.2, o cenário consiste de uma *piconet* com um mestre e sete escravos separados por uma distância d . A Figura 5.3 mostra as curvas do atraso médio de pacotes DM1 para três diferentes valores de m , considerando distâncias de sete e dez metros entre os nós com expoente de perda de percurso $\eta = 4$. Neste cenário todos os nós têm as mesmas condições de tráfego. A distribuição de Poisson é usada para gerar tráfego nas 14 filas (do mestre para os sete escravos e dos escravos para o mestre).

Pode-se observar na Figura 5.3 que o estado do canal tem grande influência no atraso médio da *piconet*, afetando o desempenho da rede. A distância entre os nós está embutida na estimativa de SNR, devido a perda de percurso do sinal transmitido. A condição de propagação do canal é dada pelo parâmetro m . A variabilidade de m reflete os efeitos de desvanecimento rápido do sinal devido à mobilidade dos nós ou de outros objetos no ambiente da rede, que afetam a propagação do sinal.

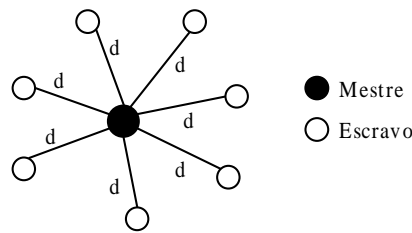


Figura 5.2: Primeiro cenário de *piconet*

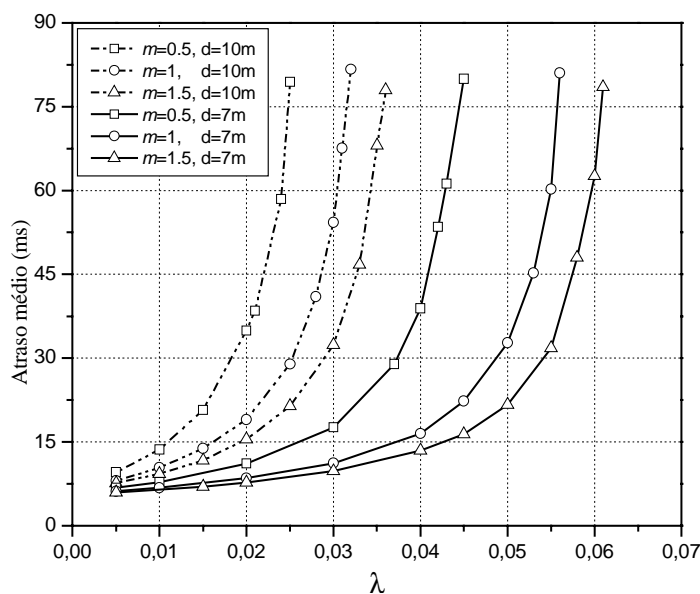


Figura 5.3: Atraso médio para diferentes valores de m usando *round robin*

No segundo cenário de simulação mostrado na Figura 5.4, uma *piconet* com um mestre e quatro escravos é considerada. As Figuras 5.5 e 5.6 comparam o atraso médio de pacotes DM1 para diferentes condições de tráfego, usando RR, ERR e o algoritmo BCSS. No algoritmo BCSS foi usado $t_j=6$, um limiar $m=1$ e limiar SNR=10 dB. Isto significa que o mestre fará o *polling* apenas dos escravos com $m > 1$ e SNR > 10 dB, sendo que os outros serão pulados por no máximo seis vezes. O tráfego é o mesmo nas filas do mestre e dos escravos. Na simulação considera-se que as condições do canal mudam a cada duas rodadas do esquema de *polling*.

Também foi considerado o cenário da Figura 5.4 com $d=10$ m para diferentes condições de tráfego nos enlaces mestre-escravo, como mostra a Tabela 5.1. Os resultados são apresentados na Figura 5.7. Das Figuras 5.5 a 5.7 pode-se observar que o algoritmo BCSS melhora seu desempenho quando o tráfego é alto. Para baixo tráfego, a estratégia ERR tem o melhor desempenho. Em [CAP01] e [JOH99] também foi concluído que o serviço exaustivo (ERR) não tem bom rendimento com tráfegos altos.

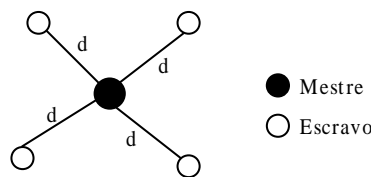


Figura 5.4: Segundo cenário de *piconet*

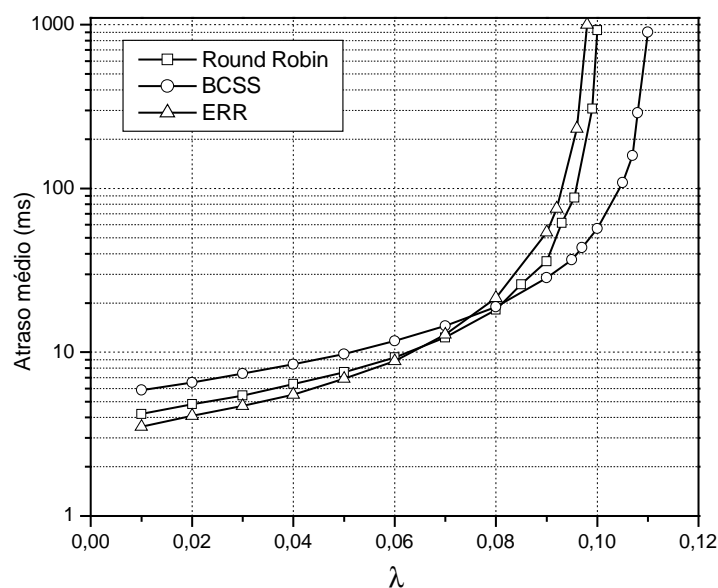


Figura 5.5: Atraso médio para $d = 7m$

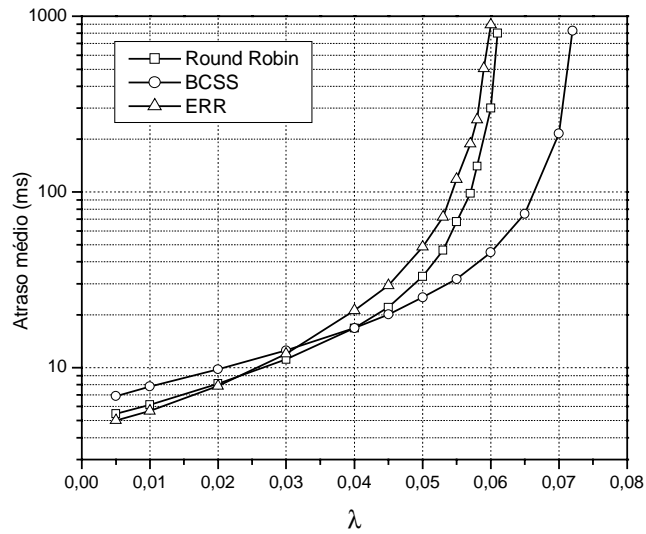


Figura 5.6: Atraso médio para $d = 10m$

Tabela 5.1: Condições de tráfego para enlaces mestre-escravo

Cenário de Tráfego	λ_1 (mestre-escravo 1)	λ_2 (mestre-escravo 2)	λ_3 (mestre-escravo 3)	λ_4 (mestre-escravo 4)
1	0,01	0,01	0,02	0,02
2	0,01	0,01	0,04	0,04
3	0,03	0,05	0,03	0,05
4	0,03	0,03	0,06	0,06

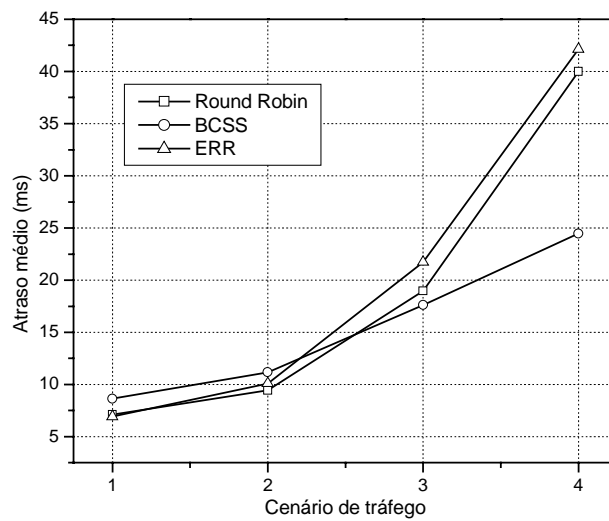


Figura 5.7. Atraso médio para diferentes cenários de tráfego

5.4.2. Tráfego HTTP

Também foi considerado tráfego HTTP para avaliar as estratégias de escalonamento. O tráfego para uma sessão HTTP foi modelado de acordo com a Tabela 5.2, baseado nos parâmetros de [STU02]. Este é um modelo apenas para geração da chegada dos pacotes nas filas de transmissão. Para um tráfego HTTP mais realista, além da geração de tráfego, também o protocolo TCP/IP deveria ser considerado no simulador. Portanto, este é um modelo bastante simplificado e é usado apenas para validar o algoritmo BCSS em fontes de tráfego diferentes de Poisson.

Neste cenário o mestre da *piconet* atua como um *gateway*, sendo um servidor responsável em dar acesso à Internet para os dispositivos escravos (Figura 5.8). O *gateway* pode ser um microcomputador conectado à Internet que oferece o serviço para dispositivos móveis Bluetooth, como computadores portáteis e PDAs, como descrito em [ROU02]. O tráfego HTTP será na direção do mestre para os escravos. Na direção dos escravos para o mestre são feitas as requisições ao servidor.

O gráfico da Figura 5.9 mostra o tempo médio para carregar uma página WWW em função do tamanho da página para uma *piconet* com o *gateway* e 4 escravos separados do mestre por uma distância de seis metros. A Figura 5.10 mostra o tempo médio para carregar uma página para uma *piconet* com sete dispositivos escravo também separados por seis metros do *gateway*. Os parâmetros do algoritmo BCSS (limiar de m e SNR) foram os mesmos usados no tráfego de Poisson. O pacote utilizado foi o DH5, que atinge a máxima taxa de transmissão entre os pacotes assíncronos. Na simulação foi considerado que as condições do canal mudam a cada duas rodadas do esquema de *polling*.

O algoritmo BCSS teve um melhor desempenho quando o tráfego é intenso, ou seja, o tamanho das páginas é maior. Este melhor desempenho para tráfegos altos e pior desempenho para tráfegos baixos também foi verificado no modelo de Poisson. Isto se deve ao fato que não adianta pular um escravo que está com uma qualidade de canal ruim se os enlaces dos outros escravos não têm dados para transmitir. A estratégia de pular os escravos com canais em más condições só é adequada se houver fluxo de dados em outro enlace. O número de dispositivos escravos piora significativamente o desempenho da rede, já que a largura de banda (1 Mbps) é a mesma e deve ser dividida entre todos.

Tabela 5.2: Modelo de tráfego HTTP

Parâmetro	Distribuição		Média
Intervalo entre páginas (s)	Exponencial negativa	$f(x) = \lambda e^{-\lambda x}$	12
Objetos por página	Geométrica	$p(n) = p(1-p)^{n-1}$	Variável
Tamanho de objetos (bytes)	Gama	$f(x) = \frac{\lambda e^{-\lambda x} (\lambda x)^{n-1}}{(n-1)!}$	3700

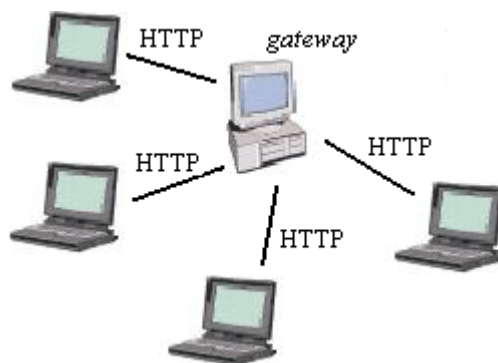


Figura 5.8: Piconet com um gateway Bluetooth

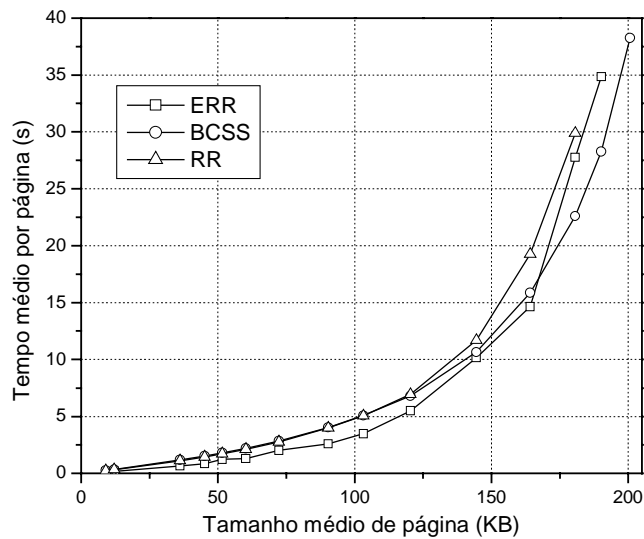


Figura 5.9: Tempo médio por página com 4 escravos

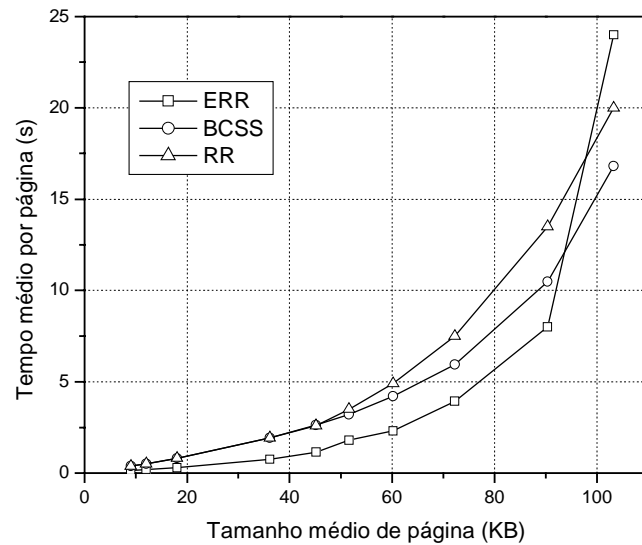


Figura 5.10: Tempo médio por página com 7 escravos

5.5. Conclusão

Neste capítulo foi apresentado o problema do escalonamento de *piconets* e *scatternets* Bluetooth. Foi proposto o algoritmo de escalonamento *intrapiconet* BCSS, usando a relação sinal-ruído e o parâmetro m de Nakagami para estimação da qualidade do canal. Resultados de simulação demonstraram que o algoritmo BCSS é eficiente para altas taxas de tráfego se as condições de canal variam rapidamente. Estas variações na qualidade do canal estão presentes em muitas aplicações do Bluetooth em ambientes com interferência e mobilidade. A estratégia BCSS pode ser combinada com outras políticas de escalonamento para melhorar seu desempenho. No próximo capítulo serão abordados os problemas de formação e roteamento em *scatternets*, tendo como foco o uso de Bluetooth em redes de sensores.

Capítulo 6

Redes de Sensores usando Bluetooth

Bluetooth é uma tecnologia promissora não apenas para redes pessoais, mas também para outras aplicações de redes *ad hoc*. As redes de sensores sem fio são aplicações recentes de redes *ad hoc* e que devem ter um grande desenvolvimento nos próximos anos. Os protocolos e algoritmos de redes de sensores devem ter capacidades de auto-organização, como vistos na Seção 2.1.3. O desenvolvimento de novos protocolos MAC para redes de sensores e a utilização de protocolos existentes é um tópico de grande importância. O protocolo MAC é responsável pela criação da infra-estrutura da rede e pelo compartilhamento dos recursos de comunicação entre os nós. Algumas tentativas de usar Bluetooth como protocolo de acesso ao meio foram feitas em [KAS01], [FER03] e [LEO03].

Ainda que o Bluetooth não tenha sido desenvolvido especificamente para redes de sensores, algumas de suas características o tornam uma excelente alternativa. O processo de descoberta de dispositivos, modos de economia de energia e baixo custo são algumas vantagens sobre outras tecnologias. No entanto, a questão de formação de *scatternets* e roteamento nestas redes continua em aberto e precisa ser tratada para que redes de sensores usando Bluetooth sejam implementadas. Neste capítulo serão discutidos os algoritmos de formação e roteamento existentes e apresentada uma nova estratégia de redes Bluetooth que pode ser usado em algumas aplicações de redes de sensores.

6.1. Formação e roteamento de *scatternets*

Em [KAS01] e [FER03] foi implementada uma rede de sensores que consiste de apenas uma *piconet*. Em [FER03] os nós sensores são os escravos da *piconet* e um computador é o mestre, que funciona como um equipamento de armazenamento dos dados.

Esta abordagem tem a limitação de oito nós (um mestre e sete escravos). Para montar uma rede de sensores com mais nós, um algoritmo de formação de *scatternets* precisa ser utilizado. Existem muitas propostas recentes na literatura para formação e roteamento de *scatternets*.

Algumas das primeiras soluções propostas requerem que todos os nós estejam ao alcance uns dos outros, como o BTCP (*Bluetooth Topology Construction Protocol*) [SAL01], TSF [TAN02a] e BSFA (*Bluetooth Scatternet Formation Algorithm*) [LAW01]. O BTCP é baseado num processo de eleição de um nó líder para coletar informação da topologia. O líder irá determinar os papéis dos nós da rede (mestres, escravos e pontes). O número máximo de nós é de 36. Os protocolos TSF e BSFA formam *scatternets* em topologias de árvores, sem limitação do número de nós. O TSF suporta ambientes dinâmicos, onde os nós podem entrar ou sair da rede a qualquer momento. Nenhum destes três esquemas considera a questão do roteamento, ainda que nas *scatternets* em forma de árvore esta tarefa não seja difícil, por existir um único caminho entre dois pontos.

Entre as soluções mais gerais, onde os nós não precisam estar todos ao alcance uns dos outros, pode-se citar o *BlueTree* [ZAR01], *BlueNet* [WAN02], *BlueStars* [PET03] e *BlueRings* [LIN03]. O *BlueTree* requer que o protocolo seja iniciado por um nó (raiz) que gera uma *scatternet* em forma de árvore. O nó raiz começa o processo de formação, se conectando com seus vizinhos. Estes, por sua vez, tentam se conectar com outros vizinhos e assim por diante. Em [LEO03] uma rede de sensores baseada em Bluetooth foi implementada usando o algoritmo *BlueTree*.

O *BlueNet* gera *scatternets* cujas *piconets* tem um número balanceado de escravos, sem uma topologia específica. O *BlueStars* consiste de três fases: descobrimento dos dispositivos vizinhos, formação de *piconets* (estrelas) e seleção de dispositivos ponte entre as *piconets* para formar a *scatternet* (constelação). A topologia resultante contém múltiplos caminhos entre qualquer par de nós. O *BlueRings* forma *scatternets* com a topologia em anel num processo semelhante ao BTCP. O nó escolhido como líder é o responsável em dar informações aos outros nós para que uma *scatternet* em anel seja formada. O *BlueRings* considera tolerância à falhas, isto é, nós podem entrar e sair da rede. A principal vantagem da topologia anel é o roteamento, que também é detalhado como parte do algoritmo *BlueRings*. Seja qual for o protocolo de formação utilizado, um algoritmo de escalonamento *interpiconet* precisa ser usado para o nó ponte. O único que prevê uma política específica de escalonamento para a topologia formada é o *BlueRings*.

Dentre todos estes protocolos citados, apenas o *BlueTree* e o *BlueRings* apresentam soluções para o roteamento. No *BlueTree*, o roteamento é feito através de tabelas de roteamento armazenadas em cada nó da rede. No *BlueRing* são apresentados algoritmos que devem ser executados nos nós em caso de transmissão ou recepção de pacotes, sem a necessidade de tabelas. Esta tarefa é facilitada pela topologia em anel. Em [SUN02] são propostas *scatternets* auto-roteáveis com topologia em árvore. Ao contrário do *BlueTree*, o processo de formação da árvore já prevê a descoberta de rotas. Cada nó deve armazenar apenas uma pequena informação de tamanho fixo, sem a necessidade de manter grandes tabelas de roteamento. O algoritmo GSFA (*Group-Scatternet Formation Algorithm*) proposto em [SHI03], forma vários grupos de *scatternets* que juntos formam uma *scatternet* maior. O roteamento na rede resultante é facilitado pela existência de vários grupos de *scatternets*.

Outras soluções para o roteamento em *scatternets* são independentes de topologia. O método de roteamento por vetores [BHA99] faz a descoberta de rotas por demanda, isto é, apenas quando um nó precisa se comunicar com outro. A rota completa é então carregada em cada pacote, o que pode levar a uma sobrecarga na *scatternet*. Outro protocolo, também por demanda de rota, é apresentado em [PRA02]. Um nó seleciona a melhor rota tendo em vista a economia de energia dos dispositivos. Uma das melhores soluções é usar o protocolo de redes *ad hoc* ZRP (*Zone Routing Protocol*) adaptado às características do Bluetooth [KAP03]. O ZRP combina as características dos protocolos por demanda (reativos) e por tabela (pró-ativos). No entanto, esta versão do ZRP para Bluetooth não suporta mobilidade na *scatternet*.

Nenhuma das soluções de formação e roteamento Bluetooth consideram cenários de redes de sensores, que possuem algumas diferenças de redes *ad hoc* tradicionais, citados na Seção 2.1.3. A principal desvantagem destes algoritmos é que os nós permanecem conectados todo o tempo, mesmo que não tenham dados para enviar. Isto pode desperdiçar energia nos nós e não é adequado para redes de sensores. Além do mais, a entrada e saída de nós na *scatternet* é uma tarefa complexa, especialmente em casos de mobilidade que geram mudanças freqüentes na topologia da *scatternet*. Em casos de muitos nós móveis, a sobrecarga de informação na rede usada para manter a *scatternet* conectada pode ser bastante grande.

Uma abordagem diferente foi proposta em [LIU03], chamada *scatternet-route*. As *scatternets* são formadas apenas por demanda de tráfego, sem necessidade de manter enlaces conectados desnecessariamente. A formação da *scatternet* é similar aos protocolos de roteamento reativos. A *scatternet* só é mantida enquanto há fluxo de tráfego, sendo totalmente

desconectada após o fim da troca de dados. Esta abordagem é mais adequada para dispositivos com limite de energia e para aplicações com alto volume de tráfego, como dados em tempo real. Para redes de sensores, a economia de energia do algoritmo é uma característica desejável. Porém em redes de sensores o tráfego é normalmente baixo [LOU03], [AKY02] e formar uma *scatternet* para a troca de alguns poucos pacotes pode não ser a melhor solução. Outra desvantagem é que quando um nó quer enviar dados, a rota tem que ser descoberta e a *scatternet* formada. A estrutura especial desta *scatternet-route* requer um novo algoritmo de escalonamento, também descrito em [LIU03].

6.2. Uma Estrutura de Redes Bluetooth para Aplicações de Sensores

As redes de sensores possuem inúmeras aplicações, sendo que um algoritmo de formação deve levar em conta as especificidades de cada tipo de aplicação. Para um protocolo de formação de redes de sensores foram consideradas aplicações onde há um nó, chamado nó de gerência, responsável em coletar ou enviar dados para os sensores. O nó de gerência inicia o processo de descoberta de nós e formação da rede. O objetivo é dar ao nó de gerência uma visibilidade geral da rede para que possa construir uma tabela de roteamento. Depois de um período para descobrir os nós e rotas, os nós sensores não precisam permanecer conectados numa *scatternet*, economizando energia.

Quando o nó de gerência quer enviar ou coletar informações de um sensor, ele já sabe a rota e pode enviar os dados sem um tempo extra de descoberta da rota, diferente da *scatternet route*. Na realidade, a *scatternet* nunca será formada! É formada apenas uma *piconet* temporária entre dois dispositivos. Um nó se conecta com outro, envia os dados e se desconecta novamente. Isto elimina também a necessidade de um algoritmo de escalonamento, tanto *intrapiconet* como *interpiconet*. Esta solução é detalhada a seguir.

O estabelecimento de um enlace entre dispositivos Bluetooth requer dois procedimentos: *inquiry* e *page* (descritos no Capítulo 2). O processo de *inquiry* é usado para descobrir os dispositivos vizinhos e para obter seus dados (endereço e relógio). Um dispositivo deve estar no estado de *inquiry* (mestre) e o outro no estado *inquiry scan* (escravo). A mensagem de *inquiry* é um pacote ID de 68 bits. O processo de *page* é similar: o mestre (no estado de *page*) pode fazer a conexão com outro nó (no estado *page scan*) usando o endereço e relógio obtidos no processo de *inquiry*. O mestre deve ficar no estado de *inquiry* por 10,24 s para receber todas as possíveis respostas de seus nós vizinhos num ambiente livre

de erros [BLU01a]. O processo de page dura no máximo um intervalo de *page scan* (1,28 s para o modo R1). Os modos R0 e R3 tem intervalos diferentes. Para a formação da rede, o nó de gerência deve estar no estado de *inquiry* e todos os nós sensores no estado *inquiry scan*. Os fluxogramas do protocolo são mostrados na Figura 6.1.

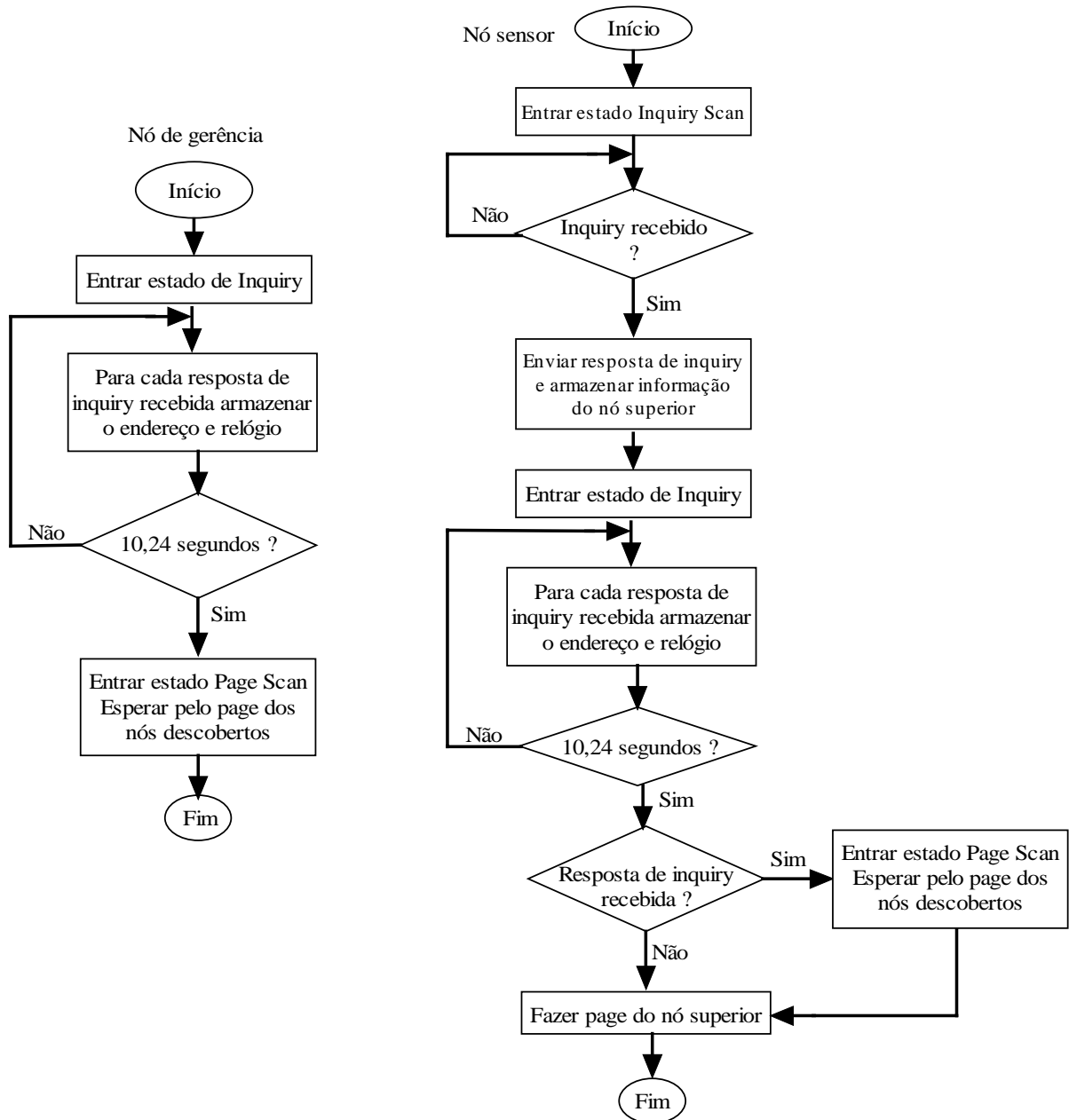


Figura 6.1: Fluxogramas do processo de descoberta

O pacote ID transmitido durante o processo de inquiry terá algumas informações extras: o endereço do dispositivo Bluetooth (BD_ADDR, 48 bits) e o relógio (26 bits). Estes

dados permitem uma descoberta simétrica dos nós. Como exposto em [LIU03], mais 144 bits podem ser adicionados na mensagem de *inquiry* e isto também foi usado no algoritmo de formação da *scatternet route*. Quando um nó sensor recebe uma mensagem de *inquiry*, ele responde ao *inquiry*, armazena o endereço e relógio do pacote ID recebido e muda para o estado de *inquiry*. Quando um nó recebe respostas para seu *inquiry*, ele armazena a informação (endereço e relógio) destes dispositivos. Um nó que muda para o estado de *inquiry* não responderá a outros dispositivos fazendo *inquiry*. Assim, cada nó responderá para apenas um dispositivo.

Um nó no estado de *inquiry* permanece fazendo esta operação por 10,24 s (para obter todas as respostas possíveis) e depois deste tempo tem duas possibilidades: mudar para o estado de *page* ou *page scan*. Se o nó recebeu pelo menos uma resposta ao seu *inquiry*, muda para o estado *page scan* e espera pelo nó (ou nós) que responderam seu *inquiry* para um processo de *page*. Se o nó não recebeu nenhuma resposta para seu *inquiry*, muda para o estado de *page* e faz o *page* do nó superior (o nó a quem ele enviou uma resposta de *inquiry*). Cada nó no estado *page scan* que recebe um *page* de um vizinho também fará o *page* de seu nó superior enviando a informação (relógio e endereço) de todos os seus vizinhos.

Este processo irá terminar no nó de gerência, que está no estado de *page scan*, esperando que seus vizinhos enviem informações sobre a rede. O processo de *page* forma uma conexão entre os nós, mas esta conexão é rompida imediatamente após a transmissão das informações. O nó de gerência recebe os endereços e relógios de todos os nós sensores. O processo de descoberta pode ser feito a qualquer momento que se queira atualizar informações, como obter novos nós que chegaram depois do primeiro processo de descoberta.

Para cada nó sensor o gerenciador saberá seus vizinhos, podendo construir uma tabela de roteamento e estabelecer uma rota para qualquer nó. Neste processo existe apenas uma rota para cada nó. Depois deste processo nenhum nó permanece conectado. Quando o nó de gerência quer coletar ou enviar dados para um sensor, ele procura a rota na tabela, faz o *page* do primeiro nó da rota, que, por sua vez, faz o *page* do segundo nó e assim por diante, até que o destino é alcançado. Nenhuma *scatternet* é formada neste processo. O gerenciador faz o *page* e envia os dados para o primeiro dispositivo, que se desconecta do nó de gerência, faz o *page* e envia as informações para o próximo nó e assim sucessivamente até o destino. Se um nó sensor quer enviar dados, faz o *page* do seu nó superior e transmite os dados. O seu nó superior faz o mesmo até que o nó de gerência seja alcançado.

Embora não exista um atraso para descoberta de rota, o processo de *page* acrescenta um tempo extra quando um nó precisa enviar dados (atraso médio de *page* de $1,28/2 = 0,64s$ por nó). Por exemplo, se a rota tem três saltos (nós), um tempo de 1,92s é adicionado. Este tempo sempre será menor que o da *scatternet route*, que precisa fazer todo o processo de descoberta da rota antes de enviar dados.

6.3. Resultados de Simulação

O simulador do processo de conexão apresentado no Capítulo 4 foi usado para analisar o desempenho do novo algoritmo. Os nós foram distribuídos geograficamente de acordo com a Figura 6.2. Foram considerados três cenários com diferentes distâncias. A distância d é o alcance máximo dos nós. No primeiro cenário a densidade dos nós é baixa, no segundo é média e no terceiro cenário a densidade é alta. A Figura 6.3 mostra o tempo médio que leva para o nó de gerência descobrir os nós e rotas (tempo médio de descoberta). Quando a densidade da rede é baixa, o tempo aumenta rapidamente com o número de nós. Redes com alta densidade têm tempos de descoberta menores. Um exemplo das rotas descobertas com dez nós para o cenário 1 é mostrado na Figura 6.4 e para o cenário 3 na Figura 6.5

Depois da fase de descoberta, todos os enlaces são desconectados. Se o nó de gerência quer enviar alguma informação (como opções de configuração) ou coletar dados do nó sensor 8 na rede da Figura 6.4, os seguintes passos vão ocorrer:

- o nó de gerência faz o *page* do nó 3 e envia os dados;
- nó 3 faz o *page* do nó 6 e transmite os dados;
- nó 6 faz o *page* do nó 8 e envia os dados.

Para completar esta tarefa, são necessários três processos de *page* ($3 \times 0.64s$) mais o tempo para enviar os dados – que será geralmente de poucos *time slots*. Para muitas aplicações este tempo é aceitável. Se o nó sensor 8 quer transmitir, faz o *page* do nó 6 (seu nó superior) e assim sucessivamente até que o nó de gerência receba os dados. Para a rede da Figura 6.5, ainda que todos os nós estão ao alcance uns dos outros, em muitos casos são necessários alguns saltos para uma transmissão. Se o nó de gerência quer transmitir para o sensor 5, os dados terão que passar pelos nós 6 e 9. Como o gerenciador tem informação de todos os nós, uma outra estratégia de transmissão pode melhorar significativamente o desempenho da rede quando a densidade dos nós é alta (Cenário 3 – Figura 6.2).

O gerenciador pode tentar fazer o *page* de um nó verificando o segundo ou terceiro salto na tabela de roteamento. Assim, para transmitir para o nó 5, apenas um *page* seria necessário. Esta abordagem funciona bem quando a densidade dos nós é alta, mas não em outros casos. No exemplo da Figura 6.4, se o nó de gerência tem dados para enviar para o sensor 8 e tenta fazer o *page* desse nó, isto não será possível, pois o nó 8 não está ao seu alcance.

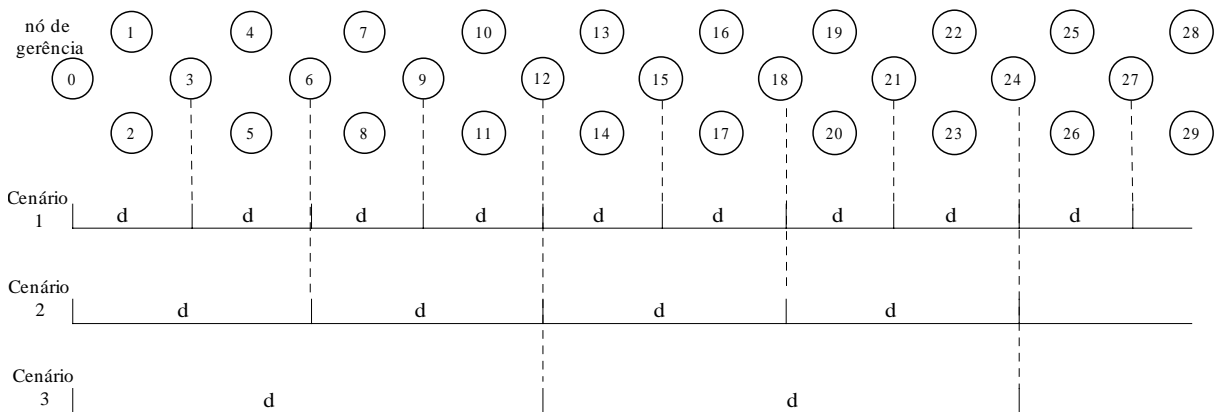


Figura 6.2: Distribuição dos nós

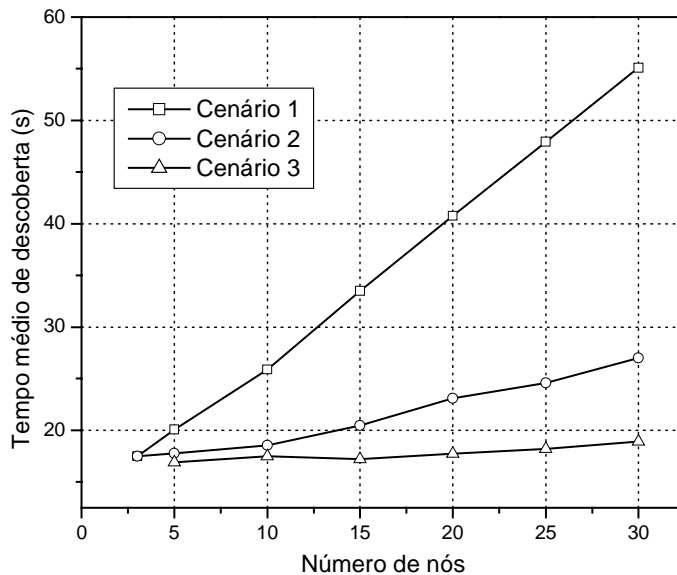


Figura 6.3: Tempo médio de descoberta

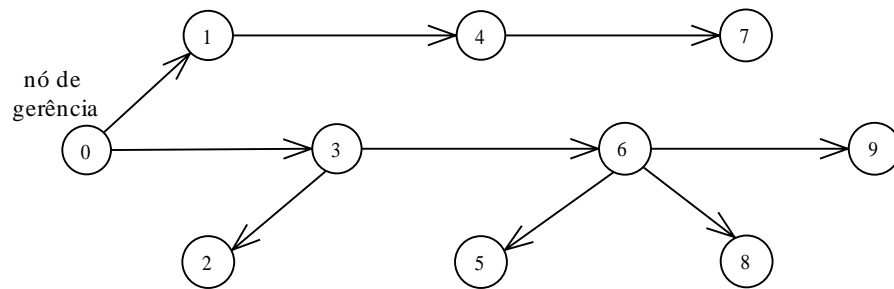


Figura 6.4: Exemplo de rotas descobertas para o cenário 1 com dez nós

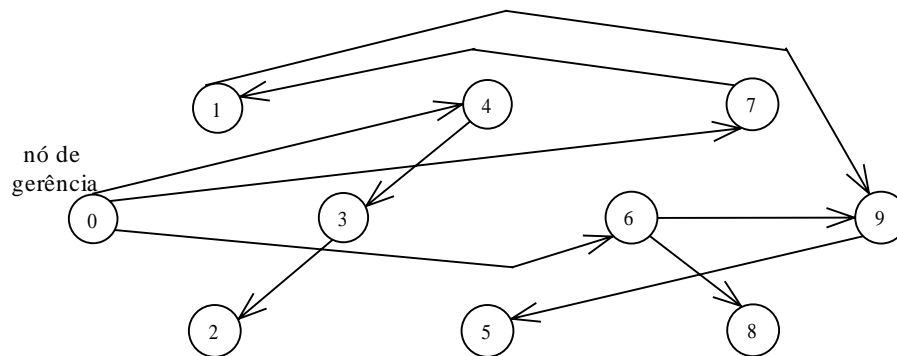


Figura 6.5: Exemplo de rotas descobertas para o cenário 3 com dez nós

6.4. Estudo de Caso de uma Aplicação de Sensoriamento usando Bluetooth

As redes de sensores podem ser usadas para monitorar, rastrear, coordenar e processar dados em diferentes aplicações. Podem ser implementadas em lugares abertos, como em florestas, ruas de uma cidade, oceano ou deserto e ainda em locais internos como prédios e residências. Uma das muitas aplicações de redes de sensores é a localização de pessoas dentro de um ambiente, como um edifício ou um hospital. Num hospital, a localização e monitoramento de médicos e pacientes podem ser bastante úteis, permitindo que funcionários enviem mensagens para médicos e enfermeiros ou monitorem a distância as condições físicas de um paciente, por exemplo. Um protótipo de um sistema remoto de monitoramento de pacientes usando Bluetooth foi desenvolvido em [AND02], usando um sensor e um computador para receber os dados, porém sem a formação de uma rede de sensores.

A idéia é ter um nó de gerência responsável em coletar e enviar dados para os nós sensores. O nó gerenciador pode ser um computador instalado numa sala. Todos os usuários do sistema têm um sensor, que pode ser um sensor de alguma condição fisiológica ou até um

PDA. Como esses nós dos usuários são móveis, deve haver outros tipos de dispositivos, os pontos de acesso (PAs). Este cenário é mostrado na Figura 6.6.

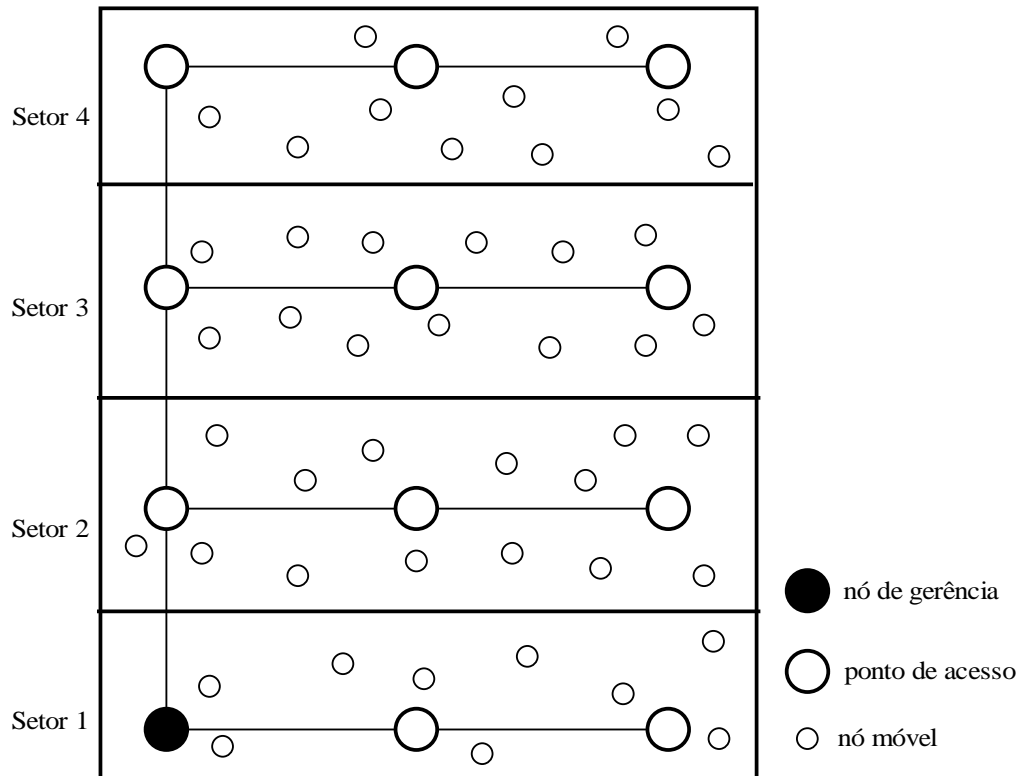


Figura 6.6: Cenário de aplicação

Os PAs podem ser colocados em diferentes andares ou setores e podem ser levados de uma área para outra de acordo com as necessidades da aplicação. Este cenário também pode servir para outras aplicações que tenham pontos de acesso cobrindo uma área, como o monitoramento das condições ambientais de uma região. Os pontos de acesso seriam pontos de coleta de dados e os nós móveis, os sensores (de temperatura, umidade, etc) que podem ser colocados na região.

Para a configuração da rede existem duas fases: a descoberta dos pontos de acesso e a descoberta dos nós móveis (dos usuários). Na primeira fase (Figura 6.1) o nó de gerência está no estado de *inquiry* e os pontos de acesso no estado *inquiry scan*. As rotas são descobertas como descrito na Seção 6.2. Para o cenário da Figura 6.6 com onze PAs, o tempo médio de descoberta, obtido através de simulação, foi de 41,68 segundos. A primeira fase precisa ser executada apenas uma vez se não forem colocados outros pontos de acesso nem movidos os existentes. Funciona como um tipo de inicialização do sistema

A segunda fase é a descoberta dos nós móveis que são os sensores carregados pelas pessoas. Este processo pode ser executado em intervalos regulares ou qualquer hora que o gerenciador queira atualizar as informações sobre os nós. Este processo é diferente da primeira fase e é mostrado na Figura 6.7. O nó de gerência faz o *page* de seus PAs vizinhos, informando que deverão também fazer o *page* dos PAs próximos a eles e entrar depois no estado de *inquiry*. Todos os PAs devem permanecer fazendo *inquiry* por 10,24s para achar os nós dos usuários que estão no estado de *inquiry scan*. Depois do tempo de *inquiry*, os PAs retornam a informação dos nós descobertos para o gerenciador por um processo de *page*. A duração média da segunda fase é:

$$T_d = (2 \times \text{tempo de page} \times \text{número máximo de saltos}) + \text{tempo de inquiry} \quad (6.1)$$

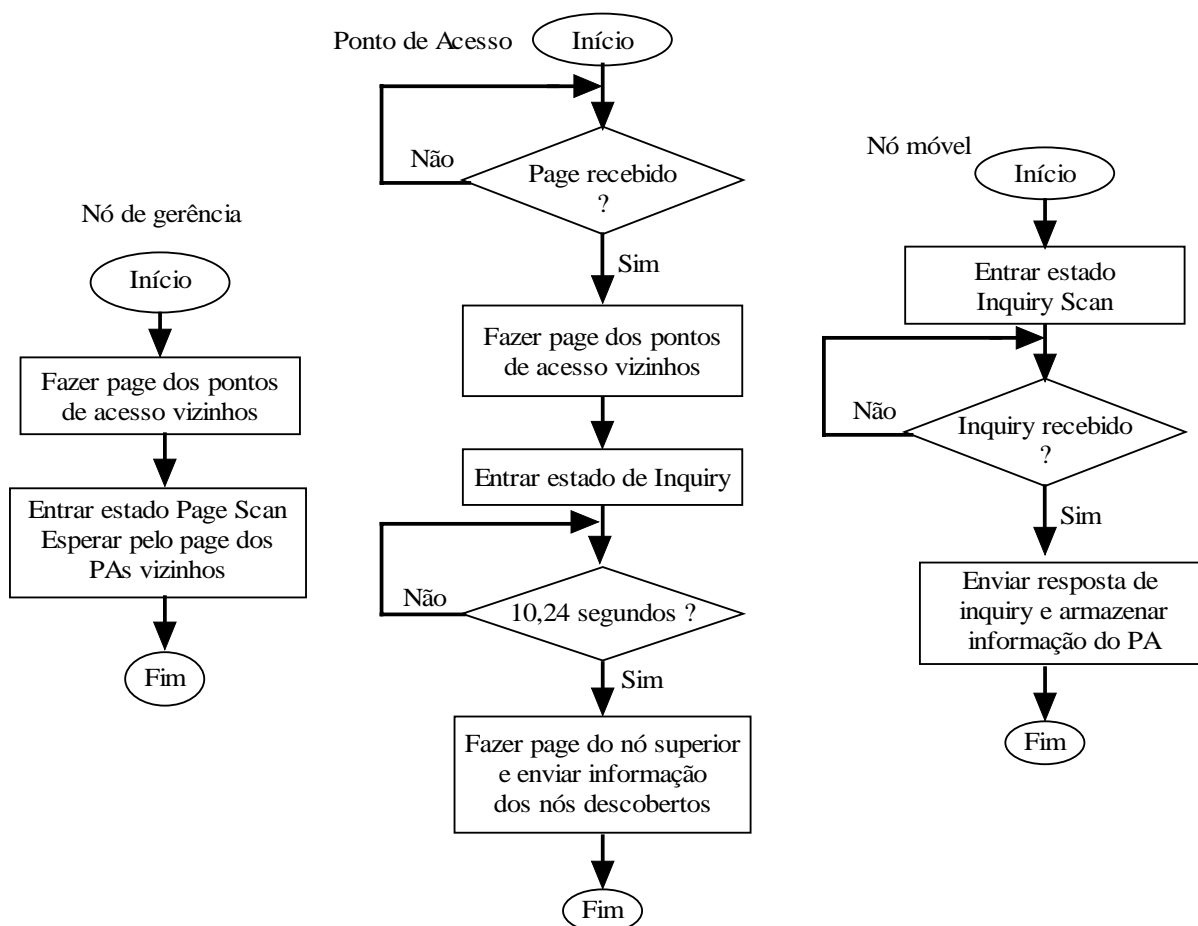


Figura 6.7: Fluxograma da descoberta dos nós móveis

O tempo médio de *page* é de 0,64s. Este tempo é multiplicado por dois (*page* no caminho de ida e volta) e somado ao tempo de *inquiry* (10,24s). No exemplo da Figura 6.6, o tempo T_d seria: $T_d = (2 \times 0,64 \times 5) + 10,24 = 16,64$ segundos. Este tempo da segunda fase é independente do número de nós móveis (46 nós móveis na Figura 6.6). O ponto de acesso fica 10,24s fazendo *inquiry* e escutando por respostas. Se houver um ou quinze nós móveis, o tempo gasto é o mesmo.

O gerenciador tem uma visibilidade de todos os nós e de quais pontos de acesso estão próximos de cada usuário. Para enviar dados para um sensor o tempo médio T_s é:

$$T_s = \text{tempo de page} \times \text{número de saltos} \quad (6.2)$$

Para cinco saltos, $T_s = 3,2$ segundos. Não está sendo considerado o tempo de transmissão, mas o volume de dados neste tipo de aplicação é pequeno e pode ser transmitido em poucos pacotes num curto período de tempo. Cada vez que os dados são enviados de um nó para o outro, a conexão é desfeita. Os nós nunca permanecem conectados e a *scatternet* nunca é formada. Apenas são formadas *piconets* temporárias.

Os nós móveis devem periodicamente tentar se conectar com o ponto de acesso a quem respondeu o *inquiry* para verificar se ainda estão ao seu alcance. Se não obtiver resposta, o nó móvel deve entrar novamente no estado *inquiry scan*. Assim o nó será descoberto por um outro ponto de acesso na próxima vez que o nó de gerência iniciar o processo de descoberta de nós dos usuários. Opcionalmente, um usuário pode manualmente colocar seu dispositivo em estado de procura (*inquiry scan*) se perceber que perdeu a comunicação com a rede. A intervenção do usuário não traz grandes mudanças no sistema, porém numa aplicação como esta é desejável que tudo seja feito de forma transparente, sem que o usuário precise verificar se está ou não na rede. Esta opção poderia ser apenas uma garantia ao usuário que precise se comunicar imediatamente e ainda não foi descoberto pelo nó de gerência ou ponto de acesso.

6.5. Conclusão

A tecnologia Bluetooth ainda tem várias questões que precisam ser resolvidas para que possa ser largamente utilizada em redes de sensores. No entanto, possui várias características importantes que se forem bem aproveitadas fazem do Bluetooth uma das principais

tecnologias para aplicações de redes de sensores. Neste capítulo foi apresentado um protocolo de redes Bluetooth para aplicações onde há um nó responsável pelo gerenciamento da rede. Nesta estrutura não há formação de *scatternet* e os enlaces são desconectados imediatamente após a comunicação, economizando energia e eliminando a necessidade de algoritmos de escalonamento. Para aplicações que o tráfego de dados é baixo, esta abordagem é mais apropriada do que manter todos os nós conectados numa grande *scatternet*. Como estudo de caso, foi considerada uma aplicação formada por nós fixos (gerenciador e pontos de acesso) e móveis (usuários). Este cenário pode ser utilizado como base para outras aplicações de redes de sensores que tenham pontos de acesso cobrindo uma determinada área. Exemplos são o monitoramento das condições ambientais de uma região ou também em shoppings e aeroportos, onde os PAs podem ser usados para dar informações do local para os usuários (equipados com um celular ou PDA, por exemplo).

Capítulo 7

Conclusão

A tecnologia Bluetooth é ainda recente e tem potencial para ser empregada nas mais diversas aplicações, muitas das quais só serão viabilizadas depois de muita pesquisa e desenvolvimento. Novas propostas apresentadas na literatura podem ser incorporadas em futuras versões da especificação Bluetooth. Neste trabalho foi feito o estudo de redes Bluetooth em canais com desvanecimento Nakagami e sua aplicação em redes de sensores. O desempenho em ambientes Nakagami foi analisado através de modelagem analítica e de simulação. Foi apresentado um algoritmo de escalonamento *intrapiconet* que usa a informação do estado de canal e uma estratégia para uso do Bluetooth em redes de sensores. Uma das principais dificuldades encontradas foi no uso de simulação para a obtenção de resultados, pois as ferramentas existentes são limitadas e possuem uma documentação incompleta. Por isso, para as estratégias de escalonamento e formação de redes Bluetooth foram implementados novos simuladores.

Baseado nos resultados e análises feitas na dissertação pode-se destacar as seguintes conclusões:

- A variabilidade das condições do canal de rádio afeta o desempenho de enlaces Bluetooth de maneira bastante significativa. Em *scatternets*, onde os pacotes passam por mais de um enlace quando são transmitidos, a vazão também sofre uma grande redução em canais ruins, como mostrados em resultados de simulação.

- O alcance dos dispositivos Bluetooth previstos na especificação, de 100 metros para classe 1 e 10 metros para classe 3, pode ser maior ou menor, dependendo das condições do canal (parâmetro m) e do ambiente em que estão operando (expoente de perda de percurso η).
- O algoritmo de escalonamento *intrapiconet* BCSS, baseado na estimação da qualidade do canal, pode ser aplicado em ambientes que as condições de canal variam rapidamente, como em algumas aplicações de redes de sensores e ambientes com mobilidade. A estratégia de escalonamento BCSS mostrou-se eficiente para condições de tráfego altas em comparações com o *round robin* e *round robin exaustivo* usando tráfego Poisson e HTTP.
- Embora a tecnologia Bluetooth não tenha sido projetada especificamente para redes de sensores, a estrutura de rede proposta, com um nó de gerência e pontos de acesso, pode ser utilizada em diversas aplicações que não requerem transmissão de dados em tempo real. Como não há formação de uma *scatternet* para a rede de sensores, não são necessários algoritmos de escalonamento e os dispositivos só se conectam quando precisam enviar ou receber dados.

Os trabalhos futuros propostos nesta área de pesquisa são os seguintes:

- Analisar o desempenho de outras topologias de *scatternets* em ambientes Nakagami através de simulação e propor topologias que podem minimizar os efeitos das condições ruins de canal.
- Determinar e investigar a influência de diferentes valores para o limiar de m e de SNR no algoritmo BCSS. Opcionalmente, propor um esquema adaptativo que altere os limiares dinamicamente, de acordo com a variação da qualidade do canal.
- Comparar o algoritmo BCSS com outras estratégias de escalonamento de *piconets* e outros modelos de tráfego.

- Implementar um algoritmo de escalonamento *interpiconet* baseado no BCSS, usando o parâmetro m e a relação sinal-ruído. O nó ponte numa *piconet* pode ser escolhido levando em consideração o valor de m , por exemplo. O nó da *piconet* que tiver o maior valor de m no enlace com o nó ponte da outra *piconet* é o escolhido.
- Utilizar o parâmetro de Nakagami como uma métrica para algoritmos de formação de *scatternets*, roteamento de pacotes e técnicas de economia de energia. As rotas numa *scatternet* podem ser escolhidas baseadas nos valores de m e SNR.
- Pesquisar outros algoritmos para redes de sensores que também não formem *scatternets*. Considerar outros tipos de aplicações, sem nó de gerência e pontos de acesso.
- Analisar o consumo de energia de diferentes estruturas de rede e topologias de *scatternets* para redes de sensores.
- Melhorar os esquemas de descoberta de nós sensores, para que o tempo de atualização de rotas seja o menor possível e a estratégia possa ser utilizada em aplicações com requisitos de tempo mais rígidos.

Referências Bibliográficas

- [AND02] ANDREASSON, J.; EKSTROM, M.; FARD, A.; CASTAÑO, J.G.; JOHNSON, T. *Remote System for Patient Monitoring Using Bluetooth*. First IEEE International Conference on Sensors, EUA, Junho 2002.
- [AKY02] AKYILDIZ, I.F.; SU, W.; SANKARASUBRAMANIAM, Y.; CAYIRCI, E. *A Survey on Sensor Networks*. IEEE Communications Magazine, pp. 102-114, Agosto 2002.
- [BHA99] BHAGWAT, P e SEGALL, A. *A Routing Vector Method (RVM) for Routing in Bluetooth Scatternets*. IEEE International Workshop on Mobile Multimedia Communications, EUA, Novembro 1999.
- [BHA01] BHAGWAT, P. *Bluetooth: Technology for Short-range Wireless Apps*. IEEE Internet Computing, Maio 2001.
- [BLU01a] BLUETOOTH SIG, *Specification of the Bluetooth System – Core, Version 1.1*. Fevereiro 2001. <http://www.bluetooth.com>
- [BLU01b] BLUETOOTH SIG, *Specification of the Bluetooth System – Profile, Version 1.1*. Fevereiro 2001. <http://www.bluetooth.com>
- [BLU03] BLUETOOTH SIG, *Specification of the Bluetooth System – Core, Version 1.2*. Novembro 2003. <http://www.bluetooth.com>
- [BUH02] BÜHRER, L. e KASPER, S. *Jini Discovers Bluetooth – Semester Thesis*. Swiss Federal Institute of Technology Zurich, Julho 2002.

- [CAP01] CAPONE, A.; GERLA, M.; KAPOOR, R. *Efficient Polling Schemes for Bluetooth Picocells*. IEEE International Conference on Communications, Finlândia, Junho 2001.
- [CHE02] CHENG, J. e BEAULIEU, N. C. *Generalized Moment Estimators for the Nakagami Fading Parameter*. IEEE Communication Letters, vol. 6, no.4, pp. 144-146, Abril 2002.
- [COR02] CORDEIRO, C. M. e AGRAWAL, D.P. *Mobile Ad Hoc Networking*. XX Simpósio Brasileiro de Redes de Computadores, Búzios, Maio 2002.
- [COR03] CORDEIRO, C. M.; GOSSAIN, H.; ASHOK, R.L.; AGRAWAL, D.P. *The Last Mile: Wireless Technologies for Broadband and Home Networks*. XXI Simpósio Brasileiro de Redes de Computadores, Natal, Maio 2003.
- [FER03] FERRARI, P.; FLAMMINI, A.; MARIOLI, D.; SISINNI, E., TARONI, A. *A Bluetooth-based Sensor Network with Web Interface*. Instrumentation and Measurement Technology Conference, Vail, EUA, Maio 2003.
- [FRE01] FREITAS Filho, P.J. *Introdução à Modelagem e Simulação de Sistemas*. Visual Books, 2001.
- [GOL02] GOLMIE, N. *Performance Evaluation of a Bluetooth Channel Estimation Algorithm*. IEEE International Symposium on Personal Indoor and Mobile Radio Communications, Lisboa, Portugal, Setembro 2002.
- [HAR02] HAR-SHAI, L.; KOFMAN, R.; ZUSSMAN, G.; SEGALL, A. *Interpiconet Scheduling in Bluetooth Scatternets*. OPNETWORK Conference, Agosto 2002.
- [IBM00] IBM. *BlueHoc: Bluetooth Performance Evaluation Tool*. Disponível em: <http://oss.software.ibm.com/developerworks/opensource/bluehoc>.
- [JOH99] JOHANSSON, N.; KORNER, U.; JOHANSSON, P. *Performance Evaluation of Scheduling Algorithms for Bluetooth*. IFIP TC6 International Conference on Broadband Communications, Hong Kong, Novembro 1999.

- [JOH01a] JOHANSSON, P.; KAPOOR, R.; KAZANTZIDIS, M.; GERLA, M. *Bluetooth: An Enabler of Personal Area Networking*. IEEE Networks, Special Issue in Personal Area Networks, Outubro 2001.
- [JOH01b] JOHANSSON, N.; ALRIKSSON, F.; JÖNSSON, U. *Jump Mode – A Dynamic Window-based Scheduling Framework for Bluetooth Scatternets*. Proceedings of ACM MobiHoc, Outubro 2001.
- [JOH02] JOHANSSON, N.; KAPOOR, R.; KAZANTZIDIS, M.; GERLA, M. *Rendezvous Scheduling of Bluetooth Scatternets*. IEEE International Conference on Communications, Nova Iorque, EUA, Abril 2002.
- [JU02] JU, M.C.; PARK, C.H.; HONG, D.K.; YOUN, K.J.; CHO, J.W. *Link Management Scheme of Bluetooth Based on Channel Quality Estimation*. IEE Electronics Letters, vol. 38. no. 15, pp. 789-790, Julho 2002.
- [KAL99] KALIA, M.; BANSAL, D.; SHOREY, R. *MAC Scheduling and SAR Policies for Bluetooth: A Master Driven TDD Pico-Cellular Wireless System*. IEEE International Workshop on Mobile Multimedia Communications, São Diego, EUA, Novembro 1999.
- [KAP03] KAPOOR, R. e GERLA, M. *A Zone Routing Protocol for Bluetooth Scatternets*. IEEE Wireless Communications and Networking Conference, EUA, Março 2003.
- [KAS01] KASTEN, O. e LANGHEINRICH, M. *First Experiences with Bluetooth in the Smart-Its Distributed Sensor Network*. Workshop on Ubiquitous Computing and Communications, Barcelona, Espanha, Setembro 2001.
- [KLE03] KLEINSCHMIDT, J.H.; PELLENZ, M.E.; JAMHOUR, E. *Bluetooth Network Performance in Nakagami-m Fading Channels*. The Fifth IFIP TC6 International Conference on Mobile and Wireless Communications Networks, Cingapura, Outubro 2003.

- [KLE04a] KLEINSCHMIDT, J.H.; PELLENZ, M.E.; LIMA JR, L.A.P. *An Efficient Polling Strategy for Bluetooth Piconets using Channel State Information*. IEEE 6th CAS Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication, Xangai, China, Maio 2004
- [KLE04b] KLEINSCHMIDT, J.H.; PELLENZ, M.E.; LIMA JR, L.A.P. *Evaluating and Improving Bluetooth Piconet Performance over Nakagami-m Fading Channels*. The Ninth IEEE International Symposium on Computers and Communications, Alexandria, Egito, Junho 2004
- [KOU02] KOUPAL, B.; WANG, M.; EDELMAN, C. *Case History: Development of a Bluetooth Transceiver in RF CMOS*. Signia and Agilent Technologies, Abril 2002.
- [LAP03] LAPEYRIE, J.B. e TURLETTI, T. *FPQ: a Fair and Efficient Polling Algorithm with QoS Support for Bluetooth Piconet*. IEEE Infocom, São Francisco, EUA, 2003.
- [LEO03] LEOPOLD, M.; DYDENSBORG, M.D.; BONNET, P. *Bluetooth and Sensor Networks: A Reality Check*, 1st ACM Conference on Sensor Systems. Los Angeles, EUA, Novembro 2003.
- [LIN03] LIN, T.; TSENG, Y.; CHANG, K.; TU, C. *Formation, Routing and Maintenance Protocols for the BlueRing Scatternet of Bluetooth*. 36th Hawaii International Conference on System Science (HICSS-36), Janeiro 2003.
- [LIU02] LIU, Y.; ZHANG, Q.; ZHU, W. *A Priority-Based MAC Scheduling Algorithm for Enhancing QoS Support in Bluetooth Piconet*. IEEE International Conference on Communications, Circuits and Systems, Julho 2002.
- [LIU03] LIU, Y., LEE, M.J.; SAADAWI, T.N. *A Bluetooth Scatternet-Route Structure for Multihop Ad Hoc Networks*. IEEE Journal on Selected Areas in Communications, vol. 21, no. 2, pp. 229-239, Fevereiro 2003.

- [LOU03] LOUREIRO, A.F.; NOGUEIRA, J.M.; RUIZ, L.B.; MINI, R.A.; NAKAMURA, E.F.; FIGUEIREDO, C.M. *Redes de Sensores sem Fio*. XXI Simpósio Brasileiro de Redes de Computadores, Natal, Maio 2003.
- [MAT00] *Matlab version 6.0 Release 12*. Site: www.mathworks.com.
- [MIL01] MILLER, MICHAEL. *Discovering Bluetooth*. Sybex Inc, 2001.
- [NS2] *ns-2 Network Simulator*. Disponível em <http://www.isi.edu/vint/nsnam>.
- [PET03] PETRIOLI, C.; BASAGNI, S.; CHLAMTAC, I. *Configuring BlueStars: Multihop Scatternet Formation for Bluetooth Networks*. IEEE Transactions on Computers, vol.52, no. 6, pp. 779-790, Junho 2003.
- [PRA02] PRABHU, B.J. e CHOCKALINGAM, A. *A Routing Protocol and Energy Efficient Techniques in Bluetooth Scatternets*. IEEE International Conference on Communications, Nova Iorque, EUA, Abril 2002.
- [PRI03] PRIESS, W.; REZENDE, J.F.; PIRMEZ, L.; CARMO, L.F.C. *Um Mecanismo de Escalonamento Parametrizável para Scatternets Bluetooth*. XXI Simpósio Brasileiro de Redes de Computadores, Natal, Maio 2003.
- [PRO01] PROAKIS, J. *Digital Communications*. New York, NY: McGraw-Hill, 4th edition, 2001.
- [RAC01] RACZ, A.; MIKLOS, G.; KUBINSZKY, F.; VALKO, A. *A Pseudo Random Coordinated Scheduling Algorithm for Bluetooth Scatternets*. Proceedings of ACM MobiHoc, Outubro 2001.
- [RAM01] RAMESH, A.; CHOCKALINGAM, A.; MILSTEIN, L.B. *SNR Estimation in Generalized Fading Channels and its Application to Turbo Decoding*. IEEE International Conference on Communications, Finlândia, Junho 2001.

- [RAP99] RAPPAPORT, T. S. *Wireless Communications: Principles and Practice*, Prentice-Hall PTR, 1999.
- [ROU02] ROUHANA, N. e HORLAIT, E. *Bluetooth Web Internet Gateway*. IEEE Symposium on Computers and Communications, Itália, Junho 2002.
- [SHI03] SHIH, K.; WANG, S.; SU, J. *A Bluetooth Group-Scatternet Formation Algorithm for Efficient Routing*. IEEE International Conference on Parallel Processing Workshops, Taiwan, Outubro 2003.
- [STU02] STUCKMANN, P. e HOYMANN, C. *Performance Evaluation of WAP-based Applications over GPRS*. IEEE International Conference on Communications, Nova Iorque, EUA, Abril 2002.
- [SUN02] SUN, M.; CHANG, C.; LAI, T. *A Self-Routing Topology for Bluetooth Scatternets*. IEEE International Symposium on Parallel Architectures, Algorithms and Networks, Filipinas, Maio 2002.
- [TAN02a] TAN, G.; MIU, A; BALAKRISHNAN, H.; GUTTAG, J. *An Efficient Scatternet Formation Algorithm for Dynamic Environments*. IASTED Intern. Conference on Communication Networks (CCN02), Cambridge, EUA, Novembro 2002.
- [TAN02b] TAN, G. e GUTTAG, J. *A Locally Coordinated Scatternet Scheduling Algorithm*. IEEE Conference on Local Computer Networks, Tampa, EUA, Novembro 2002.
- [TAN02c] TAN, G. *Blueware: Bluetooth Simulator for ns*. MIT Technical Report. Disponível em: <http://nms.lcs.mit.edu/project/blueware>.
- [VAL02a] VALENTI, M. C.; ROBERT, M.; REED, J.H. *On the Throughput of Bluetooth Data Transmissions*. IEEE Wireless Communications and Networking Conference, Orlando, EUA, Março 2002.

- [VAL02b] VALENTI, M. e ROBERT, M. *Improving the QoS of Bluetooth Through Turbo coding*. IEEE Military Communications Conference, Los Angeles, EUA, Outubro 2002.
- [WAN02] WANG, Z.; THOMAS, R.J.; HAAS, Z. *BlueNet – A New Scatternet Formation Scheme*. Hawaii International Conference on System Science (HICSS-35), Janeiro 2002.
- [WEI91] WEISER, Mark. *The Computer for the 21st Century*. Scientific American, vol. 265, pp. 94-104, Setembro 1991.
- [WWR01] WIRELESS WORLD RESEARCH FORUM, *The Book of Visions – Visions of the Wireless World*, 2001. <http://www.wireless-world-research.org>
- [ZAN02] ZANG, Q.T. *A Note on the Estimation of Nakagami-m Fading Parameter*. IEEE Communication Letters, vol. 6, no. 6, pp. 237-328, Junho 2002.
- [ZAR01] ZARUBA, G.; BASAGNI, S.; CHLAMTAC, I. *BlueTrees – Scatternet Formation to enable Bluetooth-based Personal Area Networks*. IEEE International Conference on Communications, Finlândia, Junho 2001.