

Caracterização de FHSS em Modelos Estáticos de Formação Scatternet Bluetooth

Célio Márcio Soares Ferreira
Universidade Federal de Ouro Preto

Dissertação submetida ao
Instituto de Ciências Exatas e Biológicas
Universidade Federal de Ouro Preto
para obtenção do título de Mestre em Ciência da Computação

F383c Ferreira, Célio Márcio Soares.
Caracterização de FHSS em modelos estáticos de formação scatternet
bluetooth [manuscrito] / Célio Márcio Soares Ferreira – 2013.
93 f.: il. color.; graf.; tab.

Orientador: Prof. Dr. Ricardo Augusto Rabelo de Oliveira.

Dissertação (Mestrado) - Universidade Federal de Ouro Preto. Instituto
de Ciências Exatas e Biológicas. Departamento de Computação. Programa
de Pós-graduação em Ciência da Computação.

Área de concentração: Ciência da Computação.

1. Sistema de comunicação sem fio - WPAN (*Wireless Personal Area
Network*) - Teses. 2. Teleinformática - Teses. 3. Tecnologia Bluetooth -
Scatternet - Teses. 4. Teoria dos grafos - Grafo dinâmico - Teses. I. Oliveira,
Ricardo Augusto Rabelo de. II. Universidade Federal de Ouro Preto.
III. Título.

CDU: 004.72

Catálogo: sisbin@sisbin.ufop.br

Dedico este trabalho a minha mãe Maria José Soares Ferreira e a minha avó Maria José Abreu Soares, exemplos de força de vontade, honestidade e carinho em minha caminhada até aqui.

Caracterização de FHSS em Modelos estáticos de formação Scatternet Bluetooth

Resumo

O potencial das aplicações *WPAN - Wireless Persona Network* se encontra praticamente inexplorado. O baixo custo e a sua presença na maioria dos dispositivos móveis, faz do *Bluetooth* a tecnologia a ser explorada. Os modelos que representam e formam redes *Bluetooth*, negligenciam as características de sua técnica de comunicação, o *FHSS - Frequency-hopping spread spectrum*. Parte do desafio na geração de algoritmos *Scatternet* eficientes está relacionado as características desta técnica. Definimos o grafo dinâmico *Bluetooth*, descrevendo as restrições e novos procedimentos, que representam os atrasos na inicialização de redes, efeitos colaterais do *FHSS*. Ajustamos um modelo centralizado de *Scatternets*, para resultar em topologias coerentes com as encontradas por protocolos em um modelo dinâmico.

Characterization of FHSS in Static Bluetooth Scatternet Formation Models

Abstract

The potential of WPAN – Wireless Personal Network applications is virtually untapped. The low cost and its presence in most mobile devices, make of Bluetooth the technology to be explored. The models with represent and form networks Bluetooth, neglect the characteristic of their communication technique, the FHSS - Frequency-hopping spread spectrum. Part of the challenge of generating efficient Scatternet algorithms is related to its characteristics. We define a dynamic graph of Bluetooth, describing the constraints and new procedures, which represent the delays to network startup, side effects of FHSS. We adjusted a centralized model of Scatternet, to generate topologies consistent with those found by protocols in a dynamic model.

Declaração

Esta dissertação é resultado de meu próprio trabalho, exceto onde referência explícita é feita ao trabalho de outros, e não foi submetida para outra qualificação nesta nem em outra universidade.

Célio Márcio Soares Ferreira

Agradecimentos

Agradeço ao meu *filho Marco Antonio*, pela nova razão e forma de viver.

Agradeço a todos *profissionais* que me ajudam direta ou indiretamente neste trabalho.

Agradeço a *minha esposa*, pela paciência e carinho.

Agradeço a *minha família*, pela paz interior.

Agradeço a *UFOP*, pelo conhecimento adquirido.

Agradeço a *minha mãe*, pela fé em minha capacidade.

Agradeço ao meu orientador *Ricardo Rabelo*, pelo condução, dedicação e potencialização de minha capacidade.

Agradeço a *equipe da LinuxPlace*, pela ajuda em meus momentos de ausência.

Agradeço ao *FINEP*, pelo apoio e motivação de meu tema de trabalho.

Muito Obrigado.

Prefácio

Aplicações usando Redes *Wireless Personal Area Network* (WPAN) se encontram praticamente inexploradas. Nas WPANs, as interfaces de baixo custo e baixo consumo de energia destes dispositivos, como *Bluetooth* e *Zigbee*, são usadas para formar topologias de rede de mais de um salto, interligadas por *bridges*. A WPAN estende a função tradicional destas interfaces, substituir cabos de dados em comunicação ponto a ponto, possibilitando a implementação de soluções como:

- Troca de mensagens, perfis e localização usando redes com pessoas próximas ao campo visual em formações de redes sociais móveis;
- Redes para acesso Internet usando os dispositivos móveis como repetidores e *bridges* até o *mesh* portal;
- Redes de Monitoramento para dispositivos de suporte a vida integrados a casa inteligente;
- Aplicações de prédios industriais inteligentes, em que informações de sensoria-mento do ambiente e dos equipamentos são trocadas via redes de baixo consumo energético;
- Jogos sociais e de relacionamentos, onde perfis de compatibilidade e desejos são considerados para encontro de casais ou início dos jogos, nivelados pela troca de informação de experiência do usuário.

Das tecnologias WPAN, *Bluetooth* é a mais popular , os números de vendas dos últimos anos atestam sua popularidade. Em 2010, 906 milhões de dispositivos móveis *Bluetooth* foram vendidos; 77 por cento dos 171 milhões de laptops e na indústria de

Jogos, 62 por cento dos mais de 50 milhões de consoles foram exportados. Em 2011 na área da saúde foi previsto que mais de 40 milhões de dispositivos médicos teriam *Bluetooth* habilitado. Uma previsão de que um terço dos veículos produzidos em 2011 incluiriam *Bluetooth*, com um crescimento de 70 por cento até 2016.

Uma rede *Bluetooth* é chamada *Piconet*, em que nós atuam como mestre e escravo. Cada *Piconet* possui um mestre, e um máximo de sete escravos. A um conjunto de *Piconets* interligadas por um nó no papel de *Bridge* damos o nome de *Scatternet*.

Os dispositivos *Bluetooth* comunicam usando canais em salto de frequência coordenado pelo mestre, e calculada com o *clock* e o endereço de rede do mestre, sendo um padrão de salto conhecido por toda *Piconet* em uma técnica chamada *Frequency Hopping Spread Spectrum* (FHSS).

Esta pesquisa tem resultados em dois itens relacionados ao uso do FHSS no *Bluetooth* considera:

- O custo da aleatoriedade necessária até o casamento do padrão de salto no processo de descoberta de nós;
- As influências na performance e consumo de energia das *bridges* nas formações *Scatternet*.

Os modelos existentes negligenciam o *FHSS* e parte do desafio de gerar algoritmos *Scatternets* eficientes são relacionadas a estas características. Nós propomos um modelo de *Bluetooth* usando grafos dinâmicos e mostramos o impacto de seu uso nas *Piconets* e *Scatternets*. Mostramos que pequenas mudanças no modelo centralizado de *Scatternet*, considerando o impacto das *bridges*, usando programação linear, resultam de topologias similares a modelos dinâmicos constituídos de algoritmos distribuídos.

Sumário

Lista de Figuras	xix
Lista de Tabelas	xxi
Nomenclatura	1
1 Introdução	3
1.1 Apresentação	3
1.2 Justificativa	6
1.3 Motivação	7
1.4 Contribuições	8
1.5 Organização do Texto	9
2 Revisão Bibliográfica	11
3 Wireless Personal Area Network	15
3.1 Introdução	15
3.2 <i>FHSS (Frequency-hopping spread spectrum)</i>	15
3.3 Bluetooth	16
3.4 <i>ZigBee</i>	17
3.5 Conclusão	18

4	<i>Piconet</i>	21
4.1	Introdução	21
4.2	Características	21
4.3	Conclusão	28
5	Scatternet	29
5.1	Introdução	29
5.2	Características das <i>Scatternets</i>	29
5.3	Modelos Dinâmicos	30
5.3.1	Protocolos <i>Scatternet</i>	32
5.3.2	Algoritmo Law	32
5.3.3	Bridges	34
5.3.4	Inter-Piconet Scheduling	35
5.4	Modelo Estático	36
5.4.1	Modelo centralizado de Marsan et al.	36
5.5	Influência das bridges na eficiência	37
5.6	Conclusão	38
6	Grafo Dinâmico	39
6.1	Introdução	39
6.2	Definindo <i>Bluetooth</i> como grafo dinâmico	39
6.3	Conclusão	43
7	Os Efeitos do FHSS	45
7.1	Introdução	45
7.2	Experimentos	45

7.3	Análise do tempo de convergência total das redes	46
7.4	Conclusão	48
8	Análise da Topologia	49
8.1	Introdução	49
8.2	Análise	49
8.3	Expandindo o modelo centralizado de Marsan et al.	51
8.4	Conclusão	56
9	Conclusão	59
A	Apêndice	61
A.1	Script NS2 - UCBT e Modelos AMPL	61
	Referências Bibliográficas	63
	Índice Remissivo	67

Lista de Figuras

3.1	<i>FHSS</i> no <i>Bluetooth</i>	16
3.2	Piconet	16
3.3	Zigbee	19
4.1	Piconet Bluetooth	22
4.2	Máquina de Estado <i>BlueTooth</i>	23
4.3	Estados até a formação de nova <i>Piconet</i>	26
5.1	Scatternet Bluetooth	30
6.1	Dispositivos u v ao alcance um do outro	41
6.2	u v iniciam o procedimento distribuído <i>Disc()</i>	41
6.3	Ocorre a coincidência de canal f_i	42
6.4	v retorna seu endereço de rede e relógio	42
6.5	Dispositivos usam procedimento <i>FHS()</i> para gerar F'	42
6.6	Com a mesma sequência de frequência F' agora pode ocorrer comunicação criação e sincronização da <i>Piconet</i>	42
6.7	Nós recebem etiquetas de M - mestre e S - escravo	42
7.1	Tempo até a primeira resposta ao <i>Inquiry</i> do candidato a mestre	46

7.2	Tempo até a formação da primeira <i>Piconet</i> máxima com sete escravos e um mestre	47
7.3	Número de pacotes INQUIRY necessários até a formação de uma <i>Piconet</i> com máxima capacidade	48
8.1	Topologia <i>Scatternet</i> encontrada na simulação do algoritmo Law et al. de uma rede com 20 dispositivos	50
8.2	Topologia <i>Scatternet</i> encontrada na simulação do algoritmo Law et al. de uma rede com 30 dispositivos	51
8.3	Topologia <i>Scatternet</i> de 20 dispositivos encontrado pelo modelo estático de Marsan et al.	52
8.4	Topologia <i>Scatternet</i> de 20 dispositivos, encontrada como solução de nosso modelo estático	56
8.5	Topologia <i>Scatternet</i> de 30 dispositivos, encontrada como solução do modelo de Marsan et al.	57
8.6	Topologia <i>Scatternet</i> de 30 dispositivos, encontrada como solução de nosso modelo.	58

Lista de Tabelas

3.1	Taxas de Transmissão do <i>Bluetooth</i>	17
3.2	Alcances do <i>Bluetooth</i>	17
4.1	Algoritmos <i>Intra-piconet Scheduling</i>	28
5.1	Protocolos <i>Scatternet</i>	33
8.1	Restrições do modelo estático de	54
8.2	Parâmetros de Entrada	55
8.3	Parâmetro de entrada para <i>Scatternet</i> de 30 dispositivos	55

*“no mesmo galho
uma formiga a passeio
outra a trabalho.”*

— Ricardo Silvestrin

Nomenclatura

ACL	<i>Asynchronous Connection-oriented Logical transport</i>
AFH	<i>Adaptive Frequency-hopping</i>
APPD	<i>Adaptive Presence Point Density</i>
BD_ADDR	<i>Bluetooth device address</i>
DSFP	<i>Distributed Scatternet Formation Procedure</i>
EDR	<i>Enhanced Data Rate</i>
FHS	<i>Frequency Hopping Synchronization</i>
FHSS	<i>Frequency-hopping spread spectrum</i>
FSS	<i>Flexible Scatternet Scheduling</i>
GFSK	<i>Gaussian Frequency Shift Keying</i>
HID	<i>Human Interface Device</i>
IAC	<i>Inquiry Access Code</i>
IHS	<i>inquiry hopping sequence</i>
IRHS	<i>inquiry response hopping sequence</i>
LAA	<i>Load Adaptive Algorithm</i>
LCS	<i>Locally Coordinated Scheduling</i>
MDRP	<i>Maximum Distance Rendezvous Point</i>
PCSS	<i>Pseudo-Random Coordinated Scatternet Scheduling</i>
RP	<i>Rendezvous Point</i>
SCO	<i>Synchronous Connection Oriented Link</i>

SIG	<i>Special Interest Group</i>
TDD	<i>Time Division Duplex</i>
TDM	<i>Time Division Multiplexing</i>
UDG	<i>Unit Disk Graph</i>
WPAN	<i>Wireless Personal Area Network</i>

Capítulo 1

Introdução

1.1 Apresentação

Aplicações que usam as formações *Wireless Personal Area Network (WPANS)* ainda não foram exploradas em toda sua potencialidade. A possibilidade de formar redes *Ad Hoc* de maior abrangência entre dispositivos de baixo custo e baixo consumo de energia, permite extrapolar seu uso mais comum: a substituição dos cabo de dados para conexão de dispositivos periféricos.

Podemos destacar algumas aplicações *WPAN*:

- Redes sociais *mobile* para troca de perfis, mensagens e localização, mesmo em estado *offline* com a internet,
- Redes *mesh* para acesso a internet usando dispositivos móveis como repetidores,
- Redes de dispositivos médicos de sensoriamento no suporte a vida carregados por pacientes em um hospital,
- Automação residencial (casa inteligente) e industrial,
- Aplicações de namoro, jogos sociais, podem usar a proximidade dos usuários e seu estado civil, exposta em um rede de dispositivos móveis, para formar pares de participantes.

Para formar uma *WPAN* usamos dispositivos *ZigBee* e *Bluetooth* que possuem características técnicas fundamentais como: portabilidade; baixo custo de *startup* e ma-

nutenção; baixo custo do dispositivo; baixo consumo de energia; possibilidade de formar redes espontâneas.

Das tecnologias capazes de formar *WPAN*, *Bluetooth* é a mais popular (Bluetooth.com Last Visited in 11/02/2013). Ele já é encontrado na maioria dos *Smartphones e Tablets* comercializados. Os números de vendas dos últimos anos confirmam sua popularidade:

- Em 2010, 906 milhões de dispositivos móveis *Bluetooth* foram vendidos;
- 77% dos 171 milhões de *laptops* e na indústria de Jogos, 62% dos mais de 50 milhões de consoles foram exportados;
- Em 2011 foram previstos na área da saúde, que mais de 40 milhões de dispositivos médicos *Bluetooth* seriam habilitados.
- Uma previsão de que $\frac{1}{3}$ dos veículos produzidos em 2011 incluiriam *Bluetooth*, com um crescimento de 70% até 2016.

O *Bluetooth* usa rádio frequência para gerar redes de curto alcance, sendo mais usado na substituição aos tradicionais cabos que interligam dispositivos e periféricos. Sua frequência utiliza a popular banda 2.4 GHz *ISM* (*Industrial, Scientific and Medical*), e possui velocidades de transferência superiores a 1 Mbps.

A formação de rede básica do *Bluetooth* é chamada *Piconet*. Nesta formação os nós assumem diferentes papéis, mestre e escravo. Em cada *Piconet* existe somente um mestre ativo e no máximo sete escravos. O mestre é responsável pela sincronização e controla o fluxo de dados entre os nós escravos em uma comunicação chamada *intra-Piconet*.

Para que o limite desta comunicação sejam expandidas, temos as *Scatternets*. Elas são coleções de *Piconet* formadas segundo um algoritmo distribuído, possibilitando que os nós da rede comuniquem usando mais de um salto. A esta comunicação de nós em *Piconets* diferentes chamamos *inter-Piconet*.

Na literatura, é possível encontrar rigorosos trabalhos de pesquisa de 2002 a 2006 (Whitaker, Hodge and Chlamtac 2005) introduzindo diferentes protocolos de formação *Scatternet*. Com o crescimento do uso de *Bluetooth*, os protocolos que formam *Scatternet* retornaram ao foco das pesquisas como podemos observar nos trabalhos de (Yu 2010), e (Tahir, Said and Bakhsh 2012a, Tahir, Said and Bakhsh 2012b) para citar alguns. Nenhum dos estudos em *Scatternet* até o momento, consolidou algum padrão, e nenhum produto comercial faz uso desta formação de rede. No contexto de aplicações

de rede social móvel, por exemplo: não encontramos nenhum aplicativo popular, apesar do crescimento do uso de dispositivos móveis. Estes aplicativos, serão um dos principais beneficiados do uso de *Scatternet*. Mas sua implementação continua praticamente nula, boa parte deste motivo, é devido a complexidade de implementar algoritmos *Scatternet*.

O *Bluetooth* usa uma técnica de comunicação, na qual os dispositivos se comunicam, usando canais em sequência de saltos de frequência, coordenada pelo mestre e conhecida por todos participantes da *Piconet*. Para que ocorra conexão, é necessário que os nós conheçam e estejam no mesmo padrão dos saltos. Esta técnica se chama *FHSS*.

O *FHSS* é uma técnica de comunicação comum em redes ponto a ponto *Ad Hoc*. Elas possuem melhor tolerância a colisões e a interferências de obstáculos, possibilitando inclusive algum nível de *QOS* na comunicação. A especificação *Bluetooth* inclui recursos de segurança no nível de ligação com suporte a autenticação, encriptação (Kui and Xiuying 2003). Estes dispositivos possuem também modos específicos de economia de energia. Características fundamentais, devido a necessidade de segurança avançada de algumas aplicações, e os limites de bateria dos dispositivos móveis.

Durante a formação inicial de uma *Piconet*, na fase de descoberta de novos nós, *Descoberta*. O dispositivo mestre busca por dispositivos candidatos a escravos aguardando ser encontrados. Candidatos a mestre e escravo iniciam uma sequência de saltos pseudo aleatórios até que ocorra uma coincidência de frequência. Após esta coincidência de frequências, o candidato a escravo, aguarda por um tempo aleatório para responder ao mestre, a este intervalo chamamos *Backoff*.

Um atraso também ocorre no ingresso de novos nós a uma *Piconet* existente. Durante este processo, a medida que aumentam o número de nós disponíveis para associação, maior o atraso. Este atraso ocorre por causa dos efeitos do *Scheduling*. Durante o *Scheduling*, o mestre efetua o envio de uma mensagem de *pooling* em cada escravo, que só poderá comunicar na *Piconet*, após receber esta mensagem.

Ao avaliar a eficiência da *Scatternet*, é possível notar significativas influencias do *FHSS* nos nós *bridges*, localizados na intercessão das *Piconets*, e responsáveis pela comunicação entre elas. Para que um nó possa atuar como *bridge* é necessário que este pare de comunicar em uma *Piconet* e troque seu padrão de salto para outra. Durante este procedimento, o nó *bridge* entra em um estado chamado *HOLD*. Neste estado observamos um alto custo de energia e tráfego. A troca de mestre leva em consequência, uma troca de padrão de salto, e necessidade constante de sincronismo, coordenadas por um processo chamado *Scheduling*. Estas características fazem com que a localização, volume

e tipos das *bridges*, influenciem diretamente no consumo de energia e comunicação das *Scatternets*.

Em nosso trabalho iremos focar em dois itens relacionados ao uso do *FHSS* pelo *Bluetooth*: a aleatoriedade no processo de descoberta de novos nós, e a influência das *bridges* na performance das *Scatternets*.

1.2 Justificativa

Os efeitos do *FHSS* necessitam ser corretamente identificados. Mostraremos que por usar o *FHSS*, o *Bluetooth* possui uma substancial degradação em seu processo de descoberta de dispositivos, e como este efeito afeta a formação e comunicação de suas rede básica a *Piconet*. De acordo com (Jedda, Jourdan and Zaguia 2010, Jeddah, Zaguia and Jourdan 2009), esta é a principal causa de perda de eficiência nos protocolos de formação de *Scatternets Bluetooth*.

A aleatoriedade no *Discovery* e *Backoff* do *Bluetooth*, é necessária para evitar colisões de pacote durante o sincronismo inicial, mas provoca atraso de segundos nesta conexão. Este atraso prejudica o funcionamento de algumas aplicações: como a entrada em eventos esportivos. Nesta aplicação, torcedores entram nos estádios, sendo identificados e autenticados por um *dongle* biométrico *Bluetooth*. Este caso de uso necessita de *feedback* imediato ao usuário, e o atraso de segundos acumulados na fase inicialização de uma *Piconet* a torna proibitiva.

Uma correta avaliação da influência dos nós *bridges* na eficiência das formações *Scatternet* é necessária. Estes assumem papel de escravo em mais de uma *Piconet*, alternando e coordenando sua sequência de saltos de frequência, com dois ou mais mestres. Esta característica possibilita saltos de comunicação em mais de uma *Piconet*, e um correto entendimento dos efeitos colaterais do uso de *FHSS* pelo *Bluetooth*, são essenciais para geração de modelos *Scatternet* eficientes. O atraso na inicialização da *Piconet*, os gargalos de tráfego e perda desnecessária de energia, são alguns dos problemas a serem observados durante as formações, coordenação e comunicação das *Scatternet*.

Ao estudar o modelo de (Ajmone Marsan, Chiasserini, Nucci, Carello and De Giovanni 2002), observamos nas soluções de suas instâncias, que a topologia gerada por este modelo, possui um número excessivo de *Piconets* e *bridges*, além de outras conexões proibitivas na especificação base do *Bluetooth*. Estes itens provocam perdas de

desempenho e energia, significativos para tornar proibitiva uma aplicação real.

1.3 Motivação

O uso de dispositivos móveis vem ganhando cada vez mais espaço no cotidiano. Um visível crescimento na comercialização e fabricação destes dispositivos vem levando a uma rápida disseminação e adesão em quase todos os níveis de nossa sociedade. Boa parte deste fenômeno comercial e tecnológico, é devido a grande popularização dos *smartphones* e mais recentemente *tablets*. As promessas futuristas em que teríamos eletrodomésticos, *games*, automóveis, roupas e monitoramento de suporte a vida totalmente capazes de comunicação espontâneas já apresentam demanda real. Frente a este quadro, surge cada vez mais a necessidade de simularmos novos cenários, e melhor aproveitar as interfaces *WPAN* já incluídas na maior parte dos dispositivos comercializados.

As demandas de implementação clássicas das *WPANs* são: aplicações de automação doméstica, monitoramento e suporte a vida. Casos de uso mais elaborados de *WPANs* se encontram inexplorados. Como exemplo destas aplicações, temos as redes sociais móveis. Estas redes, possibilitam por exemplo, interação entre os usuários, usando interfaces *Bluetooth* de seus *smartphones*, mesmo que o acesso a internet seja instável ou inexistente. Nestas aplicações o compartilhamento de informações de perfis, trocar arquivos e buscar novos usuários, é possível, mesmo *off-line* com o *webservice* da aplicação.

As *Scatternets Bluetooth*, permitem a expansão dos limites do alcance da conexão *P2P* da interface *Bluetooth*. Podemos utilizar uma aplicação de rede social móvel sobre esta formação para: troca de perfis, arquivos pessoais, sensoramento e contexto. Unindo estratégias de *caches* local dos dispositivos, informações podem ser consumidas entre os usuários, mesmo quando ocorrerem longas janelas sem conexão ao *webservice*.

Aplicações que usam informações de localização, podem obtê-las mesmo quando o sinal de *GPS* não existir (Thongthammachart and Olesen 2003), como no interior de shoppings, condomínios, túneis, cavernas. Nestas aplicações os dispositivos móveis com *Bluetooth*, usam o endereço de um ponto de acesso para identificar e compartilhar sua provável posição, com uma margens de erro igual ao alcance da comunicação, como por exemplo na versão 1.2, 10 metros. Estas aplicações ainda inexistentes, são um motivador de uso das interfaces *Bluetooth* para formar redes de mais de um salto como as *Scatternets*.

Ao considerarmos a aleatoriedade e atraso de conexão durante o *Discovery* do *Bluetooth*, ao iniciar uma aplicação, alguns dispositivos podem não conectar e até mesmo não serem encontrados. Ao tratarmos este cenário em uma aplicação que solicita conexões *Bluetooth* simultâneas, surgem questões como: Quem tentou conectar primeiro? Quem conseguiu autenticar? Em qual ordem?

Um exemplo, é o caso de uso de autenticação de uma fila de usuários usando dispositivos biométricos *Bluetooth*. Neste caso, as questões anteriores não podem ser respondidas somente com uso de uma tecnologia. Soluções de contorno para este tipo de cenários foram estudados em (Woodings, Joos, Clifton and Knutson 2002), que recomenda o uso do infra-vermelho para identificar os elementos da comunicação, e (Salminen, Hosio and Rieki 2006) que recomenda o uso do RFID para a identificação por proximidade. Ambos os casos são usados para lidar com o problema do atraso no *Discovery* de nós *Bluetooth*. Isto mostra a relevância, de simular cenários alternativos ao atual processo de conexão de dispositivos *Bluetooth*, identificando os efeitos do *FHSS* e sua correta inserção nos modelos existentes.

1.4 Contribuições

Nossas contribuições neste trabalho são:

- Nós avançamos o trabalho de (Pettarin, Pietracaprina and Pucci 2009), que modela o *Bluetooth* como um grafo, e acrescentamos ao modelo:
 - $FHS()$, que representa a sequência de saltos de frequência,
 - $Disc()$, que representa o processo de descoberta de nós.

Estas duas funções adicionadas ao modelo de (Pettarin, Pietracaprina and Pucci 2009) caracterizam o *Bluetooth* como um grafo dinâmico.

- Um novo Modelo de Otimização para formação de *Scatternet* estática. Nós ajustamos o modelo clássico de formação *Scatternet* estática de (Ajmone Marsan, Chiasserini, Nucci, Carello and De Giovanni 2002), penalizando as *bridges* e adicionando novas restrições. Este novo modelo resultante, produz como solução, topologias de *Scatternet* mais coerentes com as encontradas pelo modelo dinâmico de formação *Scatternet* (Law, Mehta and Siu 2002), conhecido pelo seu algoritmo distribuído de baixa complexidade.

Nós simulamos a formação de *Piconets* para analisar o atraso gerado pela: descoberta de novos nós para formar uma nova *Piconet* e na entrada de novos nós escravos a uma *Piconet* existente. E validamos nosso modelo estático de *Scatternet*, verificando se a topologia encontrada possui topologia aproximada da encontrada na simulação do modelo dinâmico *Scatternet*, algoritmo (Law, Mehta and Siu 2002).

Esta dissertação resultou nas seguintes publicações:

- SOARES, C. M. F. ; OLIVEIRA, R. A. R. ; SANTOS, Haroldo Gambini ; FRERY, Alejandro C . Static Bluetooth Scatternet Formation Models: The Impact of FHSS. In: The Ninth Advanced International Conference on Telecommunications, 2013, Roma, Itália
- SOARES, C. M. F. ; OLIVEIRA, R. A. R. ; SANTOS, Haroldo Gambini ; FRERY, Alejandro C . Characterization of FHSS in Wireless Personal Area Networks. In: The 22nd Wireless and Optical Communication Conference, 2013, Chongqing, China

1.5 Organização do Texto

O texto desta dissertação encontra-se organizado da seguinte forma:

No Capítulo 2, revisamos os trabalhos relacionados ao nosso estudo.

No Capítulo 3, apresentamos a técnica de comunicação *FHSS* e as tecnologias disponíveis para formar *WPAN*.

No Capítulo 4, exploramos a *Piconet Bluetooth*, detalhando o processo de *Discovery* de dispositivos, a formação de link, a comunicação e *Scheduling intra-Piconet*.

No Capítulo 5, nós introduzimos o modelo dinâmico e estático de *Scatternet*, e avaliamos a influência das *bridges* nestes modelos.

No Capítulo 6, avançamos o trabalho de (Pettarin, Pietracaprina and Pucci 2009) que modela o *Bluetooth* como um grafo, e acrescentamos ao modelo, dois novos procedimentos. O *FHS()*, que representa a sequência de saltos de frequência, e o *Disc()*, que representa o processo de descoberta de nós. Estas adições ao modelo de (Pettarin, Pietracaprina and Pucci 2009) passam a caracterizar o *Bluetooth* como um grafo dinâmico.

No Capítulo 7, demonstramos os efeitos do *FHSS* usando simulador, e mostramos como resultados de nossos experimentos o atraso e variações de tempo para convergência durante a fase de descoberta e formação de novas *Piconets* e *Scatternets*.

No Capítulo 8, avaliamos a eficiência de uma topologia *Scatternet* e a influência das *bridges*. Nós simulamos a formação fornecida pelo algoritmo discutido por (Law, Mehta and Siu 2002). E para obter soluções a partir do modelo de (Ajmone Marsan, Chiasserini, Nucci, Carello and De Giovanni 2002), com uma topologia similar a obtida pelo algoritmo de baixa complexidade (Law, Mehta and Siu 2002). Nós adicionamos penalidades para as *bridges*, e adicionamos duas novas restrições ao modelo original.

No Capítulo 9, as conclusões finais sobre o assunto são discutidas e as diretrizes futuras são definidas.

Capítulo 2

Revisão Bibliográfica

Em (Tahir, Said and Bakhsh 2012b), o protocolo de roteamento de *Scatternet* SROP é proposto. Ele procura estabelecer rotas com o mínimo de saltos entre origem e destino. Para que isto ocorra, o mestre procura todos os caminhos possíveis por meio de *flooding*. Sua principal vantagem é, uma vez que os mestres conhecem todos os links, este pode usar esta informação caso algum dispositivo queira comunicar com outra *Piconet*. De acordo com o autor, ocorre uma economia de tempo em procurar todas as rotas por *flooding*, e esta performance é avaliada no simulador ns-2 (NS2 Last Visited in 11/02/2013).

Em (Pettarin, Pietracaprina and Pucci 2009), estuda a expansão e diâmetro de uma família de subgrafos do grafo geométrico aleatório, que representa a topologia induzida pela fase de descoberta de dispositivos na rede *Bluetooth*. Este trabalho propõe para topologia o seguinte grafo aleatório. Os dispositivos são representados por n nós, cuja coordenadas são aleatoriamente escolhidas dentro do quadrado $[0, 1]^2$; cada nó seleciona $c(n)$ vizinhos entre todos os nós vizíveis, que estão, entre todos dentro da distância Euclidiana $r(n)$, onde $r(n)$ representa o alcance de visibilidade, que é assumido ser o mesmo para todos os dispositivos. O grafo resultante, chamado de $BT(r(n), c(n))$, é aquele onde existe uma aresta não direcionada para cada par de nós vizinhos. Este trabalho não explora as restrições necessárias para que seja formada uma rede *Bluetooth*, como as *Piconets* e *Scatternets*.

Em (Law, Mehta and Siu 2002) um novo algoritmo dinâmico de formação de *scatternet* é introduzido. O algoritmo tem $O(\log n)$ de complexidade de tempo e $O(n)$ de complexidade e mensagem. Em sua organização os dispositivos são separados por componentes. A base do protocolo é separar o processo de eleição de líder por conjuntos

chamados componentes. Um *device*, *Piconet* e *Scatternet* podem ser componentes, e o algoritmo coordena a fusão, movimentação e busca por novos nós durante a operação de uma *Scatternet*.

Em (Ajmone Marsan, Chiasserini, Nucci, Carello and De Giovanni 2002), a formação de uma *Scatternet* é representada por programação matemática. Restrições são propostas em uma formulação min-max gerando um problema de otimização, resolvido de maneira centralizada. Segundo o artigo, o número de parâmetros do modelo do *Bluetooth* é grande o suficiente para tornar uma abordagem distribuída proibitiva. O objetivo deste modelo é minimizar o tráfego nos nós que são sujeitos a maior congestionamento e consumo de energia, como os mestres e *bridges*, respeitando as restrições após a convergência total da *Scatternet*. Porém este artigo não considera os efeitos do *FHSS*. Em seu modelo de restrições, a pseudo aleatoriedade da fase de *discovery* não é incluída. A influência dos atrasos ocasionados pelo esforço envolvido na troca de canais durante o tráfego entre *Piconets* coordenado por um nó *Bridge*. As influências deste procedimento de coordenação durante um dos modos de economia de energia do *Bluetooth* chamado *HOLD*, também não é abordado. A abordagem deste estudo está limitado as regras estabelecidas a cada dispositivo, não sendo atribuído maiores penalidades para os nós *bridge*.

Em (Chiasserini, Ajmone Marsan, Baralis and Garza 2003) as restrições do modelo centralizado de (Ajmone Marsan, Chiasserini, Nucci, Carello and De Giovanni 2002) são complementadas com uma discussão e proposição de algoritmo distribuído, na formação de *Scatternet*, incluindo rotinas de inserção e remoção de nó.

Em sua pesquisa (Jedda, Jourdan and Zaguia 2010) analisa os impactos das mudanças dos parâmetros, relacionados ao *FHSS* no *Bluetooth*, durante a formação de uma *Scatternets*. Por meio de simulação em NS2 com UCBT, são mostradas as diferenças nos tempos de execução dos algoritmo *Bluetooth Scatternet Formation (BSF)*, nas versões do *Bluetooth* 1.1 e 1.2. Alguns resultados mostraram que as mudanças em alguns parâmetros do *FHSS* são mais significativas nos algoritmos estáticos, que fazem uso da estratégia para formação, em que os nós alternam entre ser encontrado e encontrar, *ALTERNATE*, que nos dinâmicos como o (Law, Mehta and Siu 2002).

O estudo de (Salonidis, Bhagwat, Tassiulas and Lamaire 2001) introduz o protocolo de construção de topologia Bluetooth (BTCP) para formação de *Scatternets*. Os autores definem regras necessárias para a eleição de um líder e a técnica de alternar o estado de um nó entre encontrar e ser encontrado para formação de rede. Este estudo descreve

como motivação, o uso de conexão entre os usuários em uma conferência, mostrando a importância de minimizar o tempo de *setup* tornando tolerável seu uso.

O tempo total de conexão entre um mestre e escravo foi estudado por (Peterson, Baldwin and Kharoufeh 2006), que mostra que os parâmetros definidos na especificação, tem valores acima do necessário. Contudo, ele não considera a diversidade de elementos envolvidos na conexão.

O software CenceMe é apresentado em (Miluzzo, Lane, Fodor, Peterson, Lu, Musolesi, Eisenman, Zheng and Campbell 2008). Por meio de sensores de voz, movimento e presença, o software permite que usuários se encontrem por *Bluetooth* e troquem informações de sensoriamento, como por exemplo: se estão dançando, dialogando ou sozinhos. Também é possível traçar perfis baseados no sensoriamento contínuo por meio dos cruzamentos de dados coletados a respeito de seu comportamento. O artigo não entra muito em detalhes sobre o *setup* das *Scatternets* nem levantou medidas de desempenho durante a fase de formação da *Piconet Bluetooth*. O CenceMe mostra ser um aplicativo que faz uso do poder das redes sociais aliados a popularização e proximidade permitidas pelos dispositivos móveis.

Capítulo 3

Wireless Personal Area Network

3.1 Introdução

Neste capítulo apresentaremos as *Wireless Personal Area Networks (WPAN)*. Elas são redes sem fio *Ad Hoc* formadas por dispositivos de baixo custo, consumo de energia e vazão de dados, que criam redes de alcances curtos, normalmente no próprio espaço de trabalho do usuário. As *WPAN* mais conhecidas são a *Bluetooth IEEE Std 802.15.1* e *ZigBee IEEE Std 802.15.4* e usam uma técnica de comunicação em saltos de frequência chamada *FHSS*.

3.2 FHSS (Frequency-hopping spread spectrum)

Apesar de não ser exclusiva da *WPAN*, o *FHSS* é a técnica de comunicação escolhida para comunicação destas redes ponto a ponto *Ad Hoc*.

Redes que usam *FHSS*, comunicam em um canal a cada *slot* de tempo, como pode ser visto no exemplo do *Bluetooth* ilustrada pela Figura 3.1. Apesar das aplicações da *WPAN* serem inicialmente desenhadas para curto alcance, as redes *FHSS* permitem flexibilidade de uso em diversas distâncias. Possuem melhor tolerância aos ruídos das redes vizinhas, com um *Quality of Service (QoS)* e segurança evoluída em comparação com *Wi-Fi*. Nos ataques a segurança destas redes, a captura de dados com *sniffers* não é trivial, devido necessidade de predição do padrão de salto pseudo aleatório pelo atacante.

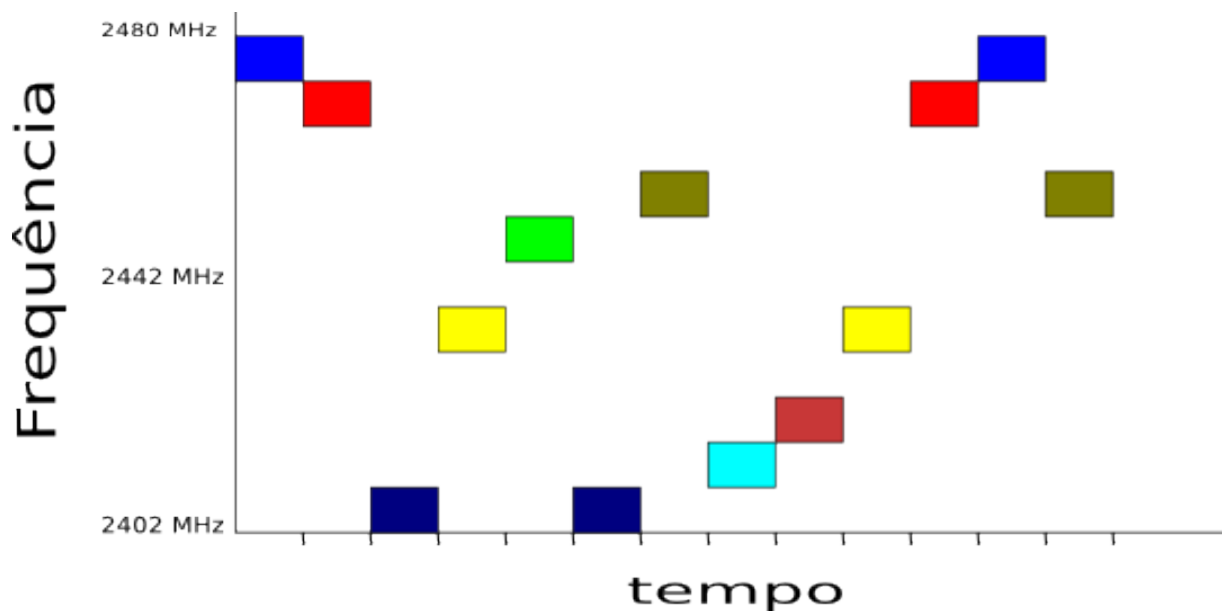


Figura 3.1: FHSS no Bluetooth

3.3 Bluetooth

O *IEEE 802.15.1* ou *Bluetooth* é uma tecnologia *WPAN*. Elas usam rádio frequência, para criar redes de curto alcance, baixo custo e baixo consumo de energia. Atualmente na versão 4.0. Seu mercado é dirigido pelo *Bluetooth Special Interest Group (SIG)*, e suas interfaces, são encontradas na quase totalidade dos dispositivos móveis recentemente comercializados.

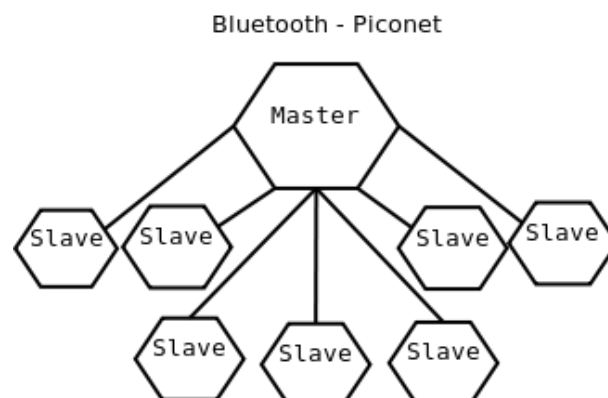


Figura 3.2: Piconet

Diversos acessórios e softwares disponíveis nesta tecnologia, são fabricados e projetados para substituir definitivamente os tradicionais cabos de dados. Opera na *ISM*

2,4 GHz com taxas de transmissão de dados variando de acordo com a tabela 3.1. O *Bluetooth* usa 79 canais, nas radio frequências de (2402 – 2480)MHz em modulação *GFSK*.

Tabela 3.1: Taxas de Transmissão do *Bluetooth*

Versão	Taxa de transmissão
até 1.2	1 Mbp/s
2.0 com <i>EDR</i>	3 Mbp/s
3.0	24 Mbp/s

Sua rede básica, possui característica ponto multi-ponto, e é chamada *Piconet*. Criado com foco em conexões curtas fazendo uso de pouca energia. Sua comunicação usa salto de frequência *FHSS*, em uma de suas variações chamada *Adaptive Frequency-hopping* (AFH). Possui uma taxa de 1600 saltos por segundo, com um alcance que varia de 1 a 100 metros, de acordo com a classe do dispositivo *Bluetooth*, como detalhado na tabela 3.2.

Cada dispositivo *Bluetooth* possui um relógio nativo, que é usado para sincronização na comunicação, e um endereço de rede de 48-bits.

Tabela 3.2: Alcances do *Bluetooth*

Classe	Potência	Alcance
Class 1	100 mW(20 dBm)	até 100 metros
Class 2	2.5 mW(4 dBm)	até 10 metros
Class 3	1 mW(0 dBm)	1 metro

3.4 ZigBee

Desenvolvida com foco em baixo custo e baixa potência, as redes *Zigbee* foram criadas para aplicações de curto alcance, em ambientes que exijam pouca ou nenhuma configuração do usuário na infra estrutura e pontos de acesso.

Possui nativamente características de conexão *Mesh* e *Multi-hop*, Figura 3.3, o que possibilita em sua formação básica de rede, a repetição de mensagens entre os nós e

sub-redes.

ZigBee ou *IEEE Std 802.15.4* é derivado do *Bluetooth*. Homologado em maio de 2003, seus casos de uso mais comuns são: redes de sensores, monitoramento, controle e automação doméstica. Já pode ser encontrado em controles remotos de *hardwares* de entretenimento, TV e eletrodomésticos. Provê a uma formação de rede *WPAN* de baixa complexidade, baixo custo e baixo consumo de energia. Foi desenvolvido para trabalhar em dispositivos de baixo custo comercial e portáteis. Possui características essenciais em aplicações que exista necessidade de baixa latência e consumo de energia, com ênfase na maximização do tempo de uso das baterias.

Os dispositivos *ZigBee* são fabricados em três perfis:

- O *ZigBee coordinator (ZC)* - Responsável por assumir o *root* da árvore *mesh* e pelo contato com outras redes. Sendo mandatório a presença de um *ZC* por rede, Figura 3.3, pois este inicia a árvore e armazena as informações básicas da comunicação da rede e chaves de segurança.
- *ZigBee Router (ZR)* - Atua como um nó comum *ZigBee* e atua como roteador entre os dispositivos.
- *ZigBee End Device (ZED)* possui apenas as funcionalidade de comunicação com *ZC* ou *ZR*, e não pode retransmitir nenhum pacote de outros dispositivos. Contém recursos apenas para comunicar com um nó pai (ou coordenador ou um roteador). Estas características, permitem que dispositivos *ZC* possam permanecer em *sleep* maior tempo possível. Este procedimento faz com que o nó adormeça uma quantidade significativa de tempo, possibilitando uma maior sobrevida de bateria. Em resumo os nós *ZED*, possuem os dispositivos mais simples em hardware, necessitando de menor quantidade de memória e um baixo custo de fabricação.

3.5 Conclusão

Redes *WPAN*, que fazem uso do *FHSS* para se comunicar, possuem baixo consumo de energia, flexibilidade de aplicações e segurança mais robusta. Otimizam o uso do espectro de frequências, e possuem baixo *overlap* de canal. Os dispositivos que fazem uso do *FHSS*, são construídos com foco em mobilidade, e comunicam por meio de suas interfaces

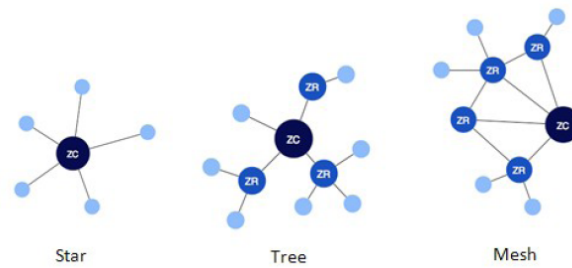


Figura 3.3: Zigbee

de baixa potência e consumo de energia. Estas características são fundamentais, devido a autonomia das baterias, e justifica aplicações projetadas para *WPAN*.

O *Bluetooth* é a tecnologia mais popular entre as *WPAN*. Seu foco inicial é formar redes de curto alcance, baixo custo e baixo consumo, e sua rede base, a *Piconet*, pode ser estendida em redes maiores chamadas *Scatternets*.

No próximo capítulo abordaremos em detalhes a formação de rede básica do *Bluetooth*, a *Piconet*, apresentando suas características e processo de conexão.

Capítulo 4

Piconet

4.1 Introdução

Nesta capítulo apresentaremos a formação de rede básica do *Bluetooth*. Tal rede é nomeada *Piconet*, e para que exista comunicação no *Bluetooth* é necessário primeiramente formá-la. Em seu processo de inicialização, dispositivos assumem papéis de mestre e escravo, em conexões ponto a ponto *Ad Hoc*.

4.2 Características

A *Piconet* de topologia estrela, possui obrigatoriamente um único mestre. E conectados a ele, um limite de 7 escravos em modo *ACTIVE* Figura 3.2. O *ACTIVE* é o modo convencional onde o dispositivo escravo se encontra sincronizado e comunicando ativamente, o modo *HOLD* é um modo de economia de energia; e no modo *PARK* o escravo permanece sincronizado com a *Piconet*, mas não participa do tráfego; somente existe a manutenção do sincronismo com o mestre com escutas ocasionais de mensagens de sincronismo. Por esta característica, o modo *PARK* permite que escravos adicionais ao número de 7 possam existira na *Piconet*.

Em sua comunicação, o fluxo de pacotes centralizado no mestre em uma topologia ponto-multi ponto. Cada *Piconet* compartilha um mesmo padrão de salto de frequência coordenada pelo mestre e seguida pelos escravos. Para que isso ocorra o relógio e o endereço de rede do mestre são usados para compor uma sequência de saltos de frequência

de forma pseudo-aleatória.

A Figura 4.1 (Farinaz Edalat n.d.) mostra uma *piconet*. O vértice vermelho, é o mestre da rede, os nós verdes são os escravos em comunicação, modo *ACTIVE*, os nós rosa dispositivos em modo *PARK*, o nó em azul um dispositivo em modo de economia de energia *HOLD* e os nós em amarelos, dispositivos fora do alcance.

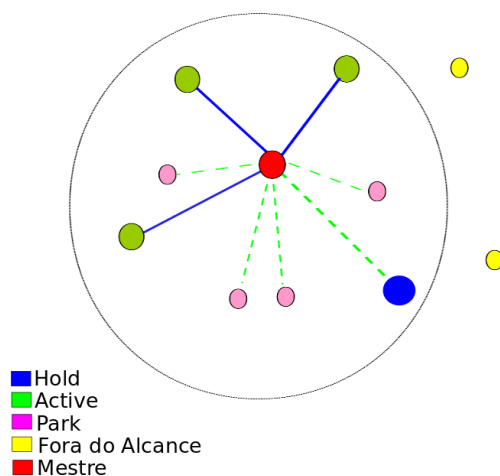


Figura 4.1: Piconet Bluetooth

Sua característica topológica, unida ao salto de frequência, diminui a interferência de outras redes, e permite conexões sem visada direta. Típicas de ambientes em que barreiras físicas são uma constante. Isto garante uma maior flexibilidade de uso em variados cenários de distâncias e direções.

Para estender os limites de alcance da *Piconet* e permitir comunicação de mais de um salto, uma nova formação é definida, a *Scatternet*.

Scatternet em sua formação, usam nós *Bridges* para comunicação entre duas ou mais *Piconets*.

Processo de formação de uma nova Piconet Bluetooth

Para formar uma *Piconet*, são necessárias duas fases distintas *Descoberta* de dispositivos e a formação de *link*. Na Figura 4.2 podemos visualizar a máquina de estados *Bluetooth* (Farinaz Edalat n.d.).

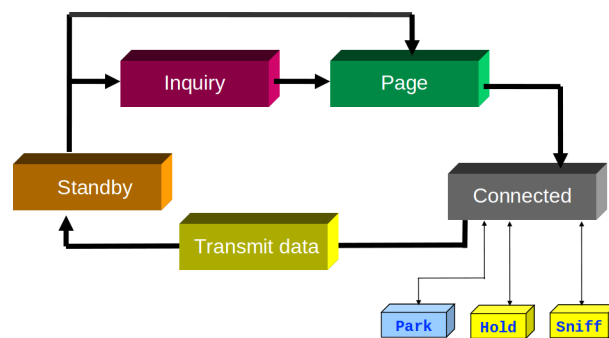


Figura 4.2: Máquina de Estado *BlueTooth*

Descoberta de nós

Na *Descoberta*, o dispositivo que assumirá o papel de mestre sai do estado de *STANDBY* e entra em estado de *INQUIRY*. E buscam por dispositivos escravos aguardando para serem encontrados em um estado chamado *INQUIRY SCAN*.

O dispositivo que procura envia pacotes *ID* contendo um identificador chamado *Inquiry Access Code (IAC)*. O *IAC* define o tipo de dispositivo *INQUIRY SCAN* que o dispositivo em *INQUIRY* procura. Por exemplo, um *smartphone* procurando por um *Human Interface Devices (HID)*. Neste exemplo somente viva voz automotivos e fones de ouvido responderiam.

Inquiry

Durante o *INQUIRY*, o candidato a mestre dispara pacotes *ID*. Estes pacotes contém o *IAC* e são enviados por *Broadcast*, nas 32 frequências divididas em dois *trains* de 16, das 79 definidas pela especificação *Bluetooth*. A esta sequência de saltos de frequências chamada de *Inquiry Hopping Sequence (IHS)*. Ela ocorre de forma pseudo aleatória com cálculo derivado do relógio do dispositivo candidato a mestre.

Os dispositivos em *INQUIRY SCAN*, escutam os *broadcasts ID*, nas mesmas 32 frequências, saltando em uma sequência pseudo aleatória derivada do seu *clock*.

Para que aumente a probabilidade do dispositivo receber um *ID*. Os dispositivos em *INQUIRY* saltam de frequência em um *time slot* de 312, 5 μ S mais rápido, que o padrão definido pelo *Bluetooth* 625 μ S, e usado pelos dispositivos em *INQUIRY SCAN*.

No processo de formação de uma *Piconet*, um dispositivo que deseja ser encontrado

em *INQUIRY SCAN* pode dar *timeout*, após aguardar um *ID* do candidato a mestre, sem resultado. Ele então, retorna ao modo *STANDBY* por um determinado tempo, e após um tempo aleatório retorna ao *INQUIRY SCAN*. Possibilitando que seja encontrado e efetue o restante dos procedimentos para conexão.

Nas versões do *Bluetooth* anteriores a 1.2, após receber um *ID* o dispositivo candidato a escravo assume um estado chamado *INQUIRY RESPONSE*. Aguarda um tempo aleatório para responder chamado de *Backoff*. Aguarda por um novo *ID* para só então enviar seu endereço de rede e clock em resposta, em um pacote chamado *Frequency Hopping Synchronization (FHS)*. Após este processo o candidato a escravo assume um estado chamado *PAGE SCAN*.

Nas versões do *Bluetooth* a partir da 1.2, ao receber um *ID* o dispositivo candidato a escravo entra em estado de *INQUIRY RESPONSE*. Envia em resposta um *FHS* em uma sequência de frequências chamada *Inquiry Response Hopping Sequence (IRHS)*. O *IRHS* cobre as 32 frequências de resposta que correspondem a *IHS* corrente. Após este procedimento aguarda um tempo de *Backoff*, e vai ao estado de *PAGE SCAN*.

O candidato a mestre ao receber o *FHS* assume o estado de *PAGE*. Ele usa as informações recebidas pelo *FHS* para sincronização e conexão com os nós candidatos a escravo, já descobertos e em estado de *PAGE SCAN*.

Backoff

Durante o *Backoff*, o dispositivo encontrado fica em espera por um valor de *Time slots* aleatório entre 0 e 639,375 *ms*.

Este estado é configurado após o dispositivo receber um *ID* do candidato a mestre. Seu objetivo é minimizar colisões de pacotes de resposta ao candidato a mestre.

Nas versões do *Bluetooth* anteriores a 1.2, quando encerrado o tempo de *BackOff*, o dispositivo aguarda um novo *ID* em estado *INQUIRY RESPONSE*. Somente após este tempo, ele responde ao candidato a mestre o *FHS*, enviando seu endereço e relógio.

Formação de link

Paging

Em estado de *PAGE*, o candidato a mestre transmite um pacote chamado *Device Access Code packet (DAC)*, ouvido unicamente pelo candidato a escravo em *PAGE SCAN*. Neste instante os clocks dos candidatados ainda não estão sincronizados, e o candidato a mestre transmite o *DAC* em várias frequências distintas estimada pelo endereço de rede do candidato a escravo.

Ao receber o *DAC*, o candidato a escravo, vai ao estado de *SLAVE RESPONSE* e responde com um *DAC* em uma sequência chamada *page response hopping sequence*. Ao receber a resposta, o candidato a mestre entra em um estado chamado *MASTER RESPONSE* e envia um *FHS* em uma sequência chamada *page hopping sequence*. O candidato a escravo usa esta informação para determinar a sequência de saltos de frequência do candidato a mestre.

Neste instante, ele assume efetivamente o papel de escravo e responde com outro *DAC*. Ao receber o *DAC* do escravo, o dispositivo em *PAGE* assume papel de mestre na conexão. Neste instante ele passa a sincronizar a recém formada *Piconet*, e pode também negociar trocas de papéis mestre e escravo.

Um detalhe é destaque, ao avaliarmos o tempo dos estados de conexão do *Bluetooth*. Em comparação com o *INQUIRY* o *PAGE* leva μS em vez de segundos, pois não usa de processos aleatórios em suas etapas.

Conectados

Após completar o processo de *PAGE*, ocorre a formação da nova *Piconet* e os dispositivos ganham status de conectados. Os primeiros pacotes trocados neste estado de conexão, contém mensagens de controle utilizadas para estabelecer links de tipo: *Synchronous Connection Oriented Link (SCO)* ou *Asynchronous Connection-oriented Logical transport (ACL)*. Uma vez estabelecidos, ocorre a troca de informações efetiva entre os dispositivos.

O *SCO* é utilizado em aplicações multimídia, no envio contínuo de dados, como voz e áudio; e não possibilita retransmissão de pacotes de dados perdidos.

O *ACL* é usado para aplicações que envolvam transferência de arquivos. A velocidade

de transmissão de dados é de até 721 *Kbps*.

Em estado de conexão, um dispositivo pode se encontrar nos modos: *ACTIVE*, *SNIFF*, *HOLD* e *PARK*.

- *ACTIVE*: modo convencional onde o dispositivo se encontra sincronizado e comunicando ativamente na sequência de frequência da *Piconet*.
- *SNIFF*: o escravo escuta o tráfego *Piconet* a uma taxa de tráfego de dados reduzida, poupando energia;
- *HOLD*: modo intermediário de economia de energia;
- *PARK*: o escravo permanece sincronizado com a *Piconet*, mas não participa do tráfego; somente existe a manutenção do sincronismo com o mestre, por meio de escutas ocasionais de mensagens de sincronismo. Este é o modo de maior economia de energia.

A Figura 4.3 mostra os estados até a formação da nova *Piconet* nas versões a partir da 1.2 do *Bluetooth*.

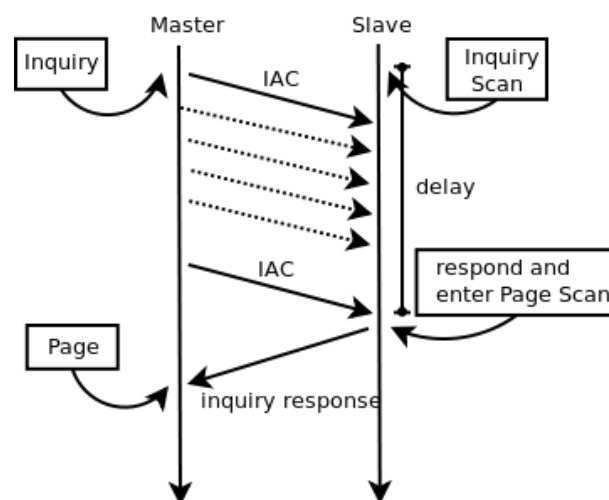


Figura 4.3: Estados até a formação de nova *Piconet*

Comunicação Intra-Piconet

Na comunicação *Intra-Piconet*, um processo de *Scheduling* é necessário. No *Scheduling* o mestre efetua um *polling* em cada escravo enviando pacotes de *POLL*, e somente após

o recebimento deste pacote, o escravo pode enviar ou receber mensagem na *Piconet*.

A ordem que estes escravos recebem este pacote é chamada: *Polling Cycle*, e ela também determina os *slots* que serão usados durante a comunicação. Se um escravo não responde a estes pacotes por um determinado número de *slots*, então ambos os dispositivos devem restabelecer a conexão novamente por *paging*.

Intra-Piconet Scheduling

O *Scheduling*, assim como as *Scatternets*, não recebem maiores detalhes na especificação do *Bluetooth*, ele é coordenado por um algoritmo, que define em qual sequência de escravos o mestre irá fazer *pooling*.

A composição de algoritmos *Scheduling*, na comunicação *Intra-piconet* considera os seguintes itens (Whitaker, Hodge and Chlamtac 2005):

- **atividade do escravo:** diminuição de *Pollings* e mesmo anulação dos escravos que não tem dados para transmitir;
- **priorização:** dará preferência a escravos de acordo com critérios de importâncias;
- **eficiência:** minimizar o atraso para escravos que esperam para transmitir dados.

A Tabela 4.1 resume os tipos e características dos algoritmos intra-piconet scheduling.

Ingresso de novos escravos a uma Piconet existente

No ingresso de novos escravos a uma *Piconet*, o mestre volta ao estado de *INQUIRY* para busca de novos dispositivos. Este procedimento faz com que os nós escravos já conectados a este mestre, saiam do modo *ACTIVE* e entrem em modo *HOLD*. Neste instante esperam, até que o o mestre saia do estado de *INQUIRY*, por um *Polling*. O tempo desta operação, aumenta em função do crescimento de dispositivos candidatos a escravos. Este atraso na conexão de novos nós a *Piconet*, se deve a soma de operações de: *Scheduling* e *Discovery* de dispositivos.

Tabela 4.1: Algoritmos *Intra-piconet Scheduling*

Algoritmo	Característica
<i>Round Robin</i> Puro	Escravos recebem <i>Polling</i> em ordem fixa de forma cíclica
<i>Round Robin</i> Exaustivo	Ordem fixa, e o mestre não troca de escravo até esvaziar a fila
<i>Round Robin</i> Estendido	Considera o histórico de tráfego para priorização, <i>polling</i> de forma pseudo aleatória
<i>Efficient Duple Cycle</i>	Ocorre uma separação no tratamento entre up-link e down-link
<i>Adaptive Flow</i>	A taxa de <i>polling</i> depende do fluxo de dados dos escravos
<i>Predictive fair Polling</i>	<i>Polling</i> baseado em estimativas aproximadas de tráfego
<i>Priority and K fairness</i>	É tratada a priorização de transmissão para quem tem dados a transmitir na fila
<i>Adaptive Capacity Rate</i>	Gerência da fila é baseada na capacidade
<i>Packet type Priority</i>	Aborda a diferença de prioridade por pacote de dados e voz
<i>Look Ahead</i>	Usa conceito de <i>Bin Packing</i> para o problema do <i>Scheduling</i>
<i>Bluetooth Interference Aware</i>	Tenta minimizar os efeitos de interferências

4.3 Conclusão

As *Piconets* suprem a necessidades básicas de um rede *Bluetooth*, mas para que o limite desta comunicação seja expandido, é necessário formar *Scattenets*. Elas são coleções de *Piconet* formadas dinamicamente, e possibilitam que os dispositivos comuniquem usando mais de um salto. No próximo capítulo abordamos em detalhes os modelos que formam estas redes.

Capítulo 5

Scatternet

5.1 Introdução

Neste capítulo abordaremos as *Scatternets*. Elas são coleções de *piconets* formadas espontaneamente sem necessidade de infraestrutura fixa. São redes dinâmicas, que possibilitam comunicação em cenários de mais de um salto.

5.2 Características das Scatternets

Na Figura 5.1 (academic.ru Last Visited in 20/05/2013) temos um exemplo de uma topologia *Scatternet* típica. Elas estendem os limites de uma *Piconet Bluetooth*, em que os dispositivos comunicam somente no alcance de um dispositivo mestre.

Nos cenários em que observamos um número maior que 8 dispositivos tentando conexão, e nem todos os dispositivos que querem conectar estão ao alcance de ao menos um membro, temos a *Scatternet* como solução (Whitaker, Hodge and Chlamtac 2005). Seu roteamento e coordenação é mais complexa, pois existe a necessidade de cruzar múltiplas *Piconets*, em busca do destino e de lidar com múltiplos caminhos alternativos e ciclos.

As *Scatternets* rompem os limites centralizados da *Piconet* tornando possível formações *mesh*. Sua formação não recebe maiores detalhes da especificação do *Bluetooth*, deixando espaço para proposições alternativas. Elas são formadas por algoritmos distribuídos, possuindo cada um diferentes resultados de topologia e otimização. Suas

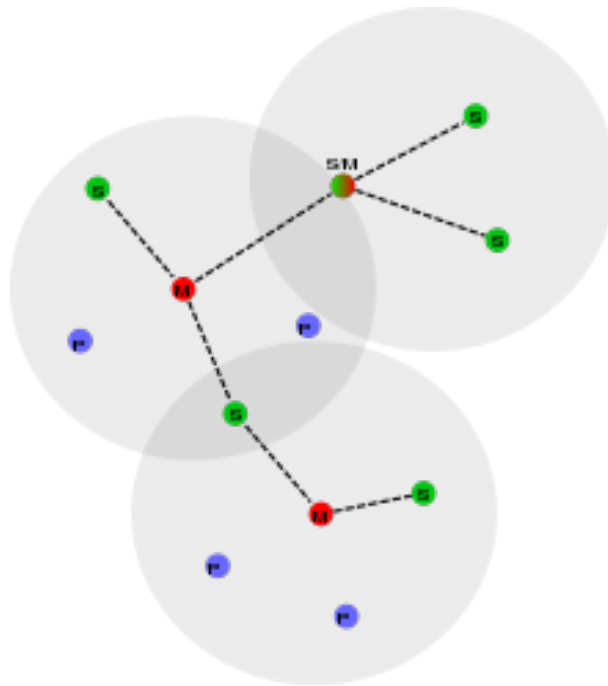


Figura 5.1: Scatternet Bluetooth

características topológicas influem diretamente no fluxo de dados da rede e perda de energia nos dispositivos.

Muitos dos algoritmos tradicionais de formação de redes e roteamento sofrem influência em seu desempenho quando adaptados e usados em redes que possuem, em sua *baseband* o *FHSS*, como no caso das *Scatternets Bluetooth*. Para isso necessitam receber um novo desenho, que consegue lidar com novas necessidades. A distribuição dos dispositivos no espaço da aplicação, a escalabilidade da solução, os diferentes papéis assumidos, escravo mestre bridge, e as limitações energéticas e tráfego, são alguns exemplos dos desafios na criação de modelos *Scatternet*. Por isso novos mecanismos de controle e coordenação tem que ser inseridos nos algoritmos de formação *Scatternet*.

Para estudar e implementar as *Scatternets* temos os modelos dinâmicos e estáticos.

5.3 Modelos Dinâmicos

Modelos dinâmicos de Scatternet são protocolos, e seus algoritmos distribuídos usam a seguinte heurística para geração de topologias eficientes (Whitaker, Hodge and Chlamtac 2005).

- Dispositivos *bridge* nunca devem ser mestres. Isto reduz a carga de *Scheduler* dos mestres, que passam somente a considerar a comunicação *intra-Piconet*;
- Restringir o número de *Piconets*. Isto reduz o número de potenciais conflitos *inter-Piconet* nos dispositivos *bridges*, mas limita o potencial de rotas alternativas;
- O número de *Piconets* deve ser a menor possível. Isto reduz o número de canais a serem usados e com isso potenciais interferências;
- *Piconets* não devem ser ligadas a mais que uma *bridge*. Isto minimiza o esforço de coordenação necessário pelo *Scheduling*;
- Um dispositivo deve participar em um menor número de *Piconets* possível. Isto faz com diminua o montante de *inter-Piconet Scheduling* no dispositivo.

Para representar uma *Scatternet* o grafo de visibilidade é usado. Ele demonstra todas as possíveis conexões entre os dispositivos no alcance.

Alguns algoritmos *Scatternets* usam grafos para gerar diferentes topologias:

- **Trees:**
 - representado por um grafo conectado sem ciclos;
 - mínimo de arestas para conexão;
 - não existem rotas alternativas de busca entre os nós;
 - mais suscetível a quebras de links na falha ou falta de energia de um dispositivo;
 - rotas mais simplificadas por existir somente um caminho entre os nós;
 - por possuir um roteamento mais simplificado, sofrem menos os efeitos do *FHSS* reduzindo a contenção nos *slots* de transmissão em *TDD* (Whitaker, Hodge and Chlamtac 2005);
 - um número mínimo de *Piconets* é esperado, garantindo que as *bridges* participem de no máximo duas *Piconets*
- **UDG:**
 - Uma aresta é definida se sua distância euclidiana é maior que um;
 - O grafo é formado de acordo com a proximidade dos nós;

- **1-factor**

- um perfeito casamento é esperado onde $(n/2)$ é esperado onde n é o número de vértices;
- uma aresta é sempre um conjunto mestre escravo;

- **Ring**

- *Scatternets* são chamadas de *Bluerings* (Whitaker, Hodge and Chlamtac 2005);
- cada dispositivo pertence a duas *Piconet* e possui dos *links* no total;
- cada dispositivo é um mestre e escravo ao mesmo tempo ;
- suporta o máximo de $\frac{n}{2}$ *links* ativos;
- rota é simplificada porque os pacotes são simplesmente encaminhados;
- um anel grande pode receber um grande atraso de resincronização, de acordo com o número de *bridges*.

5.3.1 Protocolos Scatternet

Os protocolos *Scatternet* são tratados como uma máquina de estado finita pela maioria dos autores (Whitaker, Hodge and Chlamtac 2005). Estes são construídos como mecanismos para controlar a relação entre os estados definidos pela especificação *Bluetooth: INQUIRY, INQUIRY SCAN, PAGE, PAGE SCAN*. Estes estados coordenados, unidos a informações de cada dispositivo, como: capacidade da bateria, tipo e capacidade de vazão de dados. Fazem com que ocorram uma diversidade de técnicas, resultando, topologias e otimização. A tabela 5.1 enumera os tipos de protocolos *Scatternets*.

5.3.2 Algoritmo Law

O algoritmo descrito por (Law, Mehta and Siu 2002) e (Law, Mehta and Siu 2001), se enquadra no item 7 da Tabela 5.1. Em sua execução, os dispositivos e os conjuntos interligados, são tratados como componentes. Sendo os tipos de componente: um dispositivo, uma *Piconet* ou uma *Scatternet*.

Tabela 5.1: Protocolos *Scatternet*

1	<i>BlueStars</i>
2	<i>BlueMesh</i>
3	<i>Scatternets</i> via inserção e remoção de nó
4	<i>BlueRings</i>
5	<i>DSFP - Distributed Scatternet formation procedure</i> dsfp
6	Formação <i>Scatternet</i> leve e simplificada
7	<i>Scatternet</i> via fusão, movimentação e migração
8	Formação <i>Scatternet</i> baseada em triangulação parcial
9	Árvores de <i>BlueRing</i>
10	Formação <i>Scatternet</i> via agrupamento
11	Formação <i>Scatternet</i> e extensão de manutenção
12	Protocolo de construção de topologia <i>Bluetooth</i>
13	Formação <i>Scatternet BlueTree</i> auto-roteada
14	Formação <i>Scatternet</i> em Árvore
15	Esquema de Formação <i>Scatternet Bluenet</i>
16	<i>Blueroot</i> e <i>Bluetrees</i> distribuídas
17	Ilhas de <i>BlueStarScatternet</i>
18	Formação <i>Scatternet</i> centralizada

Cada componente possui um líder. Nos componentes *Piconet* o líder é sempre o mestre, nos componentes *Scatternet* um dos mestres é eleito líder.

Este líder alterna *SEEK* e *SCAN* em uma probabilidade de $(\frac{1}{3} < p < \frac{2}{3})$. Este procedimento garante que em cada componente, pelo menos um dispositivo esteja executando estas funções. Quando um líder executa *SEEK* ele busca por novos escravos que estão em *SCAN*.

Quando um dispositivo em *SEEK* encontra um dispositivo em *SCAN*, o modo *CONNECTED* é chamado, e um novo link é estabelecido com o componente. A reorganização da *Piconet* ocorre via uma destas três operações: *MOVE*, *MERGE*, *MIGRATE*. Estas rotinas garantem que componentes novos e maiores tenham somente um líder coordenando a distribuição de escravos e mestres sem líder.

Em seu trabalho (Law, Mehta and Siu 2002) prova que seu protocolo possui $O(\log n)$ de complexidade de tempo e $O(n)$ de complexidade de mensagem.

Seu modelo segue as seguintes propriedades:

- qualquer dispositivo é um membro de no máximo duas *Piconets*;
- o número de *Piconets* deve ser próxima do ótimo;
- o número de *Piconets* possui um limite inferior de $(n - 1)/k$, sendo n o número de nós da rede e k o número de escravos numa *Piconet*.

Quando avaliamos os impactos na topologia resultante, a localização das bridges é fundamental. Elas são nós responsáveis pela comunicação *intra-Piconet*, e recebem maior *overhead* de comunicação e processamento que outros nós.

5.3.3 Bridges

Os elementos que permitem a comunicação multi-hop entre as *Scatternets*, são chamadas “bridges”. Elas são necessárias para que ocorra comunicação inter-Piconet. Elas alternam o padrão de saltos entre os mestres conectados. O modo *Bluetooth* usado durante esta operação é o *HOLD*. No contexto da *Scatternet* este modo é usado como solução para coexistência de um nó em mais de uma *Piconet*. Durante este modo o dispositivo participa de diferentes *Piconets* utilizando uma técnica de *Time Division Multiplexing* (TDM). Nas *Scatternets*, elas são usadas e implementadas em dois tipos: escravo-escravo e mestre-escravo.

O modo *HOLD* do *Bluetooth*, foi criado para liberar o dispositivo de uma conexão ativa com o mestre. Durante este modo um dispositivo já conectado a *Piconet*, pode dormir por um curto espaço de tempo possibilitando ao mestre comunicar ou buscar por novos dispositivos. Esta comunicação é chamada *inter-Piconet*.

Durante o transporte de mensagens *inter-Piconets*, o dispositivo *Bridge*, que atua em duas *Piconets*, vai ao estado *HOLD*. Este modo permite que ele alterne contato entre os mestre das *Piconets*. Este processo necessita de uma ressincronização do salto com a *Piconet* que deseja comunicar. Para que isto ocorra, a *bridge* troca seu padrão de salto e começa a ouvir *pollings* do mestre da *Piconet*, que começou a atuar. A partir deste instante ela pode enviar as mensagens da outra *Piconet*.

Alguns protocolos *Scatternet* fazem uso de dispositivos atuando como *Bridge* do tipo mestre-escravo, e uma *Piconet* só pode ter um mestre ativo. Estas *Bridges* mestre de uma *Piconet* e escravo de outra, possuem além de um custo de sincronização, a perda

temporária do mestre. Isto leva a um custo de sincronização excessivo para algumas aplicações.

O procedimento de transporte de mensagens entre as redes, possui alto esforço de resincronização. A necessidade de troca de padrão de salto com outras *Piconets*, ajustes de velocidade e a necessidade de inter-Piconet *Scheduling* para coordenar as atividades de um dispositivo itinerante, causam a este processo um *overhead*, devido ao tempo de guarda de *slots*. Durante a troca de *Piconet* são usados dois *time slots*, que ficam indisponíveis para comunicação. Estas características das *bridges*, influenciam diretamente na performance de uma *Scatternet*, sendo pontos de gargalo de consumo de energia e tráfego (Whitaker, Hodge and Chlamtac 2005).

5.3.4 Inter-Piconet Scheduling

Nas *Scatternets*, para coordenar a comunicação das *bridges* é necessário algoritmos de *inter-Piconet Scheduling*. Este procedimento, garante que a *bridge* esteja disponível para comunicação, assim que o mestre demande.

Uma solução mais comum para este problema, é uma reserva de *slots* para a comunicação da *bridge*, pois dois mestres compartilhando o mesmo escravo, não podem usar o mesmo para *polling*. Estes *slots* reservados, são chamados de pontos de *rendezvous*.

Em comum a todos os algoritmos de *intra-Piconet Scheduling*, existe a necessidade de determinar e controlar os pontos de *rendezvous* (*RP*). E sua correta integração com os algoritmos *inter-Piconet Scheduling*, se faz necessária.

Podemos enumerar algumas das abordagens de *inter-Piconet Scheduling* (Whitaker, Hodge and Chlamtac 2005), e seus protocolos que lidam com as janelas de *rendezvous*.

1. Algoritmo *Maximum Distance Rendezvous Point* (MDRP);
2. *Scheduling* adaptativo usando a *RP* max-min ;
3. *Flexible Scatternet Scheduling* (*FSS*), adaptativo ao tráfego e taxa de *pollings* não concluídos;
4. *Adaptive Presence Point Density* (*APPD*), algoritmo adaptativo usando créditos, de acordo com a presença de pontos de densidade ;

5. Algoritmo *Pseudo-Random Coordinated Scatternet Scheduling* (PCSS), usa sequência pseudo-randômica de *RP*;
6. *Locally Coordinated Scheduling* (LCS), tenta eliminar todos os conflitos de *Scheduling*, com um custo adicional de *overhead* na associação;
7. *QOS*, o *Scheduling* é visto como um problema de otimização, e uma análise de capacidade ocorre antes da disseminação das rotas;
8. *Load Adaptive Algorithm* (LAA), o algoritmo determina a duração das atividades das *Bridges* em cada *Piconet*;
9. Proposição de um novo modo *JUMP*, este novo módulo já teria as regras específicas para operar mais de uma *Piconets*;
10. *Scheduling* por conhecimento de interferência;
11. Análise da fila teórica;

5.4 Modelo Estático

O modelo estático de *Scatternet*, também conhecido como modelo centralizado de *Bluetooth*, não é um protocolo. Ele descreve uma formação *Scatternet* usando programação matemática. Neste modelo, restrições são propostas em uma formulação de min-max, levando a um problema de otimização resolvido de uma forma centralizada.

Este modelo, procura como solução a melhor topologia possível para um grafo de *Scatternet*, respeitando as restrições de uma *Piconet Bluetooth*.

O objetivo deste modelo é minimizar o tráfego nos dispositivos sujeitos a grande *overhead* e consumo de energia, como mestres e *bridges*. Durante a busca pela solução, restrições são respeitadas até a convergência de uma *Scatternet* totalmente conectada.

5.4.1 Modelo centralizado de Marsan et al.

O modelo centralizado de (Ajmone Marsan, Chiasserini, Nucci, Carello and De Giovanni 2002), discute alguns requisitos de uma *Scatternet* centralizada:

- Conectividade da Rede: Deve existir ao menos um caminho entre dois dispositivos na rede;
- Complexidade do sistema: para reduzir a complexidade da rede, o número de *Piconets* é limitado a um valor fixo;
- Demanda de tráfego: a rede deve suportar o necessário para conexão origem-destino;
- Papel do nó: Algumas restrições devem ser aplicadas para alguns nós, de acordo com o papel que ele exerce: mestre ou escravo.

As restrições e requisitos usados neste modelo são:

- Estrutura de Rede:
 - Nós ativos participando da *Piconet* não podem ser maior que 8;
 - Dois dispositivos para comunicar devem estar no alcance do outro;
 - Um nó só pode ser mestre em uma *Piconet*;
- Capacidade do Sistema: O máximo *bit rate* de uma *Piconet* é igual a 1 *Mbit/s*

Estes requisitos e restrições são usados por (Ajmone Marsan, Chiasserini, Nucci, Carello and De Giovanni 2002) para gerar um critério min-max que é resolvido usando CPLEX (IBM Last Visited in 11/02/2013).

5.5 Influência das bridges na eficiência

A localização das *bridges*, é crítica para avaliação do impacto de uma topologia *Scatternet*. Elas são responsáveis pela comunicação *inter-Piconet*, e são submetidas a maior sobrecarga de comunicação e processamento que os outros nós.

Avaliamos as influências do número de *bridges* em uma *Scatternet*. Para atuar como *bridge*, o escravo *Bluetooth* entra em modo *HOLD*, necessário para o sincronismo de comunicação entre as *Piconets*. Durante o funcionamento neste modo, observamos um alto custo energético, e este dispositivo recebe maior esforço, durante a resincronização de frequências com os mestres, necessitando uma correta coordenação de uso das janelas

de *rendezvous*. Para correto sincronismo deste processo, uma estratégia de inter-Piconet *Scheduling* é necessária.

Durante o *intra-Piconet Scheduling*, em algoritmos de (*RP*) mais simples, a *Bridge* aguarda por *pollings* dos mestres, para destinação de mensagens.

Algumas razões que levam a obter uma topologia *Scatternet*, eficiente:

1. Poucas *bridges*, significa menos atraso em migrar para outra *Piconet* durante a transmissão e recebimento de pacotes;
2. Poucos mestres na rede, resultam em menos *Piconets* e *bridges*, conseqüentemente menos esforço de sincronização de canais e influências de *inter-Piconet Scheduling*.

5.6 Conclusão

O volume de *Bridges*, e sua posição na *Scatternet*, influenciam diretamente no consumo de energia e no tempo de tráfego de pacotes. Um correto entendimento dos efeitos colaterais do uso de *FHSS*, são essenciais para geração de modelos *Scatternet* que resultem em uma topologia otimizada. No próximo capítulo, mostraremos avanços no trabalho de (Pettarin, Pietracaprina and Pucci 2009). Apresentaremos seu modelo de grafo *Bluetooth*, e acrescentaremos ao modelo, o sincronismo de frequências do *FHSS* e o processo de *Discovery*. Estes novos elementos caracterizam o *Bluetooth* como grafo dinâmico.

Capítulo 6

Grafo Dinâmico

6.1 Introdução

Neste capítulo, iremos discutir e propor novos avanços aos modelos clássicos de *Bluetooth*, como o proposto por (Pettarin, Pietracaprina and Pucci 2009). Em seu estudo, ele discute expansão e diâmetro da topologia *Bluetooth*, que é induzida pela fase de *Discovery*, e representada por um *Random Geometric Graph (RGG)*. Portanto, este estudo não aborda as características de uma *Piconet*, nem as características do *FHSS*. Nós avançamos este trabalho acrescentando ao modelo clássico características, definindo o grafo que representa o *Bluetooth* como dinâmico.

6.2 Definindo Bluetooth como grafo dinâmico

A rede *Bluetooth* pode ser descrita como um grafo, seguindo a definição de (Gupta and Kumar 2000) and (Pettarin, Pietracaprina and Pucci 2009).

Considere o grafo direcionado $\mathcal{G} = \{\mathcal{V}, \mathcal{E}'\}$ composto pelo conjunto de $n \geq 1$ nós $\mathcal{V} = \{v_1, \dots, v_n\}$ e de arestas $\mathcal{E}' \subset \{\mathcal{V} \times \mathcal{V}\}$, tal que $(e_i, e_j) \in \mathcal{E}' \iff (e_j, e_i) \in \mathcal{E}'$. Definiremos um grafo dinâmico *Bluetooth*, que permite descrever a formação de *Piconets* e *Scatternets*, com base neste grafo genérico, e a inclusão de restrições espaciais e *labels* aos nós.

Uma suposição comum para colocação de nós é o modelo de distribuição totalmente independente ou distribuição binomial (Ramos, Guidoni, Nakamura, Boukerche,

Frery and Loureiro 2011). De acordo com este modelo, dado n nós, suas coordenadas $(x_i, y_i)_{1 \leq i \leq n}$ em um quadrado $[0, 1]^2$ são resultados de $2n$ independentes identicamente distribuídas uniformemente $[0, 1]$ variáveis aleatórias.

Os nós representam dispositivos, e uma vez que eles são implantados, os canais que permitem a comunicação são construídos de acordo com alcance $r(i)$ de cada dispositivo.

A regra geométrica diz que a aresta $(e_i, e_j) \in \mathcal{E}'$ pode existir somente se $d(v_i, v_j) \leq \min\{r(i), r(j)\}$, i.e., nós v_i e v_j podem comunicar somente se ambos podem se falar. A função distância $d: [0, 1]^2 \times [0, 1]^2 \rightarrow R_+$ é arbitrária, e nela pode ser incorporada qualquer informação prévia sobre o meio ambiente, como por exemplo: os obstáculos. A escolha pelo quadrado unitário não diminui qualquer necessidade de generalidade no modelo. Muitas aplicações assumem a configuração de comunicação recíproca $r(i) = r$ para cada $1 \leq i \leq n$.

A definição de protocolos para operação do *Bluetooth* requer um outro ingrediente: identificação de mestres e escravos. Cada nó recebe um *label*, “M” ou “S” para representar seu estado corrente. Com isso, um novo grafo agora pode ser definido, desde que o grafo \mathcal{G} seja definido como anteriormente. O Grafo *Bluetooth* BT é um subconjunto de \mathcal{G} tal que $BT = \{\mathcal{V}, \mathcal{E}\}$ uma vez que, possivelmente, nem todas as conexões permitidas foram definidas. A especificação da comunicação \mathcal{E} tem que satisfazer as seguintes requisições:

1. Existe ao menos um nó M.
2. Todos nós S conectam com um e somente um M.
3. Nós M não possuem conexão entre eles.
4. Nós S não possuem conexão entre eles.

Uma *Piconet* só é formada, se existe somente um nó M e todos nós S conectados a ele. Se existe mais que um nó M, existe a presença de uma *Scatternet*.

(Pettarin, Pietracaprina and Pucci 2009) descreve a situação de dispositivos em movimento, em que o grafo BT é dinâmico, e é uma função do tempo $BT(t) = \{\mathcal{V}, \mathcal{E}(t)\}$. O Autor descreve a sequência de grafos por meio de uma chegada de Markov, cujas transições expressam as mudanças nas conexões devido ao movimento de nós.

Denote $F = \{f_i : 0 \leq i \leq 79\}$ o conjunto de frequências usadas no *FHSS*, então *FHS* é a função $FHS(CLK, MS)$, onde *CLK* é o clock dos elementos envolvidos e *MS*

é o endereço do mestre da *Piconet*. Os detalhes do *FHS* são dados pela especificação do *Bluetooth*. Cada link mestre-escravo tem uma única sequência pseudo-aleatória de frequências $F' = (f'_1, f'_2, \dots)$, de modo que $f'_i \in F$.

Considere u e v serem nós. Nós definimos o processo de descoberta como uma operação $\text{Disc}(u, v, f_i)$ que consiste na inserção da aresta (u, v) em \mathcal{E} .

O processo $\text{Disc}()$ tem sua execução distribuída. Mestre e escravo começam uma sequência de saltos pseudo aleatórios, até que ocorra uma coincidência de f_i . Após o casamento de frequências ocorrer, o escravo espera por um tempo aleatório e responde um $FHS()$ ao mestre, este é chamado intervalo de *Backoff*. Isto é necessário porque o $FHS()$ deve ser trocado entre os nós. Por esta razão, após $\text{Disc}()$ ter sido aplicado, o escravo retorna se endereço de rede e relógio para que o mestre gere a correta sequência pseudo aleatória F' para conexão. Os labels “M” e “S” são dados aos nós selecionados como mestre e escravo, respectivamente.

Estas mudanças na conectividade de rede *Bluetooth*, como descritas acima, fazem desta um grafo dinâmico (Frigioni and Italiano 1997).

As Figuras 6.16.26.36.46.56.66.7 ilustram a formação da aresta (u, v) .

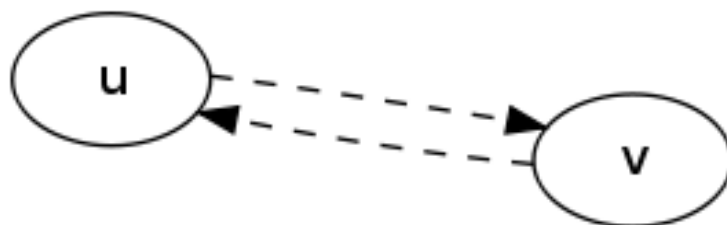


Figura 6.1: Dispositivos u v ao alcance um do outro

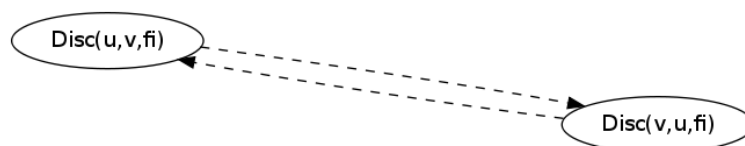


Figura 6.2: u v iniciam o procedimento distribuído $\text{Disc}()$

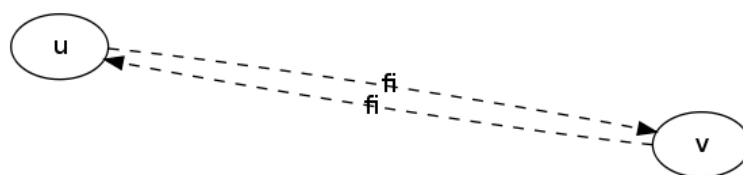


Figura 6.3: Ocorre a coincidência de canal f_i

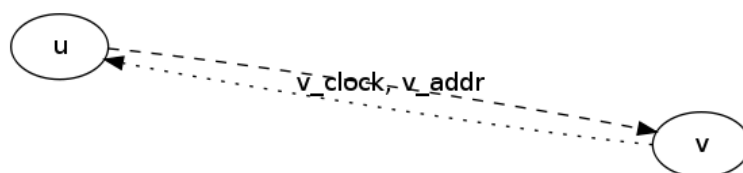


Figura 6.4: v retorna seu endereço de rede e relógio

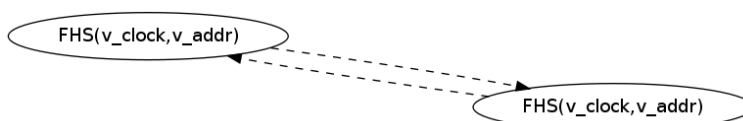


Figura 6.5: Dispositivos usam procedimento $FHS()$ para gerar F'

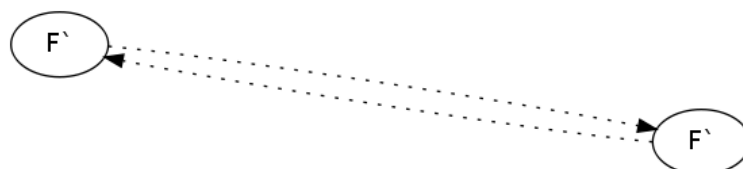


Figura 6.6: Com a mesma sequência de frequência F' agora pode ocorrer comunicação criação e sincronização da *Piconet*

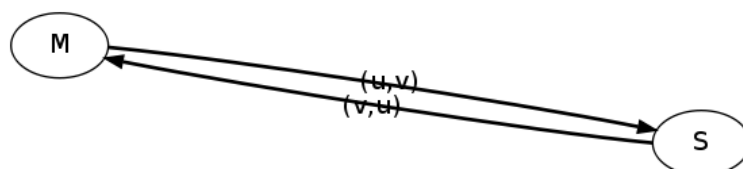


Figura 6.7: Nós recebem etiquetas de M - mestre e S - escravo

6.3 Conclusão

Uma representação em grafo dinâmico do *Bluetooth*, ajuda a explorar as particularidades desta rede, possibilitando identificar os efeitos colaterais do uso de *FHSS*.

Esta correta representação é essencial para o estudo dos atrasos no processo de formação de uma *Piconet*. A aleatoriedade deste processo, leva a possibilidade de alguns dispositivos não conseguirem conectar e mesmo não serem encontrados.

No próximo capítulo, demonstraremos estes efeitos, simulando alguns cenários comuns nas formações de rede *Bluetooth*.

Capítulo 7

Os Efeitos do FHSS

7.1 Introdução

Neste capítulo, simulamos a formação de *Piconets* para analisar alguns dos efeitos colaterais do uso de *FHSS*, como: descoberta de novos nós para formar uma nova *Piconet*, a entrada de novos nós escravos a uma *Piconet* existente e tempos de convergência total da formação. Para isso usamos o UCBT (Q. Wang Last Visited in 11/02/2013), uma extensão que simula *Bluetooth* no ns-2 (NS2 Last Visited in 11/02/2013), desenvolvido pela Universidade de Cincinnati.

7.2 Experimentos

Afim de avaliar o atraso durante a fase de descoberta na formação de novas *Piconets*, geramos instâncias com um mestre i , e n dispositivos no alcance $r(i)$ que são candidatos a escravos. A experiência foi repetida 30 vezes, com sementes diferentes para cada valor de n .

De acordo com (Pettarin, Pietracaprina and Pucci 2009), nós podemos observar que, a medida que o valor de n aumenta, o mesmo acontece com a probabilidade de conexão. Este comportamento é observado na Figura 7.1, onde a medida que aumentam o número de dispositivos ao alcance do mestre, maior a probabilidade de uma coincidência de frequência e mais rápido ocorre uma resposta do candidato a escravo.

Isto ocorre porque durante o *Discovery*, todos os dispositivos INQUIRY SCAN aguardam

dando ser encontrado realiza saltos pseudo-aleatórios em *time slots* mais lentos que o mestre, até ocorrer uma coincidência de frequências.

Este comportamento mostra que, apesar do aumento da densidade de dispositivos no intervalo de mestre, o *FHSS* recebe menos interferências mesmo que ocorra saturação do espectro de frequências.

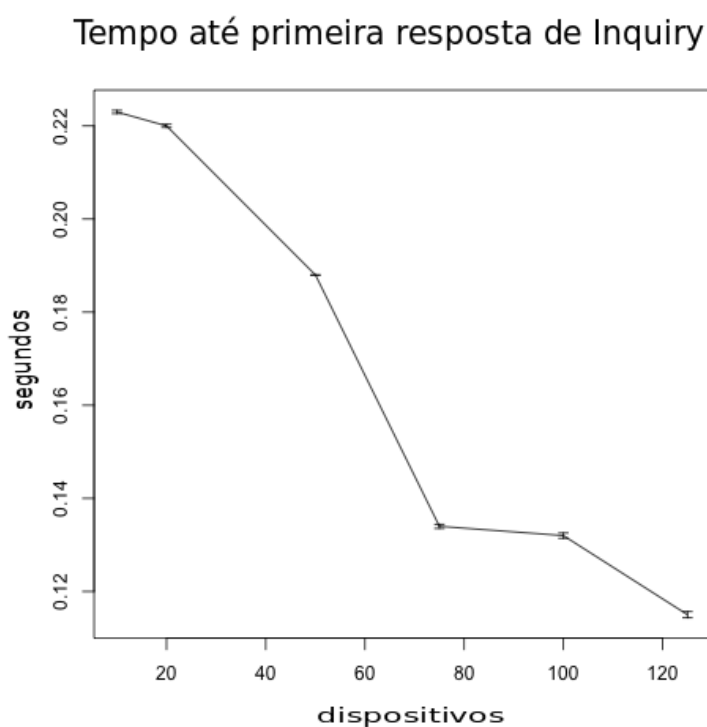


Figura 7.1: Tempo até a primeira resposta ao *Inquiry* do candidato a mestre

7.3 Análise do tempo de convergência total das redes

A Figura 7.2 mostra o tempo até a formação de uma *Piconet* teórica máxima, representada por um mestre e sete escravos, observamos a variação de tempo necessário para formação ariando com o número dispositivos ao alcance do mestre. Para gerar este gráfico, geramos 30 instâncias de simulação, com os valores de 10, 20, 50, 70, 100, 130 dispositivos ao alcance do mestre.

A barra de desvio padrão representa a variação nos tempos de formação em instâncias de mesmo número de dispositivos. Esta variação tem seu comportamento explicado,

pela presença do tempo até a coincidência de canal na fase de descoberta, somados ao intervalo de *Backoff* e esforço de coordenação do *Scheduling intra-Piconet*. Durante esta formação, o mestre executa novamente a descoberta, e os escravos já conectados entram em modo *HOLD*, a espera de novos *pollings* do mestre que os autorizem a comunicar. O tempo total desta operação cresce com o aumento de dispositivos candidato a escravos ao alcance do mestre, e representa a soma de tempo despendido comempo *gintra-Piconet Scheduling* e descoberta, durante o ingresso de novos escravos a *Piconet*.

A barra de erro na Figura 7.2 mostra o desvio padrão e *delay* de conexão, representado pela aleatoriedade de variáveis associadas ao processo de *Discovery* de escravos, intervalo de *Backoff* e *intra-Piconet Scheduling*.

Tempo até formação de uma Piconet Máxima

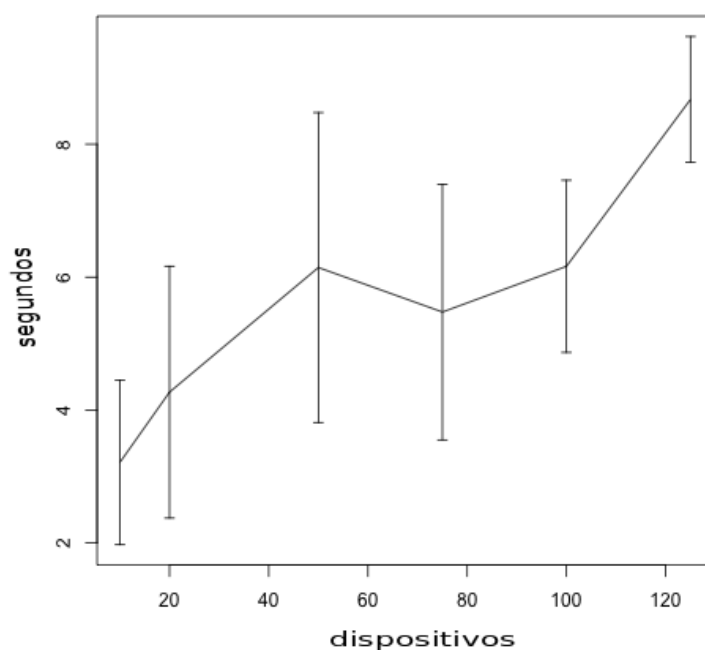


Figura 7.2: Tempo até a formação da primeira *Piconet* máxima com sete escravos e um mestre

A Figura 7.3 mostra a média do número de pacotes *INQUIRY* necessários para formação de uma *Scatternet* com n dispositivos. O elevado número de pacotes de *Inquiry*, mostra a necessidade de uso constante da fase de descoberta, na formação de redes maiores que uma *Piconet*.

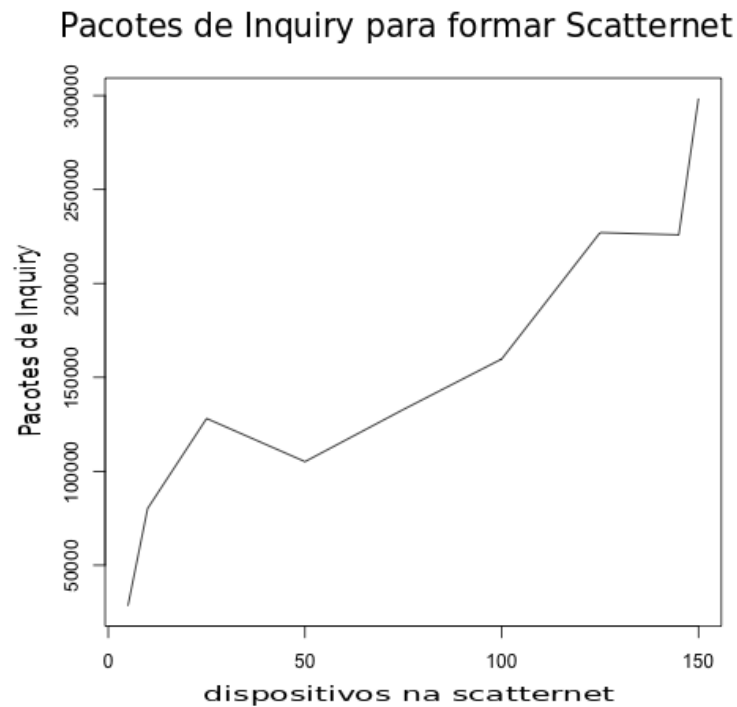


Figura 7.3: Número de pacotes INQUIRY necessários até a formação de uma *Piconet* com máxima capacidade

7.4 Conclusão

Ao simular a inicialização de uma *Piconet*, notamos que a aleatoriedade do *Discovery* e *Backoff* apesar de necessárias para minimizar a colisão, introduzem até segundos de delay no processo de conexão do *Bluetooth*.

Estes atrasos prejudicam a aplicação de algumas implementações, em que o atraso de segundos do *Discovery* a torna proibitiva, devido a necessidade de uma responsividade adequada. No próximo capítulos avaliaremos os efeitos do *FHSS* em uma topologia *Scatternet*

Capítulo 8

Análise da Topologia

8.1 Introdução

Para avaliar a eficiência de uma topologia *Scatternet* e a influência das *Bridges*. Nós simulamos a formação fornecida pelo algoritmo discutido por (Law, Mehta and Siu 2002). Geramos 30 instâncias com 20 e 30 dispositivos até a convergência total da *Scatternet*.

8.2 Análise

Simulamos usando algoritmo *LawProt* para formar *Scatternet* em *ns-2*, com 30 instâncias para cada rede de 20 e 30 dispositivos. A Figura 8.1 e a Figura 8.2, representam as topologias *Scatternet* mais comumente encontradas nas simulações.

A topologia destes grafos seguem as regras de eficiência proposta na Seção 5.5 e usadas pelo algoritmo (Law, Mehta and Siu 2002).

Analisando o grafo gerado por um instância de 20 nós usando o modelo estático em (Ajmone Marsan, Chiasserini, Nucci, Carello and De Giovanni 2002) mostrado na Figura 8.3, podemos notar que algumas das regras que influenciam na performance de uma *Scatternet* são desrespeitadas:

- A conexão entre o nó mestre 13 com o nó 0, é um *link* mestre/mestre;
- O nó 9 é *Bridge* de três *Piconets*, que não é aceitável, devido ao enorme custo de

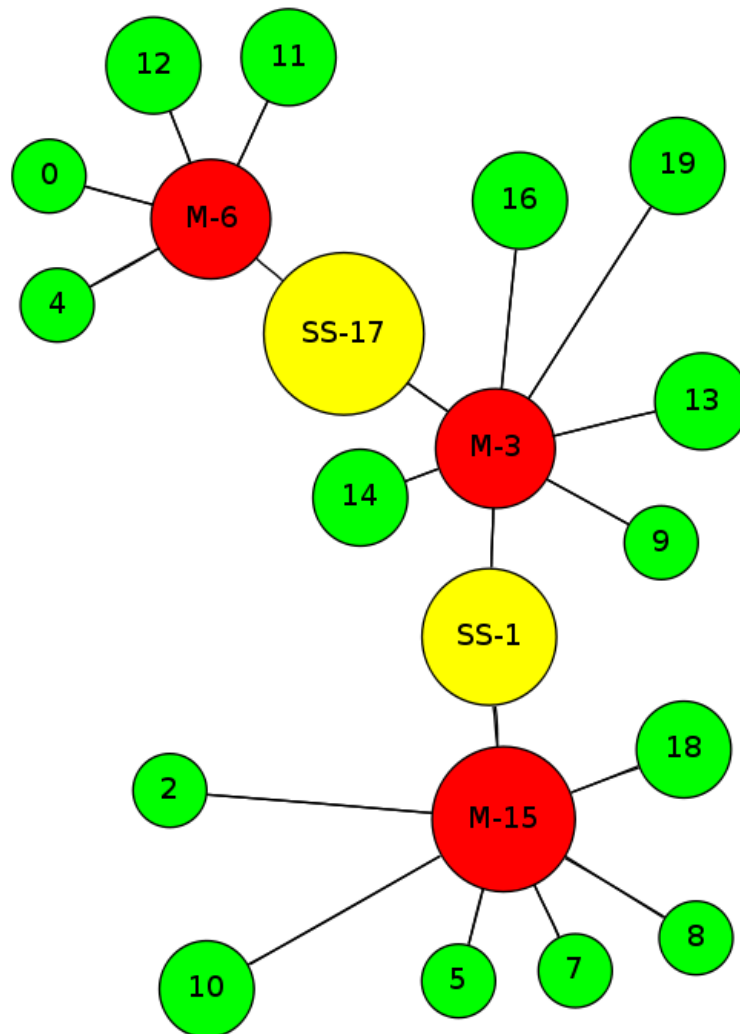


Figura 8.1: Topologia *Scatternet* encontrada na simulação do algoritmo Law et al. de uma rede com 20 dispositivos

coordenação e *Scheduling*.

- Podemos observar diversos *loops* de rede entre os mestres 7 e 17, ligados pelos nós 9 e 15;
- Quatro *Piconets* é um número excessivo para uma *Scatternet* de 20 nós.

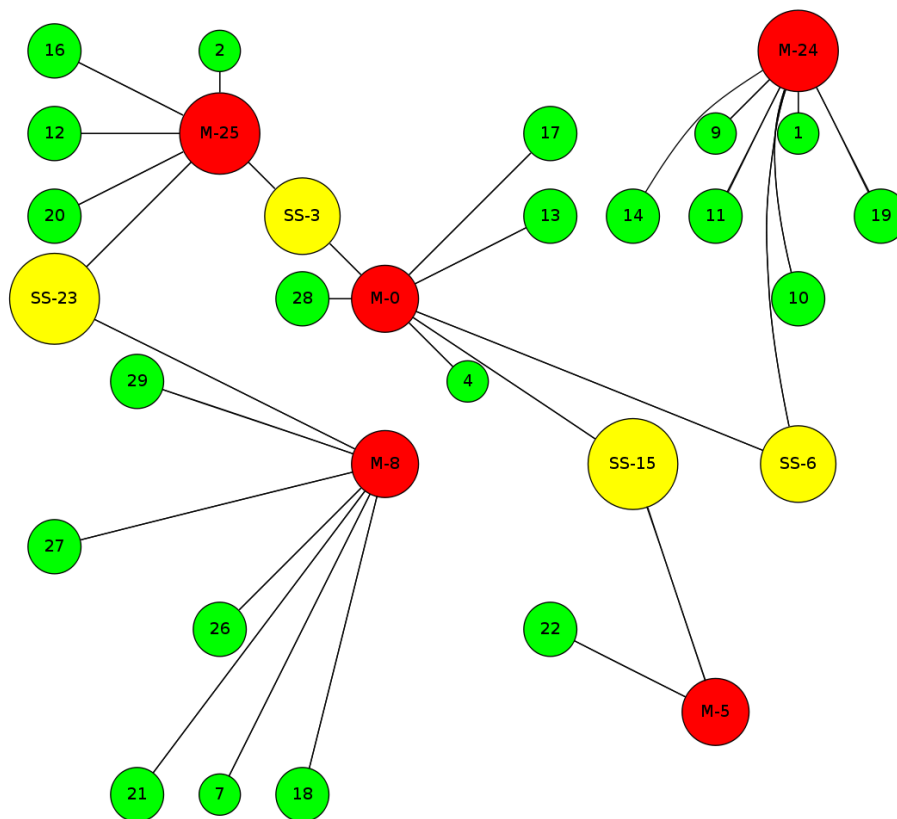


Figura 8.2: Topologia *Scatternet* encontrada na simulação do algoritmo Law et al. de uma rede com 30 dispositivos

8.3 Expandindo o modelo centralizado de Marsan et al.

O modelo centralizado de *Scatternet* de (Ajmone Marsan, Chiasserini, Nucci, Carello and De Giovanni 2002), precisam ser modificados para obter resultados próximos ao algoritmo (Law, Mehta and Siu 2002), resultando em uma formação eficiente em tráfego e consumo de energia.

O modelo (Ajmone Marsan, Chiasserini, Nucci, Carello and De Giovanni 2002) é descrito com os seguinte parâmetros:

- N - Numero de nós;
- C - Conexões de rede;
- M_{MAX} - Máximo de *Piconets*;
- X_{MAX} - Número Máximo de nós ativos na *Piconet*;

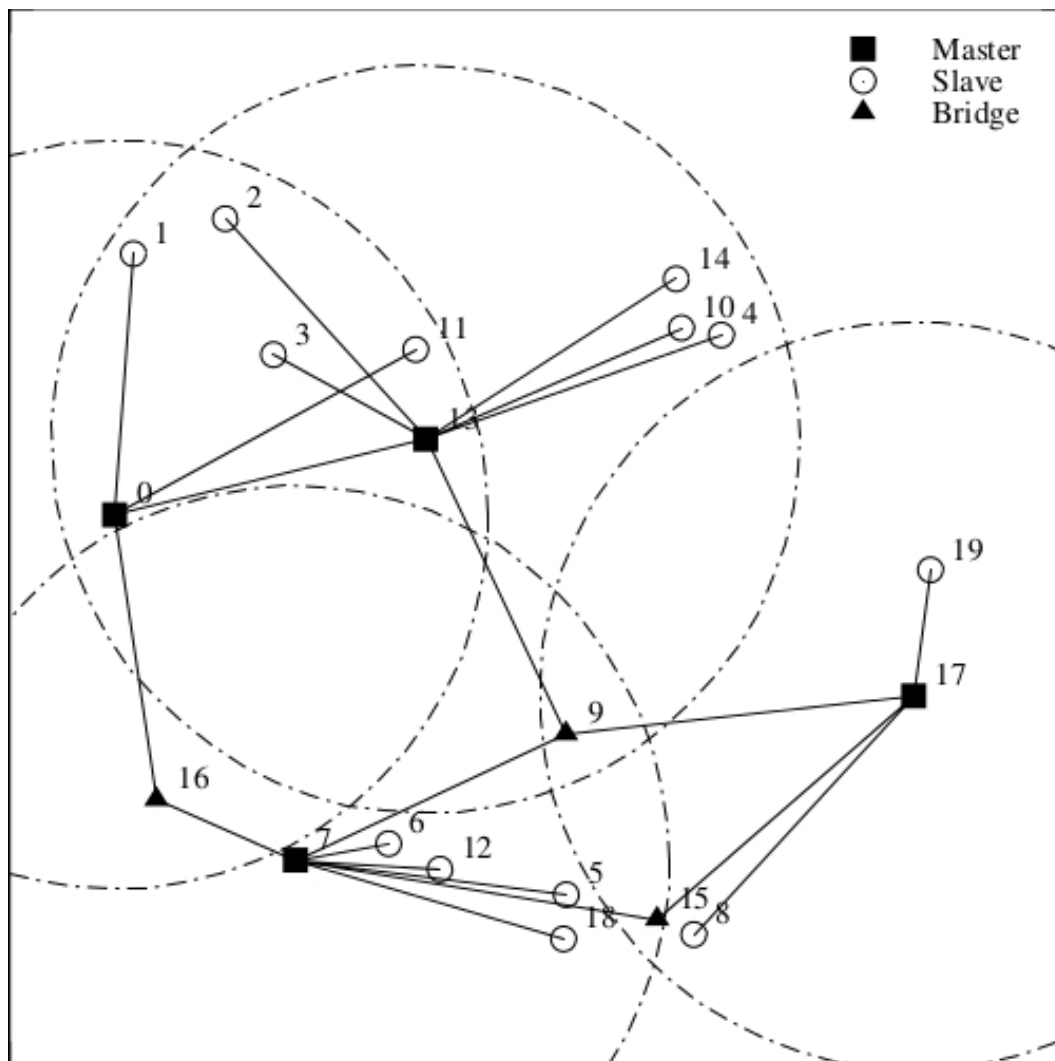


Figura 8.3: Topologia *Scatternet* de 20 dispositivos encontrado pelo modelo estático de Marsan et al. (Ajmone Marsan, Chiasserini, Nucci, Carello and De Giovanni 2002)

- Z_{MAX} - Raio Máximo da *Piconet*.
- M - Nós obrigados a agir como mestre;
- V - Nós obrigados a agir como escravos.

Para cada nó i , $i \in N$, três variáveis binárias são definidas, μ_i , β_i , e σ_i e:

- μ_i recebe o valor de 1 caso o nó seja um mestre e 0 caso contrário;
- β_i recebe o valor de 1 caso o nó seja um *Bridge* e 0 caso contrário;

- σ_i recebe o valor de 1 caso o nó seja um *escravo* e 0 caso contrário.

Para cada par de nós (i, j) , $i, j \in N$, os elementos do conjunto $X = \{x_{ij}\}$, x_{ij} são configurados para 1 se j está associado ao mestre i e 0 em outro caso.

O modelo obedece as seguintes restrições, apresentadas na Tabela 8.1 :

$$\mu_i + \beta_i + \sigma_i = 1, \quad \forall i \in N \quad (8.1)$$

$$\sum_{i \in N} x_{ij} \leq \sigma_j + |N| \cdot \beta_j + |N| \cdot \mu_j, \quad \forall j \in N \quad (8.2)$$

$$\sum_{i \in N} x_{ij} \geq 2 - \sigma_j - \mu_j, \quad \forall j \in N \quad (8.3)$$

$$x_{ii} = \mu_i, \quad \forall i \in N \quad (8.4)$$

$$x_{ij} \cdot z_{ij} \leq Z_{MAX} \cdot \mu_i, \quad \forall i, j \in N \quad (8.5)$$

$$\sum_{j \in N} x_{ij} \leq X_{MAX} \cdot \mu_i, \quad \forall i \in N \quad (8.6)$$

$$2 + x_{ji} \geq \mu_i + \mu_j + x_{ij}, \quad \forall i, j \in N, \quad i \neq j \quad (8.7)$$

$$x_{ik} + x_{jk} \leq 4 - \mu_i - \mu_j - x_{ij}, \quad \forall i, j, k \in N, \quad i \neq j, \quad j \neq k \quad (8.8)$$

$$\sum_{i \in N} \mu_i \leq M_{MAX} \quad (8.9)$$

$$\sum_{i \in M} \mu_i = |M| \quad (8.10)$$

$$\sum_{i \in V} \sigma_i = |V| \quad (8.11)$$

Afim de obter na solução de (Ajmone Marsan, Chiasserini, Nucci, Carello and De Giovanni 2002), uma topologia similar a obtida pelo algoritmo (Law, Mehta and Siu 2002), adicionamos penalidades aos nós *Bridge*, e duas novas restrições ao modelo original (Soares Ferreira, Rabelo Oliveira, Gambini and Frery 2013a).

- Um mestre deve pertencer a uma *Piconet* somente: $\mu_i + \mu_j + x_{i,j} \leq 2$, para cada $i, j \in N$ e $i \neq j$;
- Uma *Bridge* deve ligar somente duas *Piconets*: $\beta_i + x_{ij} + x_{ji} + x_{ik} + x_{ki} + x_{kl} + x_{lk} \leq 3$, para cada $i, j, k, l \in N$ e $i \neq j \vee i \neq k \vee i \neq l \vee j < k \vee k < l$.

Ao adicionar penalidades nas *Bridges* e estas duas novas restrições, podemos afirmar que o grafo resultante da solução, possui uma topologia menos propensa aos efeitos

Tabela 8.1: Restrições do modelo estático de (Ajmone Marsan, Chiasserini, Nucci, Carello and De Giovanni 2002)

<i>Restrições</i>	<i>Descrição</i>
8.1	um nó ou é um mestre, ou um escravo ou uma <i>Bridge</i> ;
8.2	um escravo é atribuído a um mestre, no máximo;
8.3	um escravo ou um mestre são atribuídos a um <i>Piconet</i> pelo menos; enquanto que uma <i>bridge</i> é atribuída a duas <i>Piconets</i> , pelo menos;
8.4	um mestre é atribuído a ele mesmo;
8.5	a distância máxima de conexão é Z_{MAX} ;
8.6	o tamanho limite de nós em uma <i>Piconet</i> é X_{MAX} ;
8.7	força a atribuição de i para j se j é atribuído para i , dado que i e j são mestres;
8.8	previne ciclo entre conjunto de três nós;
8.9	o número máximo de mestres é M_{MAX} ;
8.10	nós em M serão mestres;
8.11	nós em V serão escravos.

dos atrasos do nó *Bridge*, conectados em modo *HOLD*, e responsáveis por coordenar a comunicação entre as *Piconets* (Soares Ferreira, Rabelo Oliveira, Gambini and Frery 2013b).

Considerando o grafo dinâmico, estas penalidades são associadas ao custo dos efeitos do *FHSS* representadas na operação *FHS()* apresentada antes.

Estes requisitos e restrições levam a um critério min-max que resolvemos usando o *solver* CPLEX (IBM Last Visited in 11/02/2013).

Afim de avaliar a solução gerada pela adequação de nossa proposta, foi utilizado como exemplo a instância de (Ajmone Marsan, Chiasserini, Nucci, Carello and De Giovanni 2002), representada pelos parâmetros da Tabela 8.2 e o o grafo da Figura 8.3.

Nós notamos que, a topologia *Scatternet* encontrada por nosso modelo representado pela Figura 8.4, todos itens necessários para uma formação eficiente discutidas em 5.5 e 5.3 são respeitadas.

Na topologia encontrada na solução de nosso modelo, são encontrados 3 mestres em

Tabela 8.2: Parâmetros de Entrada

N	C	M_{MAX}	X_{MAX}	Z_{MAX}	M	$ V $
20	15	4	8	$\frac{10\sqrt{2}}{3}$	{7, 17}	0

3 Piconets e 2 *Bridges*, topologia aproximada de nossa simulação do algoritmo (Law, Mehta and Siu 2002) utilizando 20 dispositivos como mostrada na Figura 8.1.

Estes resultados são significativos, pois o algoritmo (Law, Mehta and Siu 2002) possui um custo $O(\log n)$ de complexidade de tempo e $O(n)$ de complexidade de mensagem. Dado este resultado podemos afirmar que o grafo resultante de nosso modelo representa uma *Scatternet* com distribuição ideal de fluxo de dados e consumo de energia.

Afim de validar o modelo, usamos a topologia *Scatternet* encontrada por simulação de uma instância de 30 dispositivos em algoritmo (Law, Mehta and Siu 2002), representada pelo grafo da Figura 8.2. Usamos as coordenadas deste grafo, e as variáveis da Tabela 8.3 como entrada. Buscamos a solução do modelo estático original de (Ajmone Marsan, Chiasserini, Nucci, Carello and De Giovanni 2002) e do nosso modelo, e geramos o grafo da topologias *Scatternets* obtida, representadas pelos grafos: Figura 8.5 e Figura 8.6 respectivamente.

Tabela 8.3: Parâmetro de entrada para *Scatternet* de 30 dispositivos

N	C	M_{MAX}	X_{MAX}	Z_{MAX}	M	$ V $
30	4	8	8	$\frac{10\sqrt{2}}{3}$	{5, 25}	0

Podemos observar que, a topologia *Scatternet* de 30 dispositivos resultada pelo modelo estático de Marsan mostrada na Figura 8.5, possui os seguintes itens que influenciam em sua performance:

- As conexões entre o nó 4 e 25, 27 e 5, 8 e 4, são links mestre - mestre;
- 8 *Piconets* e 6 *bridges* são um número excessivo para uma *Scatternet* de 30 dispositivos.;

A *Scatternet* encontrada pelo modelo alterado de (Ajmone Marsan, Chiasserini, Nucci, Carello and De Giovanni 2002) possui topologia similar a formada pelo algo-

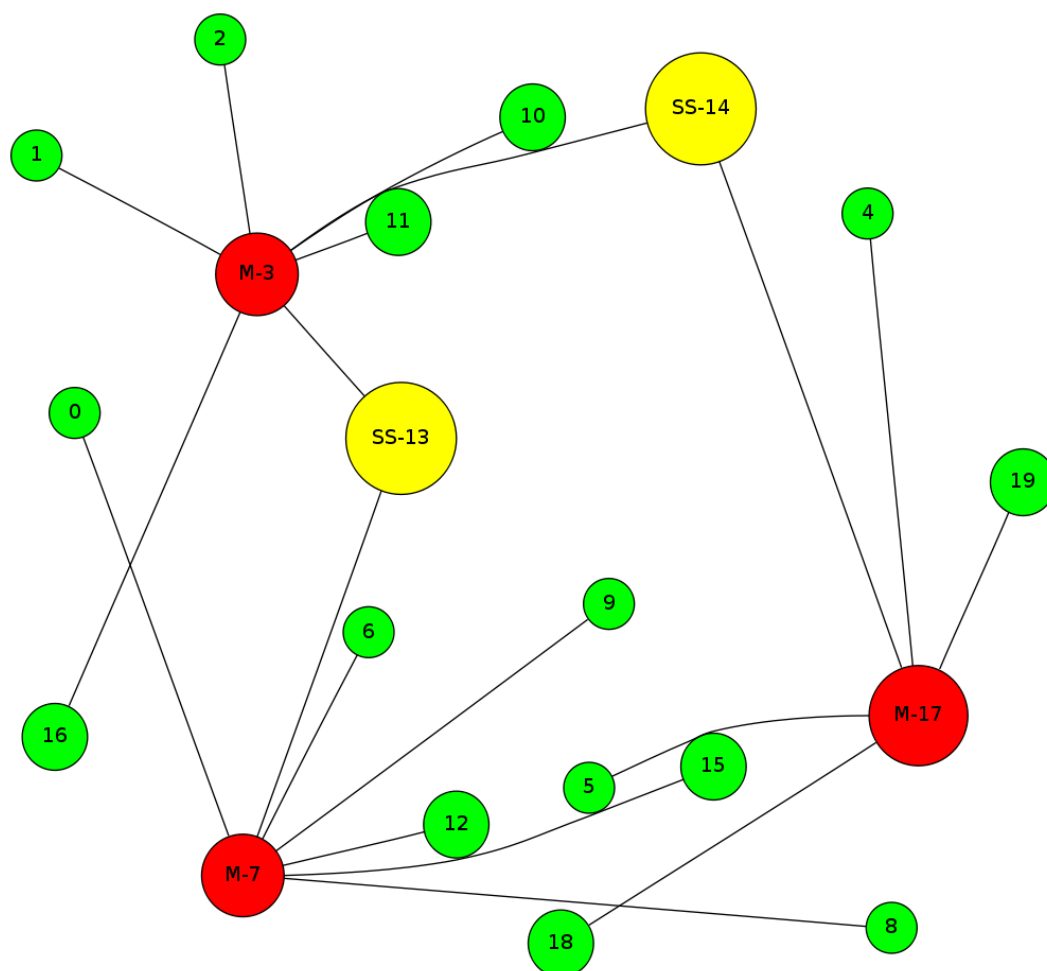


Figura 8.4: Topologia Scatternet de 20 dispositivos, encontrada como solução de nosso modelo estático

ritmo (Law, Mehta and Siu 2002), 5 Mestres em 5 *Piconets* e 4 *bridges*. Esta topologia segue o limite inferior de *Piconets* descrito em 5.3.2, recebe menor influência do esforço de coordenação do *inter-Piconet Scheduling*, necessário durante o modo *HOLD*. Esta topologia também segue o limite inferior de *Piconet* proposto por *LawProt*, e possui um número mínimo de *bridges* possível.

8.4 Conclusão

Os dispositivos atuantes como *Bridges*, são efetivamente pontos de maior perda de eficiência de uma topologia *Scatternet*. Elas são os responsáveis pela coordenação de tráfego de pacotes *inter-Piconet*. O esforço deste processo, que ocorre durante o modo

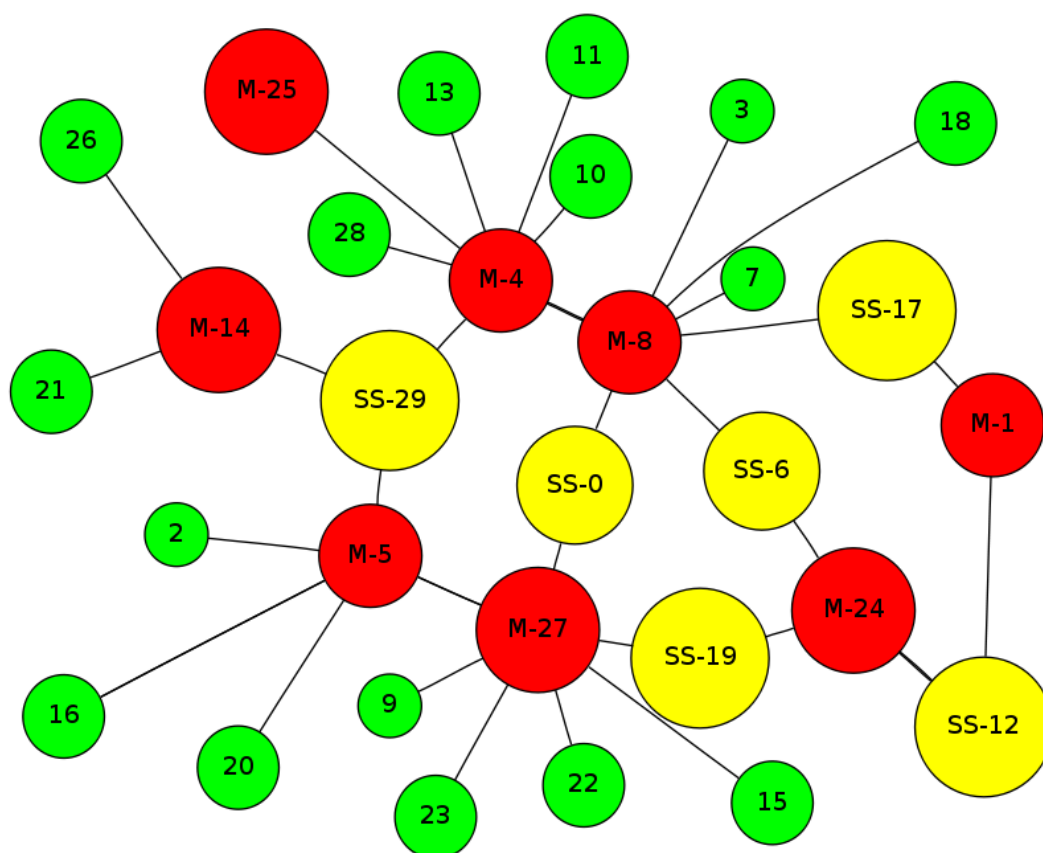


Figura 8.5: Topologia Scatternet de 30 dispositivos, encontrada como solução do modelo de Marsan et al.

HOLD, faz delas gargalos de rede e pontos de maior consumo de energia por definição.

Ao simularmos *Scatternet* com algoritmo *LawProt* e extrairmos seu grafo visual, obtemos uma topologia em árvore, formada por um protocolo que possui complexidade de tempo $O(\log n)$ e $O(n)$ de complexidade de mensagem. E ao comparamos a topologia das soluções obtidas originalmente pelo modelo de *Marsan*, observamos na solução deste modelo diversos problemas de performance associados a redes *Bluetooth*. Estes problemas são diretamente relacionados aos gargalos gerados pelo posicionamento de suas *bridges* e número excessivo de *Piconets*.

Quando adicionamos ao modelo original de *Marsan*, penalidade para as *bridges* e duas novas restrições, obtemos soluções com topologia *Scatternet* em árvore, similares em número de *bridges*, *Piconets* e Mestres obtidos por simulação do protocolo discutido em (Law, Mehta and Siu 2002).

Considerando a similaridade da topologia obtida, podemos considerar que nosso re-

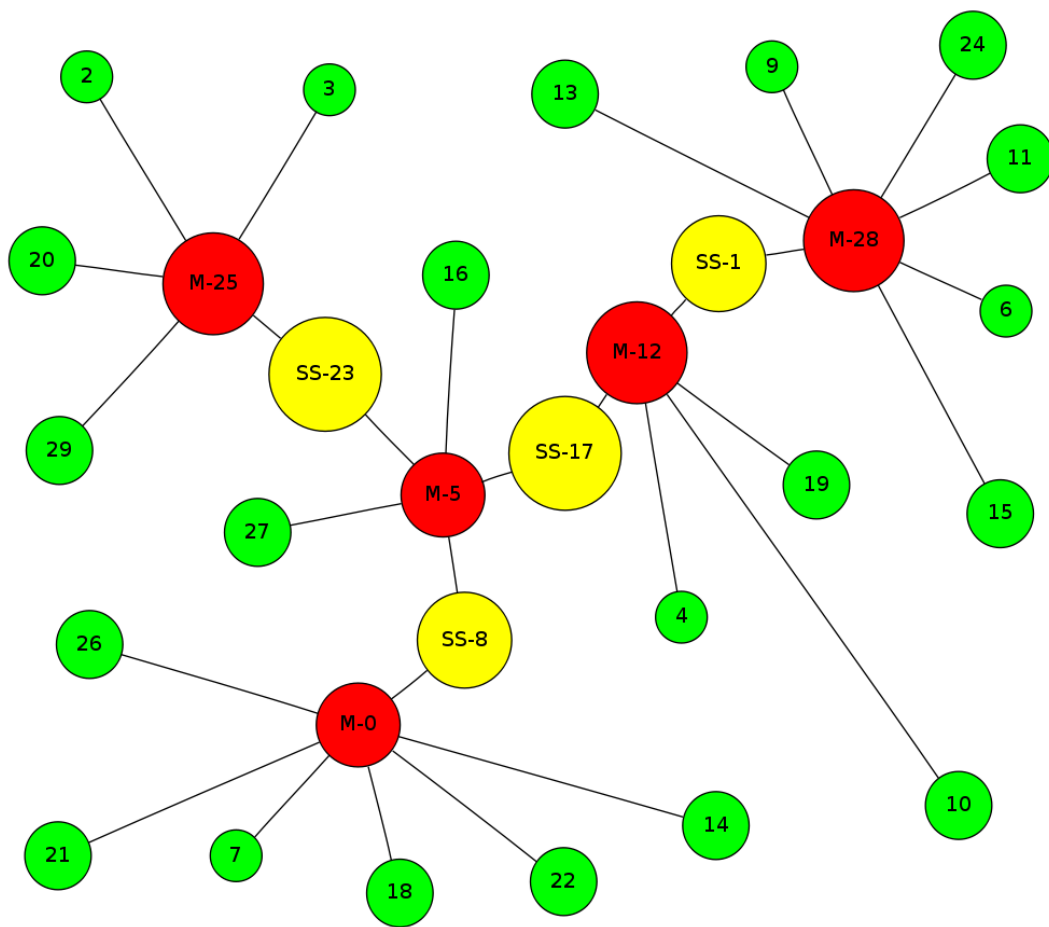


Figura 8.6: Topologia Scatternet de 30 dispositivos, encontrada como solução de nosso modelo.

sultado possui melhor desempenho que o modelo original.

Capítulo 9

Conclusão

O atraso do processo de conexão *Bluetooth* e perda de eficiência dos algoritmos *Scatternet* estão diretamente relacionadas com o *FHSS*. Um mapeamento correto de suas peculiaridades, e a inserção de suas características nos modelos que o representam, é fundamental para um bom design de soluções e aplicações *WPAN Bluetooth*.

O atraso no processo de *Discovery* é fator determinante para restringir a implementação de aplicações mais simples, que fazem uso de uma *Piconet*, e necessitam de uma resposta de conexão imediata. Os procedimentos de *Disc()* e *FHS()* definidos no gráfico dinâmico discutidos no Capítulo 6, representam as operações que geram atraso na conexão inicial. E descrevem importantes procedimentos que influenciam na performance das formações de *Piconet* e *Scatternet*.

Foram simuladas *Piconets* e *Scatternets Bluetooth* para demonstrar, o atraso causado pelo processo de *Discovery* de dispositivos durante sua formação inicial e o ingresso de novos escravos à *Piconet* existente.

Os modelos centralizados usando programação matemática, podem ser úteis na avaliação de performance das topologias *Scatternet* mais simples. Ao adequarmos o modelo de (Ajmone Marsan, Chiasserini, Nucci, Carello and De Giovanni 2002) a realidade observada em simulação, alterando o pesos das *bridges* e adicionando novas restrições, alcançamos resultados neste modelo válidos para base de novos trabalhos.

Também podemos concluir que, o grafo resultante da solução de nosso modelo estático de *Scatternet Bluetooth*, representa uma formação *Scatternet* com uma distribuição ideal do fluxo de dados e consumo de energia. Esta topologia resultante, é similar a obtida

por simulação do algoritmo (Law, Mehta and Siu 2002). Este modelo é provado ter complexidade de tempo é $O(\log n)$ e complexidade mensagem é de $O(n)$.

Notamos com nosso estudo, que existem espaços significativos para avanços que propõem mudanças na especificação da *baseband Bluetooth*. O crescimento da popularidade do *Bluetooth* em *Smartphones* e *Tablets*, alavancam, uma nova frente de demanda de aplicações, e algumas soluções e casos de uso *Scatternet* se encontram ainda inexplorados. Isto acontece principalmente em virtude das dificuldades de coordenação causadas pelos efeitos colaterais do uso de *FHSS*.

Como diretrizes futuras, afirmamos que nos protocolos *Bluetooth* de Formação *Scatternet*, estático ou dinâmico, os impacto do *FHSS* precisam sempre ser considerados. Isto fará, que os resultados obtidos, sejam mais reais, permitindo que exista um caminho para padronização de um modelo de *Scatternet* na especificação *Bluetooth*.

Apêndice A

Apêndice

A.1 Script NS2 - UCBT e Modelos AMPL

Disponíveis em: <http://www.decom.ufop.br/imobilis/wp-content/uploads/2012/06/anexosDissertacaoSoares.zip>

Scripts NS2 - UCBT usados para simulações de *Piconets* e *Scatternets*

- Script de simulação de Inquiry, **test_inq.tcl**
- Script de simulação de Piconet, **pico.tcl**
- Script de simulação de convergência de Scatternet usando algoritmo Law et al, **scat-form-law.tcl**

Modelo Soares et al. em AMPL de Scatternet Centralizada, tendo como base as constraints de *Marsan et al.*, **soares.mod**

Saída dos Logs:

- Log até a convergência total da Piconet, **pico.out**
- Log de processo de discovery, **test_inq.out**
- Log da simulação do algoritmo Law et al, **scat30.out**

Soluções AMPL:

- Solução do Modelo Marsan em instancia de 20 dispositivos, **solMarsan20.sol**
- Solução do Modelo Marsan em instancia de 30 dispositivos, **solMarsan30.sol**
- Solução do Modelo Modificado em instancia de 20 dispositivos, **solCelio20.sol**
- Solução do Modelo Modificado em instancia de 30 dispositivos, **solCelio30.sol**

Referências Bibliográficas

academic.ru: Last Visited in 20/05/2013, Academic dictionaries and encyclopedias.

URL: <http://de.academic.ru/dic.nsf/dewiki/181852>

Ajmone Marsan, M., Chiasserini, C. F., Nucci, A., Carello, G. and De Giovanni, L.: 2002, Optimizing the topology of bluetooth wireless personal area networks, *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Vol. 2, pp. 572–579 vol.2.

Bluetooth.com: Last Visited in 11/02/2013, The bluetooth network effect.

URL: <http://www.bluetooth.com/Pages/network-effect.aspx/>

Chiasserini, C. F., Ajmone Marsan, M., Baralis, E. and Garza, P.: 2003, Towards feasible topology formation algorithms for bluetooth-based wpan's, *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, pp. 10 pp.-.

Farinaz Edalat, Ganesh Gopal, S. M.: n.d., Bluetooth technology, <http://pt.scribd.com/doc/48331364/16645836-Bluetooth>.

Frigioni, D. and Italiano, G. F.: 1997, Dynamically switching vertices in planar graphs (extended abstract), *Proceedings of the 5th Annual European Symposium on Algorithms*, Springer-Verlag, London, UK, pp. 186–199.

Gupta, P. and Kumar, P.: 2000, The capacity of wireless networks, *Information Theory, IEEE Transactions on* **46**(2), 388 –404.

IBM: Last Visited in 11/02/2013, Ibm ilog cplex optimizer.

URL: <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>

Jedda, A., Jourdan, G.-V. and Zaguia, N.: 2010, Some side effects of fhss on bluetooth networks distributed algorithms, *Proceedings of the ACS/IEEE International*

- Conference on Computer Systems and Applications - AICCSA 2010*, AICCSA '10, IEEE Computer Society, Washington, DC, USA, pp. 1–8.
- Jeddah, A., Zaguia, N. and Jourdan, G.-V.: 2009, Analyzing the device discovery phase of bluetooth scatternet formation algorithms, *Distributed Computing Systems Workshops, 2009. ICDCS Workshops '09. 29th IEEE International Conference on*, pp. 468–471.
- Kui, M. and Xiuying, C.: 2003, Research of bluetooth security manager, *Neural Networks and Signal Processing, 2003. Proceedings of the 2003 International Conference on*, Vol. 2, pp. 1681–1684 Vol.2.
- Law, C., Mehta, A. K. and Siu, K.-Y.: 2001, Performance of a new bluetooth scatternet formation protocol, *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '01, ACM, New York, NY, USA, pp. 183–192.
- Law, C., Mehta, A. K. and Siu, K.-Y.: 2002, A new bluetooth scatternet formation protocol, *MobiHoc 2002, The ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Lausanne, Switzerland, pp. 183–192.
- Miluzzo, E., Lane, N. D., Fodor, K., Peterson, R., Lu, H., Musolesi, M., Eisenman, S. B., Zheng, X. and Campbell, A. T.: 2008, Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application, *Proceedings of the 6th ACM conference on Embedded network sensor systems*, SenSys '08, ACM, New York, NY, USA, pp. 337–350.
URL: <http://doi.acm.org/10.1145/1460412.1460445>
- NS2: Last Visited in 11/02/2013, The network simulator 2 - ns2.
URL: <http://www.isi.edu/nsnam/ns/>
- Peterson, B. S., Baldwin, R. O. and Kharoufeh, J. P.: 2006, Bluetooth inquiry time characterization and selection, *IEEE Transactions on Mobile Computing* **5**, 1173–1187.
- Pettarin, A., Pietracaprina, A. and Pucci, G.: 2009, On the expansion and diameter of bluetooth-like topologies, in A. Fiat and P. Sanders (eds), *Algorithms - ESA 2009*, Vol. 57 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 528–539.

- Q. Wang, D. A.: Last Visited in 11/02/2013, Ucbt - bluetooth extension for ns2.
URL: *www.cs.uc.edu/cdmc/ucbt/*
- Ramos, H. S., Guidoni, D. L., Nakamura, E. F., Boukerche, A., Frery, A. C. and Loureiro, A. A. F.: 2011, Topology-related modeling and characterization of wireless sensor networks, *PE-WASUNâ2011 ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, 2011ACMi, Miami, EUA.
- Salminen, T., Hosio, S. and Riekkii, J.: 2006, Enhancing bluetooth connectivity with rfid, *of IEEE International Conference on Pervasive Computing and Communications* **0**, 36–41.
- Salonidis, T., Bhagwat, P., Tassiulas, R. and Lammaire, R.: 2001, Proximity awareness and ad hoc network establishment in bluetooth, *Technical report*, University of Maryland.
- Soares Ferreira, C. M., Rabelo Oliveira, R. A., Gambini, H. S. and Frery, A. C.: 2013a, Characterization of fhss in wireless personal area, *WOCC 2013, The 22nd Wireless and Optical Communication Conference*.
- Soares Ferreira, C. M., Rabelo Oliveira, R. A., Gambini, H. S. and Frery, A. C.: 2013b, Static bluetooth scatternet formation models: The impact of fhss, *AICT 2013, The Ninth Advanced International Conference on Telecommunications*, pp. 25–31.
- Tahir, S., Said, A. M. and Bakhsh, S. T.: 2012a, Bluetooth network re-formation protocol for reducing path length (bnr), *Computer Information Science (ICCIS), 2012 International Conference on*, Vol. 2, pp. 755– 759.
- Tahir, S., Said, A. M. and Bakhsh, S. T.: 2012b, A bluetooth scatternet route optimization protocol, *AASRI Procedia - AASRI Conference on Power and Energy Systems*, Vol. 2, pp. 142 – 148.
- Thongthammachart, S. and Olesen, H.: 2003, Bluetooth enables in-door mobile location services, *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*, Vol. 3, pp. 2023–2027 vol.3.
- Whitaker, R. M., Hodge, L. and Chlamtac, I.: 2005, Bluetooth scatternet formation: A survey, *Ad Hoc Netw.* **3**(4), 403–450.
URL: *http://dx.doi.org/10.1016/j.adhoc.2004.02.002*

Woodings, R., Joos, D., Clifton, T. and Knutson, C.: 2002, Rapid heterogeneous ad hoc connection establishment: accelerating bluetooth inquiry using irda, *Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE*, pp. 342 – 349.

Yu, C.-M.: 2010, A study on the global configured method of blueweb routing protocol, *Proceedings of the Second international conference on Computational collective intelligence: technologies and applications - Volume Part III, ICCCI'10*, Springer-Verlag, Berlin, Heidelberg, pp. 272–281.

URL: <http://dl.acm.org/citation.cfm?id=1948072.1948108>

Índice Remissivo

- acl, 25, 26
- active, 21, 26, 27
- ad hoc, 3, 5, 15
- AFH, 17
- algoritmo dinâmico de formação scatternet, 11
- algoritmo distribuído, 8, 12
- alternate, 12
- appd, 35
- backoff, 5, 24, 41, 47
- bluemesh, 33
- bluering, 33
- bluerings, 32
- bluestar, 33
- bluetooth, 3–6, 8, 9, 12, 13, 15–19, 21–24, 26–30, 32, 34, 36, 38, 40, 41, 47, 61, 62
- bluetooth device discovery, 8, 9, 47
- bluetree, 33
- bridge, 5, 6, 8, 12, 31, 32, 34–38, 40, 51, 55, 56, 59
- bridges, 22
- bsf, 12
- btcp, 12
- cenceme, 13
- clock, 24, 41
- complexidade, 8
- componentes, 11
- comunicação inter-piconet, 4
- comunicação intra-piconet, 4
- connected, 33
- consumo de energia, 12
- cplex, 37
- dac, 25
- descoberta, 5, 23
- discovery, 12, 27, 39, 45, 47, 61
- dispositivo móvel, 13
- dispositivos móveis, 5
- dongle biométrico bluetooth, 6
- escravo, 4, 6, 9, 13, 32, 45
- espectro, 46
- fhs, 24, 25
- FHSS, 15, 17, 18
- fhss, 5, 6, 12, 30, 31, 38, 39, 41, 43, 46, 47, 56, 61, 62
- flooding, 11
- fss, 35
- gargalo de tráfego, 6
- GFSK, 17
- grafo dinâmico, 8, 9, 38, 40, 41, 56
- hid, 23
- hold, 26, 27, 34, 56, 59
- hold mode, 5
- iac, 23

- id, 23, 24
- lhs, 23
- inquiry, 23–25, 27, 32, 49
- inquiry response, 24
- inquiry scan, 23, 24, 32
- inquiry-scan, 45
- inter-piconet, 31, 34, 35, 37
- intra-piconet, 27, 31, 34, 35, 47
- irhs, 24
- ISM, 17

- jogos sociais, 3
- jump, 36

- laa, 36
- lcs, 36

- mínimo de saltos, 11
- MASTER RESPONSE, 25
- mdrp, 35
- merge, 33
- mesh, 17, 18, 29
- mestre, 4–6, 12, 13, 21, 32
- migrate, 33
- modelo centralizado de scatternet, 12
- modelo dinâmico de formação scatternet, 8, 9
- modelo estático de formação scatternet, 8, 9
- move, 33
- multi-hop, 17

- netwok simulator, 11
- ns2, 12

- otimização, 12
- overhead, 36
- overlap de canal, 18

- page, 24, 25, 32
- page hopping sequence, 25
- page response hopping sequence, 25
- page scan, 24, 25, 32
- paging, 27
- park, 21, 22, 26
- pcss, 36
- perda de energia, 6
- Piconet, 17, 32
- piconet, 4–6, 9, 11–13, 19, 21–29, 31–39, 41, 45, 49, 51, 53, 56, 59, 61
- poll, 27
- polling, 27, 34, 38
- polling cycle, 27
- pollings, 27
- ponto-multi ponto, 21
- pooling, 27
- programação matemática, 12
- protocolo de roteamento de scatternet, 11
- pseudo aleatoriedade, 12

- QOS, 15
- qos, 5, 36

- redes mesh, 3
- redes sociais *mobile*, 3
- redzvous, 35
- rendezous, 35
- round robin, 28
- rp, 35

- salto de frequência, 8, 9
- saltos de frequência, 5
- scan, 33
- scatternet, 4–6, 12, 19, 27–32, 34–38, 51–53, 57–59, 61, 62
- scheduler, 31
- scheduling, 5, 27, 31, 35, 36, 38, 47, 52

schedulling, 47
sco, 25
seek, 33
sensores, 13
SIG, 16
slave, 5
slave response, 25
slot, 27
sniff, 26
standby, 23, 24

tdd, 31
tdm, 34
tempo total de conexão, 13
time slot, 24, 35
topologia, 9, 34, 39, 51, 57
topologia otimizada, 8
topologia scatternet, 12
tráfego inter-piconet, 12
trees, 31
troca de canais, 12

ucbt, 12
udg, 31

WPAN, 15, 16, 18
wpan, 3, 4, 18, 19, 61

zigbee, 3, 17, 18
zigbee coordinator, 18
zigbee end device, 18
zigbee router, 18