

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**MODELO DE PROTOCOLO PARA
FORMAÇÃO DE REDES
*AD HOC BLUETOOTH***

MÁRCIO GARCIA MARTINS

Dissertação apresentada como requisito parcial à
obtenção do grau de Mestre em Ciência da
Computação na Pontifícia Universidade Católica do
Rio Grande do Sul

Orientador: Prof. Dr. Eduardo Augusto Bezerra

Porto Alegre, Brasil
2007

Dados Internacionais de Catalogação na Publicação (CIP)

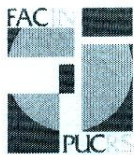
M386m Martins, Márcio Garcia
Modelo de protocolo para formação de redes ad hoc
bluetooth / Márcio Garcia Martins. – Porto Alegre, 2007.
67 f.

Diss. (Mestrado) – Fac. de Informática, PUCRS.
Orientador: Prof. Dr. Eduardo Augusto Bezerra

1. Informática. 2. Protocolos de Comunicação.
3. Bluetooth (Telecomunicações). I. Bezerra, Eduardo
Augusto. II. Título.

CDD 004.62

**Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS**



TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "**Modelo de Protocolo para Formação de Redes Ad hoc Bluetooth**", apresentada por Márcio Garcia Martins, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Sistemas Digitais e Arquitetura de Computadores, aprovada em 22/03/2007 pela Comissão Examinadora:

Prof. Dr. Eduardo Augusto Bezerra –
Orientador

PPGCC/PUCRS

Prof. Dr. Fabiano Passuelo Hessel –

PPGCC/PUCRS

Profa. Dra. Ana Cristina Benso da Silva –

FACIN/PUCRS

Prof. Dr. Jorge Luís Victória Barbosa –

UNISINOS

Homologada em...18./05./10....., conforme Ata No. 08/10..... pela Comissão Coordenadora.

Prof. Dr. Fernando Luís Doti
Coordenador.

PUCRS

Campus Central

Av. Ipiranga, 6681 – P. 32 – sala 507 – CEP: 90619-900

Fone: (51) 3320-3611 – Fax (51) 3320-3621

E-mail: ppgcc@inf.pucrs.br

www.pucrs.br/facin/pos

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus pelos dons, pela família maravilhosa e amigos que me deu, fazendo deles a ponte e apoio que possibilitaram alcançar todos os meus objetivos. Obrigado senhor pela companhia nas noites de solidão e por me carregar nos braços nos momentos em que parecia não haver saída.

Agradeço também a toda a minha família, iniciando pelo meu pai Alexandre, pelo exemplo de fé, coragem e amor pela família que ele demonstra a cada dia mesmo com as dificuldades que a vida lhe impôs, a minha mãe Luci uma fortaleza e abrigo em meio ao mar de tempestades da vida, a minha irmã Letícia pelo incentivo e carinho em todos os momentos e meu cunhado Alex pelo exemplo de alegria e paz que ele representa. E também a minha vó Nadir por todas as noites sem dormir rezando por mim e pelo apoio financeiro indispensável para conclusão de mais essa fase da minha vida.

Agradeço agora aos meus amigos, em especial a Gabriel Almeida, Márcio Farias, Luis Henrique Ries, Jordana Oliveira, Rafael Krolow e Fábio Cecin pelo incentivo, força e horas de conversas e risadas que foram tão importantes para vencer todas as barreiras impostas em mais essa etapa vencida.

Aos professores Jorge Barbosa e meu orientador Eduardo Bezerra, ao primeiro pela amizade demonstrada desde os tempos de graduação e força em momentos difíceis que ocorreram no decorrer do curso e ao segundo pela compreensão nesses mesmos momentos o que me possibilitou a conclusão deste trabalho.

Também agradeço aqueles ao qual me deram suporte acadêmico e financeiro para que esse trabalho pudesse ser concluído. Inicialmente agradeço a PUC por me prover o suporte acadêmico necessário para contribuir com a minha formação. Obrigado ao CNPq por me prover o suporte financeiro necessário para o desenvolvimento do trabalho durante esse período.

Enfim, agradeço a todos aqueles que direta ou indiretamente contribuíram para que esse trabalho pudesse ter sido realizado e concluído.

MODELO DE PROTOCOLO PARA FORMAÇÃO DE REDES *AD HOC* *BLUETOOTH*

RESUMO

Este trabalho tem como principal objetivo a investigação de protocolos para formação de redes ad hoc, visando a modelagem de novo protocolo para formação de scatternets Bluetooth. Esse protocolo tem por característica principal atender os requisitos de aplicações interativas, que dependem mais de uma maior disponibilidade da conexão do que de uma conexão com largura de banda elevada e com baixo atraso de comunicação. O protocolo modelado mantém conexões funcionando mesmo na presença de eventos de entrada e saída de nós da rede móvel. O protocolo foi proposto visando a implementação de um SoC Bluetooth para a formação de scatternets. A validação inicialmente foi idealizada para utilização da plataforma Fênix do projeto Brazil IP, sendo ao final realizada uma simulação para levantamento de resultados do modelo para geração de scatternets.

Palavras-chave: *Redes Ad Hoc; Bluetooth; Scatternet.*

PROTOCOL FOR BLUETOOTH AD HOC NETWORK FORMATION

ABSTRACT

This work's main goal is the research of protocols for ad hoc network formation, targeting the modeling of a scheme for building a new Bluetooth scatternet protocol. This protocol has as its main feature to be compliant with the requirements of interactive applications, which are more dependent on a better connection availability than on large bandwidth with low communication delay. The modeled protocol keeps the connection up, even in the events of nodes going in and out the mobile network. The protocol has been proposed targeting a Bluetooth SoC implementation for scatternet formation. Initially, the validation strategy had been conceived targeting the IP's Fênix Platform. A simulation was performed in order to obtain experimental data from the proposed model for scatternet's generation.

Keywords: Ad Hoc Networks; Bluetooth; Scatternet.

SUMÁRIO

1 INTRODUÇÃO	15
1.1 Conceitos Básicos de Grafos	17
1.2 Redes <i>Wireless</i>	19
1.3 Tecnologia <i>Bluetooth</i> na Formação de Redes <i>Ad hoc</i>	21
1.4 Objetivos e Contribuição	22
1.5 Organização do texto	24
2 ESTADO DA ARTE	25
2.1 Descoberta de Dispositivos em Redes <i>Bluetooth Multi-hop</i>	25
2.2 <i>BlueTrees</i>	26
2.3 <i>BlueNet</i>	28
2.4 <i>BlueStars</i>	32
2.5 Protocolo <i>LSBS</i>	33
2.6 Considerações Finais sobre o Estado da Arte	34
3 APRESENTAÇÃO DO MODELO PROPOSTO	37
3.1 Conceitos Básicos da Tecnologia <i>Bluetooth</i>	37
3.2 Estudo de Caso	39
3.3 Modelo Proposto	40
3.3.1 Fase de Descoberta de Dispositivos	40
3.3.2 Fase de Formação de <i>Piconets</i>	41
3.3.3 Fase de Formação de <i>Scatternets</i>	45
3.3.4 Problema: Queda de Ponte	46
4 ESTRATÉGIAS PARA IMPLEMENTAÇÃO DO MODELO	49
4.1 Projeto de SoC para Formação de <i>Scatternets</i>	49
4.2 Simulação do Modelo de Protocolo Proposto	50

5 EXPERIMENTO E ANÁLISE DOS RESULTADOS	55
6 CONCLUSÕES	63
REFERÊNCIAS BIBLIOGRÁFICAS	65

Capítulo 1

INTRODUÇÃO

Nos últimos anos assistiu-se a uma proliferação de tecnologias no segmento de redes locais sem fio, (*Wireless Local Area Network - WLAN*) tais como *Bluetooth* [BLU04] e *WiMAX* [WEL05]. Uma consequência é o aumento considerável de usuários carregando consigo dispositivos portáteis tais como notebooks, telefones celulares e PDA's para uso profissional e/ou pessoal. Todos estes equipamentos oferecem uma solução atraente para aqueles que pretendem uma rápida e simples instalação sem os problemas associados à conexão via cabo.

Entretanto, a maioria destes dispositivos não se comunica entre si. Neste contexto foram criadas as redes móveis sem fio que permitem aos dispositivos portáteis comunicar-se e interagir entre si. As redes móveis sem fio dividem-se, basicamente, em dois tipos [BAS04]: as redes infra-estruturadas e as redes *ad hoc* (*Mobile Ad Hoc Network - MANET*).

As redes infra-estruturadas [BAS04] são aquelas em que o dispositivo móvel está em contato direto com uma estação de suporte à mobilidade, também conhecida como ponto de acesso a uma rede fixa. O funcionamento deste tipo de rede móvel é semelhante ao da telefonia celular, onde toda a comunicação deve, necessariamente, passar pela central, mesmo que os equipamentos móveis estejam a uma distância em que poderiam, eventualmente, comunicar-se diretamente.

Sendo assim nas redes infra-estruturadas a comunicação entre os nós móveis é feita através de estações de suporte à mobilidade. Neste caso, os nós móveis, mesmo próximos uns dos outros, estão impossibilitados de realizar qualquer tipo de comunicação direta.

Já nas redes *ad hoc* [BAS04] os dispositivos são capazes de trocar informações diretamente entre si. Ao contrário do que ocorre em redes convencionais, não há pontos de acesso, ou seja, não existem estações de suporte à mobilidade (sem infra-estrutura de conexão) e os nós dependem uns dos outros para manter a rede conectada. Por esse motivo, redes *ad hoc* são indicadas principalmente em situações onde não é possível, ou não faz sentido, instalar uma rede fixa.

Sendo uma rede *wireless* os nós de uma rede *ad hoc* podem se mover arbitrariamente. Deste modo, a topologia da rede muda frequentemente e de forma imprevisível. Assim, a conectividade entre os nós móveis muda constantemente, requerendo uma permanente adaptação e reconfiguração

de rotas. Como já comentado a utilização de uma rede *ad hoc* está associada a cenários onde exista uma necessidade de se instalar rapidamente uma rede de comunicação. Normalmente, são situações onde não há uma infra-estrutura de rede previamente instalada. Algumas das aplicações possíveis para redes tipo *ad hoc* são:

- Coordenação de resgates em situações de desastre;
- Troca de informações táticas em campos de batalha;
- Compartilhamento de informações em reuniões e salas de aula.

As principais vantagens da utilização de redes *ad hoc* são:

- Rápida instalação, pois esse tipo de rede pode ser estabelecido dinamicamente em locais onde não haja previamente uma infra-estrutura instalada;
- Pode-se considerar que redes *ad hoc* possuem características de sistemas tolerantes a falha, pois a permanente adaptação e reconfiguração das rotas nessas redes permitem que perdas de conectividade entre os nós possam ser facilmente resolvidas desde que uma nova rota possa ser estabelecida;
- Apresenta também acréscimos em relação a conectividade, pois nesse modelo de rede dois nós móveis podem se comunicar diretamente desde de que cada nó esteja dentro da área de alcance do outro.

Já as desvantagens principais a serem citadas são [WEL05]:

- O roteamento, pois a mobilidade dos nós e uma topologia de rede dinâmica contribuem diretamente para tornar a construção de algoritmos de roteamento um dos principais desafios em redes *ad hoc*;
- A taxa de erros associada a enlaces sem-fio é mais elevada quando comparada aos enlaces em redes estruturadas;
- A banda passante que com cabeamento convencional pode chegar a 1Gbps, nos enlaces via redes wireless tem-se taxas de até 2 Mbps tipicamente;
- E por último uma questão importante em redes *ad hoc* é a localização de um nó, pois além do endereço da máquina não ter relação com a posição atual do nó, também não há informações geográficas que auxiliem na determinação do posicionamento desse nó.

Convém salientar ainda que o protocolo de comunicação da rede móvel deve considerar certas características que não ocorrem em uma rede fixa. É importante lembrar que os equipamentos móveis de uma rede *ad hoc* apresentam várias limitações de recursos, principalmente largura de banda e autonomia de bateria [BAS04].

Nas próximas seções serão expostas a teoria de grafos (seção 1.1), importante para o entendimento dos protocolos de formação de *Scatternet*, outras tecnologias de redes *wireless* (seção 1.2) e a tecnologia *Bluetooth* para formação de redes *ad hoc* (seção 1.3).

1.1 Conceitos Básicos de Grafos

Esta seção tem por finalidade descrever conceitos básicos da teoria dos grafos [COR90], pois é através dos grafos que são representadas as redes *ad hoc* presentes nesse trabalho.

Grafo é uma noção simples, abstrata e intuitiva, usada para representar a idéia de alguma espécie de relação entre "objetos". Graficamente, aparece representado por uma figura com nós ou vértices, significando os objetos, unidos por um traço denominado aresta ou arco configurando a relação imaginada. A Figura 1.1 mostra a representação de um grafo.

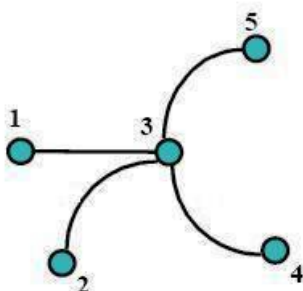


Figura 1.1: Representação de um grafo

Os grafos podem ser dirigidos ou não-dirigidos. Um grafo dirigido G é um par (N, A) , onde N é um conjunto finito e A é uma relação binária entre os componentes de N . Onde N é o conjunto de nós (ou vértices) de G e A conjunto de arcos (ou arestas) de G . Enquanto em um grafo não-dirigido $G = (N, A)$, o conjunto de arcos A consiste de um conjunto desordenado de pares de nós N .

A Figura 1.2 representa um grafo dirigido onde N é o conjunto de nós 1,2,3,4,5,6 e A é o conjunto arcos $(1,2),(2,2),(2,4),(2,5),(4,1),(4,5),(5,4),(6,3)$. Enquanto a Figura 1.3 representa um grafo não-dirigido onde N é o conjunto de nós 1,2,3,4,5,6 e A é o conjunto de arcos $(1,2),(1,5),(2,5),(3,6)$.

Dois nós são ditos adjacentes se existe um arco interligando-o. Por exemplo, no grafo dirigido representado pela Figura 1.2 o nó 2 é adjacente ao nó 1, enquanto o nó 1 não é adjacente ao nó 2 pois o arco $(2,1)$ não pertence ao grafo. Já no grafo não-dirigido representado pela Figura 1.3 a relação de adjacência é simétrica, ou seja, o nó 2 é adjacente ao nó 1 e o nó 1 é adjacente ao nó 2.

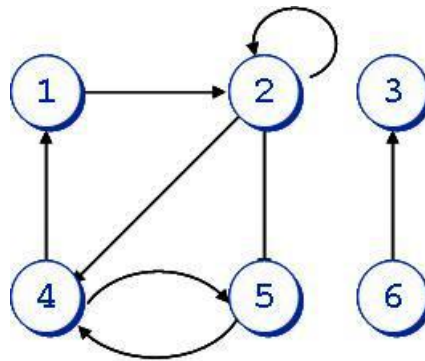


Figura 1.2: Representação de um grafo dirigido

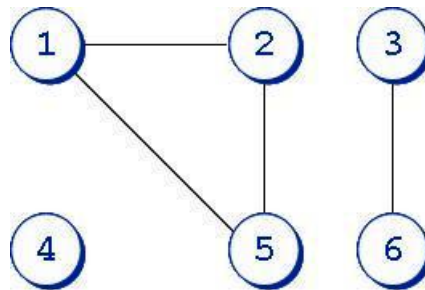


Figura 1.3: Representação de um grafo não-dirigido

Grau de um nodo: é medido pelo número de arcos incidentes. Em um grafo dirigido o grau é obtido pela soma do grau de Saída, ou seja, arcos que saem do nó e do grau de Entrada, que são aqueles que chegam no nó. Na Figura 1.2 o nó 1 possui grau 2 (1 + 1), enquanto o nó 2 possui grau 5 (3 + 2). Nos grafos não-dirigidos o grau é verificado pelo número de arestas que se relacionam com um nó, por exemplo, na Figura 1.3 o nó 2 possui grau 2, o nó 6 possui grau 1 e o nó 4 possui grau 0.

Um caminho c de um nó u para um nodo u' em um grafo $G = (N, A)$ é a seqüência de nodos $\{n_0, n_1, n_2, \dots, n_k\}$, onde $u = n_0$ e $u' = n_k$, e (n_{i-1}, n_i) pertence a A para $i = 1, 2, \dots, k$. O tamanho do caminho c é o número de arcos existente em c . Por exemplo na Figura 1.2, existem dois caminhos entre os nós 1 e 4 esses caminhos são c_1 (1, 2, 5, 4) e c_2 (1, 2, 4) e os tamanhos desses caminhos são c_1 igual a 3, pois a esse caminho pertencem os arcos (1,2), (2,5) e (5,4) e c_2 igual a 2, pois desse caminho fazem parte os arcos (1,2) (2,4).

Outro conceito citado neste trabalho é o de grafos conectados. Um grafo não-dirigido $G = (N, A)$ é conectado se cada par de nós é conectado por um caminho, ou seja, um grafo não-dirigido é conectado se possuir exatamente um componente conectado, ou seja, se cada nó é alcançável a partir de cada um dos outros nós. Tendo como exemplo a Figura 1.3 se divide em três componentes conectados que os nós (1, 2, 3), os nós (3,6) e o nó 4, resultando assim em um grafo não conectado.

O último conceito da teoria dos grafos importante para o entendimento do modelo proposto é

o de árvore. Uma árvore é uma estrutura de dados que possui uma relação hierárquica entre seus elementos. Trata-se de um conjunto finito de um ou mais nós, onde um deles é denominado raiz e os demais, recursivamente, formam uma sub-árvore. Quando num determinado nível a sub-árvore tem um único nó ele é chamado de folha. A árvore da Figura 1.4 tem as seguintes folhas: D, E, G, H e I.

Nesta seção foram introduzidos alguns conceitos básicos de grafos que serão de grande importância para o entendimento, tanto dos protocolos de formação de *scatternet* estudados quanto do modelo proposto.

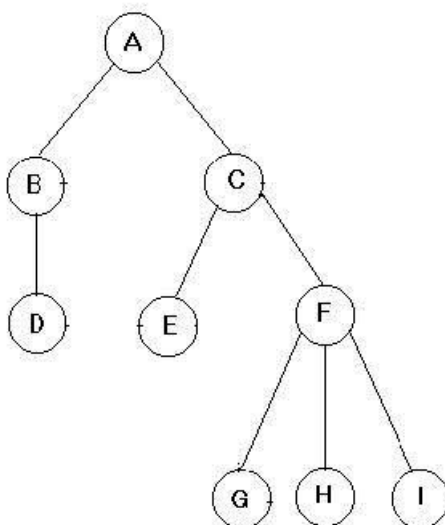


Figura 1.4: Representação de um grafo em forma de árvore

1.2 Redes *Wireless*

Nesta seção serão descritos alguns modelos de redes *wireless* com características semelhantes as redes formadas pelo modelo proposto.

Uma Rede de Sensores Sem Fio (RSSF) pode ser definida como um tipo especial de rede móvel *ad hoc*, podendo ser utilizada para monitorar e, eventualmente, realizar o controle no ambiente no qual se encontra.

É notório que a tecnologia de Rede de Sensores Sem Fio vem crescendo rapidamente, e com isso a sua aplicabilidade está sendo utilizada em diversas áreas como [THO04][MAR04][WOO04][SZE04]: indústria (controle de processo), monitoração de ambientes, segurança, medicina (monitoração de órgãos), biologia (detectar substâncias impróprias) e militar (detectar inimigos ou materiais indesejáveis).

Para que tenhamos uma Rede de Sensores Sem Fio é necessário um conjunto de dispositivos compactados e automatizados chamados de nós sensores. Um nó sensor [HIL04] é um pequeno dispositivo computacional cujo objetivo, dentro da rede, é captar informações (através de seus sensores),

realizar algum tipo de processamento e transmití-las para que se possa realizar o monitoramento físico de um fenômeno ambiental. Esse mesmo sensor produzirá uma resposta mensurável a mudanças em condições físicas, tais como temperatura, campo magnético e luz.

As Redes de Sensores são constituídas, basicamente, por três características: o sensor, dispositivo que realiza a monitoração de uma área física, através da observação de um fenômeno; o observador, usuário interessado nas informações coletadas pelo sensor; e o fenômeno, entidade de interesse do observador (como por exemplo uma mudança de temperatura).

A classificação de uma Rede de Sensores depende de seu objetivo e área de aplicação. A aplicação influenciará diretamente nas funções exercidas pelos nós da rede, assim como na arquitetura desses nós (processador, memória, dispositivos sensores, fonte de energia, transceptor), na quantidade de nós que compõem a rede, na distribuição inicialmente planejada para a rede, na escolha dos protocolos da pilha de comunicação, no tipo de dado que será tratado, no tipo de serviço que será provido pela rede e conseqüentemente no tempo de vida dessa rede.

De acordo com [PER04] [BUL04], as Redes de Sensores podem ser classificadas segundo a configuração, o sensoriamento e segundo o tipo de comunicação. Uma Rede de Sensores também pode ser classificada segundo o tipo de processamento que executa.

Outro tipo de rede wireless é a *Wireless Mesh Network (WMN)* [RUI03][AKY05], que foi desenvolvida a partir da necessidade de fornecer os serviços já oferecidos pelas redes *wireless* com maior qualidade e não dependente das redes fixas. Neste tipo de rede os nós roteiam o tráfego entre si dinamicamente.

Por apresentar a característica de roteamento entre os nós de forma dinâmica, estas redes assemelham-se as redes móveis *ad hoc*. A principal diferença das redes *mesh* para as redes *ad hoc* é que nas redes *mesh* os nós possuem localização fixa, apesar da localização não ser predeterminada, por conseqüência disto os algoritmos de roteamento apresentam muita semelhança entre si.

Devido à localização dos nós serem conhecidas, as redes *mesh*, também conhecidas como rede de malha sem fio, podem ser utilizadas em diversas aplicações como *backup* distribuído, cache de conteúdo da Internet e divulgação de informação de relevância local.

As redes *mesh* apresentam como vantagens o custo, robustez, escalabilidade, simplicidade, além das vantagens da própria tecnologia *wireless*. Redes *mesh* são ditas robustas devido às mesmas apresentarem a característica dinâmica, onde ao modificar um *link* com a adição ou remoção de um nó, ou mesmo através de interferência no *link*, a rede se adapta sem interferência humana [BRU05][SUB06].

Em conseqüência desta característica a rede torna-se mais robusta do que uma rede convencional, pois a quebra de um nó ou de um *link* não implica em quebra da conectividade da rede, por conseqüência dos nós estarem conectados a mais de um nó.

Portanto o custo das redes cai significativamente devido a não necessidade de manutenção na rede, além do custo dos *links* cabeados poderem incluir a necessidade de obras caras, trabalhosas e inconvenientes.

Porém as redes *mesh* apresentam as seguintes desvantagens [BRU05][SUB06]: degradação da

largura de banda, interferência eletromagnética, desvantagem inerente a todo tipo de rede *wireless*, e perda freqüente de pacotes. Na degradação da largura de banda, os nós de uma rede sem fio precisam utilizar a mesma freqüência, devido a isto somente um nó pode transmitir de cada vez. Nas arquiteturas convencionais, N nós comunicam-se apenas com um nó, que está ligado a uma rede fixa, nesta situação a largura de banda é dividida por N. Nas redes *mesh* este problema aumenta, pois cada nó pode usar 1/N do tempo para transmitir os pacotes ao nó mais próximo.

Um outro tipo de redes *wireless ad hoc* são as redes formadas pela tecnologia *Bluetooth*, essas redes possuem características diferentes das demais para sua formação devido a especificação da tecnologia *Bluetooth*. Na próxima seção será abordada a tecnologia *Bluetooth* na formação de redes *ad hoc*.

1.3 Tecnologia *Bluetooth* na Formação de Redes *Ad hoc*

A tecnologia *Bluetooth* [BLU04] como descrita em sua especificação é considerada promissora para construção de redes *ad hoc* [HSI06][CHI03]. Com a idéia inicial de permitir a substituição de cabos, a especificação define maneiras para que cada dispositivo *Bluetooth* possa conectar-se a múltiplos dispositivos através de uma comunicação *multi-hop*, ou seja onde a comunicação se dá com a utilização de intermediários, pois os nós podem não estar na área de transmissão um do outro. A maneira pela qual a tecnologia permite uma comunicação *multi-hop* é descrita como segue: quando dois nós *Bluetooth* querem estabelecer uma ligação, um deles deverá assumir o papel do mestre da comunicação enquanto o outro se transforma em seu escravo. Esta rede simples *one-hop* é chamada de *piconet* e pode incluir diversos escravos que podem ativamente se comunicar com o mestre ao mesmo tempo. Se um mestre tiver mais de 7 escravos, alguns escravos têm que ser estacionados, ou seja, colocados em estado de espera e ativados apenas no caso da saída de um dos 7 escravos em comunicação com o mestre. A especificação [BLU04] permite que cada nó possua papéis múltiplos. Um nó pode ser um mestre em uma *piconet* e um escravo em uma ou em mais *piconets*, ou um escravo em múltiplas *piconets*. Os dispositivos com múltiplos papéis agem como passagens as *piconets* adjacentes que dão assim forma a uma rede *ad hoc multi-hop* chamada *scatternet*. A propagação dos dispositivos com *Bluetooth* em uma área geográfica pode fornecer a extensão *wireless* que falta as várias infra-estruturas de redes *wireless* heterogêneas. O problema fundamental da formação de *scatternets*, é o de organizar os dispositivos *Bluetooth* em uma rede *multihop* [BAS04].

A formação de *scatternets* não está definida formalmente na especificação do grupo de interesse especial de *Bluetooth*(SIG). Em conseqüência, diversos protocolos foram propostos. No presente trabalho é realizada uma investigação e definição de um protocolo para formação de redes *ad hoc*, vislumbrando a criação de *scatternets*.

Como citado anteriormente a especificação atual do *Bluetooth* não prevê um protocolo para formação de *scatternet* [BAS04]. Na Figura 1.5 é mostrado um cenário onde com a utilização de um protocolo de formação de *scatternet* uma rede *ad hoc Bluetooth* poderia ser obtida. A Figura 1.5 mostra o caso

onde 13 dispositivos Bluetooth foram divididos em quatro *piconets* (a, b, c e d).

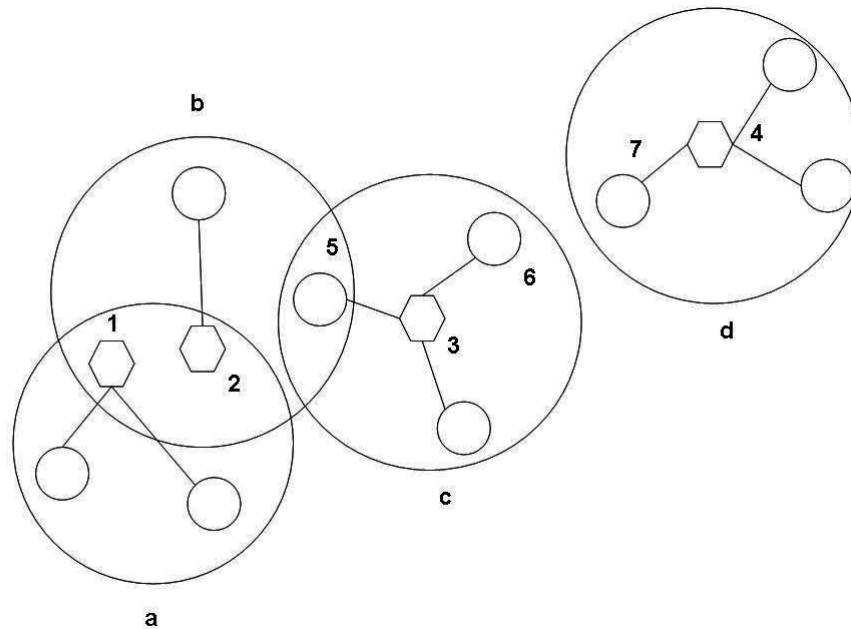


Figura 1.5: 13 dispositivos *Bluetooth* formando 4 *piconets*

Os mestres são representados pelos pentágonos (cercados por um círculo grande que represente seu raio da transmissão), enquanto os escravos são descritos como círculos pequenos. As *piconets* adjacentes podem ser interconectadas de maneiras diferentes. As *piconets* a e b descrevem a lógica do mestre-mestre, quando dois mestres são vizinhos e a interconexão é feita tendo um dos dois mestres juntando a *piconet* do outro como escravo (na Figura 1.6, o nó 2 se transformou em escravo do nó 1). Duas *piconets* podem ser agrupadas por um escravo comum, denominado como escravo de passagem. Esta é a lógica de interconexão das *piconets* b e c que são agrupadas pelo nó 5. O terceiro caso é quando as *piconets* são interconectadas com um par de escravos de borda, chamado de passagens intermediárias, lógica utilizada para interconexão das *piconets* c e d, agrupada pelos nós 6 e 7. No último caso, a interconexão requer que uma das duas passagens intermediárias se transforme em mestre de uma nova *piconet* que inclua a outra passagem intermediária como escravo (na Figura 1.6, o nó 7 se transforma em mestre da *piconet* extra). Com a criação da *piconet* e, as cinco *piconets* da Figura 1.6 dão forma a uma *scatternet* conectada.

1.4 Objetivos e Contribuição

O trabalho proposto tem por objetivo principal a investigação de protocolos para formação de redes *ad hoc*, visando a geração de tecnologia com capacidade para formação de *scatternets*. Esse protocolo deverá, principalmente, manter a conexão independente da resiliência, ou seja, eventos de

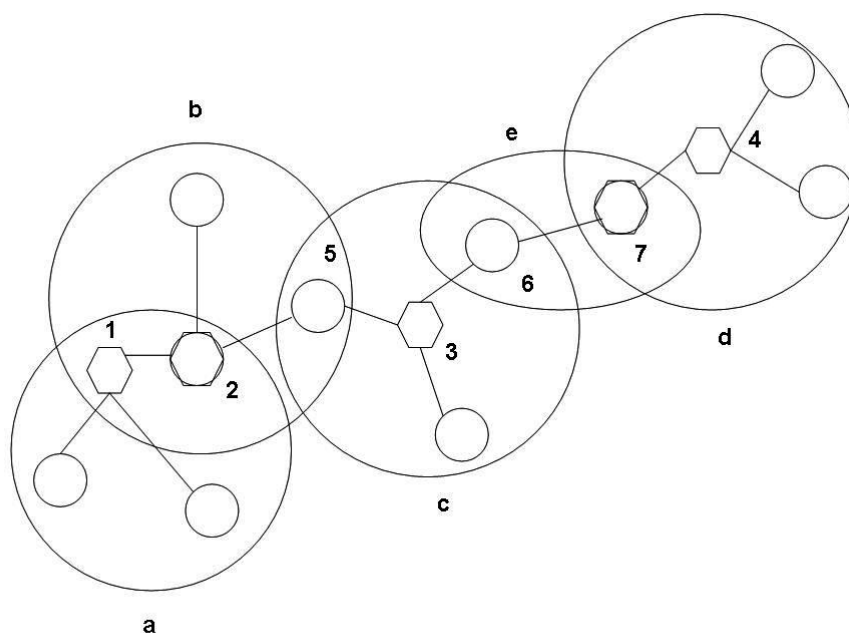


Figura 1.6: Rede *ad hoc bluetooth* (*scatternet* montada a partir das 4 *piconets* da Figura 1.5)

entrada e saída de nós na rede móvel, e que possibilite em trabalhos futuros a utilização da tecnologia *Bluetooth* para formação de *redes ad hoc*. Os objetivos específicos são os seguintes:

1. Estudo de redes *ad hoc* e características importantes como vantagens, desvantagens, e desempenho para utilização em cenários de uso onde não é possível e/ou estas redes se adaptem melhor que as redes infra-estruturadas;
2. Estudo da tecnologia *Bluetooth*, que entre as tecnologias do segmento de redes locais sem fio disponíveis é de maior interesse do grupo de pesquisa em que esse projeto está inserido;
3. Estudo de protocolos para formação de *scatternets* já propostos, visando melhor entendimento do que pode ser feito para manter a conexão da rede, mesmo quando sujeito a efeitos de resiliência na rede móvel;
4. Estudo da tecnologia *Bluetooth* e suas limitações e problemas para formação de *scatternets*;
5. Modelagem do protocolo para formação de *scatternets*;
6. Validação do protocolo.

A contribuição deste trabalho é a criação de modelo de protocolo de formação de *scatternets*. Este modelo é baseado em protocolo existente, porém otimizado para aplicações como jogos, onde o grafo de visibilidade dos nós se aproxima de um grafo completo, ou seja, a maioria dos nós está dentro

de área de transmissão de grande parte dos outros. No tipo de aplicação como a do estudo de caso (detalhado na Seção 3.1) é importante eliminar ou reduzir drasticamente o tempo em que a rede está se reconstruindo devido a entrada e saída de nós durante uma sessão de jogo, o que é desconsiderado pelo modelo existente original. Ou seja, o protocolo proposto visa aumentar a resiliência da rede frente a desconexões para que a aplicação seja interrompida o mínimo possível dentro de um período de tempo determinado que se aproxima de uma seção casual de jogo, por exemplo a partir de 20 min.

1.5 Organização do texto

O restante do documento está organizado e estruturado da seguinte forma: o Capítulo 2 apresenta o estado da arte da pesquisa de protocolos de formação de *scatternet*. O Capítulo 3 detalha o modelo desenvolvido no presente trabalho. No Capítulo 4 é explicado o projeto e implementação da simulação do modelo proposto, bem como uma idéia proposta para trabalhos futuros visando a validação do protocolo definido nessa dissertação na Plataforma *Fênix* do Projeto *Brazil IP*. O Capítulo 5 mostra os resultados obtidos na simulação. No Capítulo 6 são apresentadas as conclusões do presente trabalho.

Capítulo 2

ESTADO DA ARTE

Os protocolos de formação de *scatternets* são classificados basicamente em protocolos *single-hop*, onde todos os nós estão dentro da área de transmissão um do outro e nos protocolos *multi-hop* já descritos anteriormente. Os protocolos *single-hop* foram os primeiros propostos entre eles alguns podem ser destacados. O proposto em [SAL01] é baseado na eleição de um líder que coleta a informação de topologia e baseado nesta informação, decide qual será o papel de todos os outros nós. Para obter boas propriedades na *scatternet* resultante, o número de nós tem que ser $n \leq 36$, no caso quando $n > 36$ outros esquemas centralizados podem ser usados como os propostos em [MAR02] e [BAA02]. Os protocolos descritos em [BAA02] e [LAW03] executam sobre topologias *single-hop* sem limitação no número de nós suportados, porém, a *scatternet* resultante é uma árvore, o que limita a eficiência e a robustez. Como as soluções *single-hop* tem uma premissa muito restritiva, passou-se a trabalhar com protocolos *multi-hop* esta seção aborda detalhadamente os protocolos *BlueTrees* [ZAR01], *BlueStars* [PET03], *BlueNet* [WAN02] e *LSBS Protocol*[LI04].

2.1 Descoberta de Dispositivos em Redes *Bluetooth Multi-hop*

Em [BAS04] é descrito que a fase da descoberta de dispositivos deve fazer com que cada um dos nós da rede se torne consciente dos nós que estão no seu alcance de transmissão. Se um nó v conhece um nó u , o nó u também deve conhecer o nó v (todos os protocolos de formação de *scatternets multi-hop* dependem disso). Para entendimento do conhecimento mútuo a Figura 2.1 mostra os nós u e v e suas áreas de transmissão, que são representadas pelos círculos pontilhados. Os mecanismos providos pelas especificações de *Bluetooth* para descoberta de dispositivos, isto é, os procedimentos de *inquiry* [BLU04], não levam ao necessário conhecimento simétrico sobre vizinhos. Os nós que estão usando esse procedimento, *Inquiry*, não transmitem o seu identificador único *Bluetooth*, e portanto se mantém desconhecidos ao nó que recebe o pacote de *inquiry*. Além disso, os mecanismos de descoberta do *Bluetooth* requerem que os nós estejam em modos opostos de *inquiry* (chamados de modos *inquiry*

e *inquiry scan*) para que seja possível a comunicação entre eles. Porém, nenhum método é descrito na especificação do *Bluetooth* sobre como garantir que dois nós vizinhos estão em modos opostos. Portanto, mecanismos aderentes à especificação precisam ser definidos para garantir que, para cada par de nós v e u , eles estão eventualmente em modos opostos e que, quando o nó v descobre o nó u , o nó u também se torna consciente do nó v .

Na solução proposta em [ZAR01] há uma descrição detalhada das operações de descoberta em um cenário *multi-hop*. A cada dispositivo é permitida a alternância entre os modos *inquiry* e *inquiry scan*, permanecendo em cada modo por um tempo selecionado aleatoriamente e uniformemente em um dado intervalo. O processo todo é realizado por um tempo pré-definido de descoberta de dispositivo. As operações realizadas, enquanto o dispositivo está em cada um dos dois modos, são aquelas descritas nas especificações.

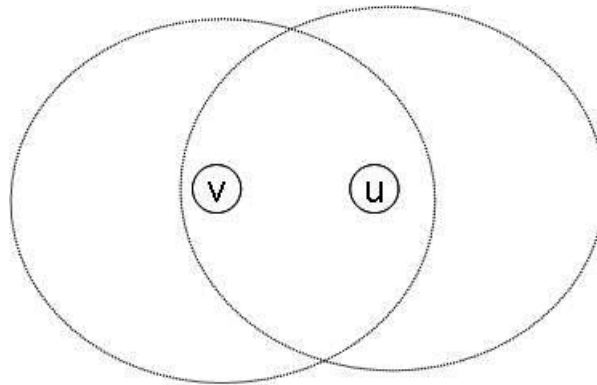


Figura 2.1: Condição importante: nós vizinhos devem se conhecer mutuamente

Quando dois nós em modos opostos de *inquiry* fazem o *handshake*, isto é, quando esses nós estabelecem uma conexão, eles criam uma *piconet* temporária que dura apenas o tempo necessário para que eles troquem os seus IDs e possivelmente outra informação necessária para as fases seguintes do protocolo. A formação de *piconets* temporárias e a troca de informações é feita para alcançar o requerido conhecimento mútuo.

Foi verificado em [ZAR01] que pode-se manter a fase de descoberta de dispositivos razoavelmente curta nos melhores casos fica na ordem de 28 segundos, e ao mesmo tempo obter uma topologia de descoberta conectada, que é um requisito necessário para geração de *scatternets* conectadas. O preço a pagar é que, na topologia descoberta, não é garantido que dois nós que estão no alcance de transmissão uns dos outros se descobrirão, pois quanto menor é tempo definido para duração da fase de descoberta, menor a chance de descoberta de todos os vizinhos.

2.2 *BlueTrees*

O protocolo de formação de *scatternet* apresentado em [ZAR01] é o primeiro a resolver o problema do estabelecimento de rede *scatternet* em uma topologia *multi-hop*. O protocolo é iniciado por

um nó designado, chamado de *blueroot* e gera uma *scatternet* em forma de árvore.

O *blueroot* começa o procedimento de formação ao adquirir como escravos os seus vizinhos *one-hop*. Estes, por sua vez, começam a tentar conectar com seus próprios vizinhos, os nós que estão a até dois *hops* da raiz, e assim sucessivamente, em uma forma expansão de onda, até que toda a árvore esteja construída. Para limitar o número de escravos por *piconet*, é observado que no caso de um nó possuir mais de 5 vizinhos, então pelo menos dois deles estão no alcance de transmissão um do outro. Esta observação é usada para re-configurar a árvore de uma forma em que cada nó mestre não tenha mais do que sete escravos. Se um mestre *v* tem mais do que sete escravos, ele seleciona dois deles que estão necessariamente no alcance de transmissão um do outro, e instrui um dos dois para ser o mestre do outro, que então é desconectado da *piconet* de *v*. Este tipo de "reorganização de galhos" é feita pela rede, levando a uma *scatternet* onde cada *piconet* não tem mais do que 7 escravos.

Além de produzir uma topologia similar a uma árvore, o que limita a robustez da *scatternet* obtida, o *BlueTrees* depende de um nó selecionado para iniciar o procedimento de formação, de forma que esta solução não funciona em redes cujas topologias após a fase de descoberta não são conexas.

Foram encontrados dois problemas ao implementar o *BlueTrees*. Para limitar o número de escravos por *piconet* para um número ≤ 7 , o protocolo assume que a topologia resultante da descoberta de dispositivos é de tal forma que se dois nós vizinhos *u* e *v* se descobrirem mutuamente e os dois tem um vizinho comum *z*, então ou os dois descobriram *z* ou nenhum deles descobriu o *z*. Esta propriedade, que é necessária para uma reconfiguração do *BlueTrees* em uma *scatternet* com não mais do que 7 escravos por *piconet*, não pode ser garantida pela fase de descoberta que foi descrita anteriormente.

Considere, por exemplo, uma topologia similar à uma estrela com 9 nós onde o nó central *v* se torna mestre e escolhe 7 entre os seus 8 vizinhos. Considere que o vizinho *z* é o que foi deixado de fora da *piconet* de *v*. Para a propriedade geométrica mencionada acima, ao menos um dos vizinhos de *v*, digamos *u*, está no alcance de transmissão de *z*. Se *u* e *z* não se descobrirem mutuamente, é provável que a *scatternet* formada não é conexa (*v* não consegue alcançar o seu vizinho *z*). Portanto, para que o protocolo funcione corretamente é necessário realizar operações extras para alcançar um conhecimento consistente sobre a vizinhança de cada nó. Estas operações, que visam estabelecer alguns dos *links* que não foram descobertos, é chamada de Fase de Reabastecimento do protocolo. Esta fase pode ser implementada da seguinte maneira: ao final da fase de descoberta, todos os nós que se descobrirem mutuamente trocam entre si as listas de nós que cada um acabou de descobrir. Esta troca de listas leva à construção, em cada nó *v*, de um conjunto *Av* (*Av* = conjunto de todos os nós descobertos pelos vizinhos de *v* mas que não foram descobertos pelo próprio nó *v*). Assim que as trocas de listas são concluídas, o nó *v* começa a contatar os nós em *Av* para ver se esses nós estão no seu alcance de transmissão (isto é, vizinhos seus ainda não descobertos). Para isso, o nó *v* alterna, por um período pré-definido de tempo, entre os modos *page* e *page scan*, tentando descobrir os nós em *Av*. Mais especificamente, enquanto no modo *page* o nó *v* tenta paginar seus nós *Av* um após o outro. Cada vez que dois nós *u* e *v* se descobrem mutuamente, eles se removem dos seus conjuntos *Au* e *Av* e trocam suas listas de vizinhos. Isto pode levar à novos nós para *u* e *v*

adicionarem aos seus conjuntos A_u e A_v , isto é, novos nós para eles paginarem. O comprimento desta fase deve ser cuidadosamente escolhido para que sejam descobertos todos os nós em A_v que estão de fato no alcance de transmissão de v . O segundo problema está relacionado à maneira que o protocolo *BlueTrees* foi definido. De acordo com [ZAR01], assim que um nó foi adquirido como escravo por um mestre, ele se torna um mestre ele mesmo, e começa a contatar todos os seus vizinhos (exceto o seu próprio mestre). Considere que um mestre v adquiriu os nós u e z como escravos, e considere que u e z são nós que se descobriram mutuamente durante a fase de descoberta. Então um *deadlock* pode ocorrer devido ao fato de que u começa a paginar z e vice-versa. Para resolver este problema foi associado um *time-out* ao ato de “Paginar” os dispositivos vizinhos. Se um vizinho descoberto não responde a um *page* dentro de um certo período de tempo, é considerado que ele deve estar no modo *page* e portanto já deve ser membro de alguma *piconet*.

2.3 *BlueNet*

O esquema de formação de *scatternet* proposto em [WAN02], *BlueNet*, produz uma *scatternet* cujas *piconets* tem um número limitado k de escravos. Após a fase de descoberta de dispositivo, cada nó aleatoriamente entra no modo *page* e *page scan* com uma probabilidade f_0 (fase 0).

Quando um nó é bem-sucedido em conseguir pelo menos um escravo, ele procede para a fase 1 e tenta adquirir até k vizinhos como escravos. Os nós que são adquiridos como escravos também procedem para a fase 1 junto com o mestre. Escravos que vão para a fase 1 ficam no modo *page scan*. Os mestres que vão para a fase 1 fazem a dupla tarefa de:

1. conseguir conhecer os mestres vizinhos e os mestres dos escravos vizinhos que estão afiliados em outras *piconets*, e
2. encontrar nós que se encontram na fase 0 e que estão no modo *page scan* e adquirí-los como escravos.

Para realizar estas tarefas, um mestre faz *page* para todos os seus vizinhos um a um e então decide se vai para o modo *page* ou *page scan* aleatoriamente com probabilidade f . Ao alternar desta forma entre *page* e *page scan*, todo mestre adquire até k dentre os seus vizinhos como escravos e consegue conhecer a informação necessária sobre os outros vizinhos.

Se um nó não é bem-sucedido em adquirir ao menos um escravo ou em ser convidado a se juntar à *piconet* de algum outro nó, ele fica executando a fase 0, decidindo novamente se fica em modo *page* e *page scan* com probabilidade f_0 , até que ele se torne consciente que todos os nós vizinhos são parte de alguma outra *piconet*. No caso de que um nó na fase 0 se mantenha isolado ele entra na fase 2, isto é, ele vai para o modo *page* e tenta interconectar à *piconets* vizinhas ao tentar adquirir como escravos um nó de cada uma destas *piconets* (até o limite k). Após completar esta tarefa, um nó na fase 2 sai do protocolo.

Um mestre na fase 1 que já contactou todos os seus vizinhos e adquiriu até k deles na sua *piconet* procede para a fase 3, a fase de interconexão de *piconets*. Nesta fase, os escravos das *piconets* formadas na fase 1, ao alternarem entre os modos *page* e *page scan*, tentam criar *links* com nós vizinhos de outras *piconets* da fase 1, como instruído pelos seus mestres. Os mestres selecionam um dos seus escravos para ir para o modo *page* e instrui todos os seus outros escravos para irem para o modo *page* e *page scan*. O escravo selecionado faz *page* nos seus vizinhos um a um, cada um por um período máximo chamado de T_{page} , para cada *page* bem-sucedido, o escravo pergunta ao mestre se o *link* correspondente deve ser estabelecido ou não. Quando todas as tentativas de *page* para os vizinhos foi feita, o mestre instrui o escravo a ir em modo *page scan* e seleciona outro escravo para ir para o modo *page*. Quando apenas um escravo restou com alguns nós vizinhos para contactar, este escravo é instruído a alternar entre os modos *page* e *page scan* para poder criar os *links* restantes. O processo todo termina quando todos os *links* necessários a *piconets* adjacentes forem estabelecidos. A conectividade da *scatternet* resultante não é garantida, isto é, nem todas as *BlueNets* são conectadas mesmo quando as topologias resultantes da fase de descoberta são conectadas. Um contra-exemplo simples é ilustrado pela Figura 2.2.

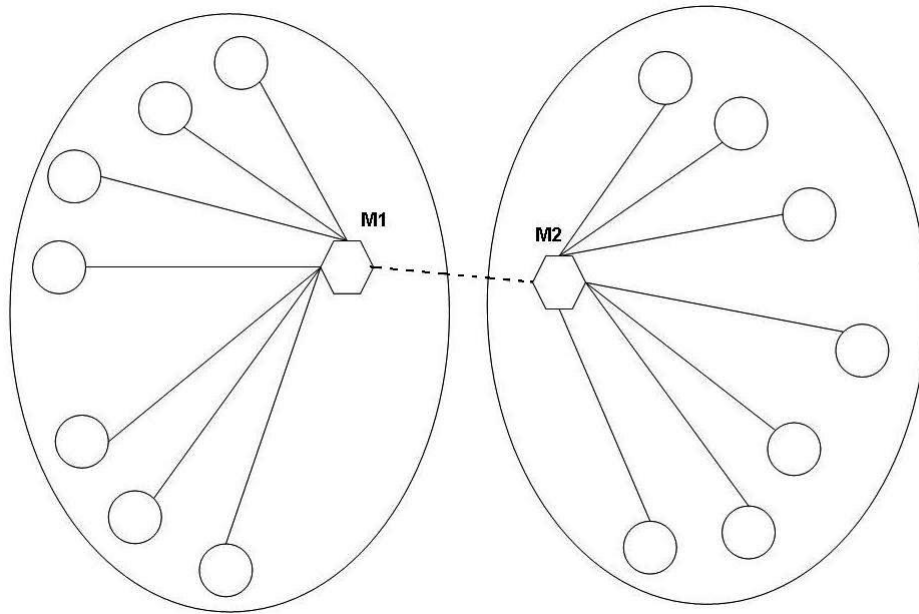


Figura 2.2: Exemplo de problema para formação do *BlueNets*

Assuma que os nós M1 e M2 descobriram um ao outro (linha pontilhada) e que eles são os únicos dois nodos que vão para o modo *page* na fase 0. Assuma também que eles convidam com sucesso para juntarem-se às *piconets* os 7 vizinhos mais à esquerda e os 7 vizinhos mais à direita, respectivamente. Neste caso, os nós em ambas as *piconets* vão para a fase 3. Porém, não há maneira para se interconectar as *piconets* seja por pontes intermediárias (nenhum escravo da *piconet* M1 é vizinho de nenhum dos escravos da *piconet* M2 e vice-versa) ou por interconectar diretamente M1 e M2, visto que eles já tem 7 escravos cada um.

De acordo com a descrição do protocolo [WAN02], escravos em uma *piconet* precisam ser bloqueados de estabelecerem conexões com nós que pertencem a sua própria *piconet*. Além disso, múltipla interconexão entre o mesmo par de *piconets* adjacentes tem que ser evitadas. O primeiro requisito é alcançado ao se ter o mestre informando cada um dos seus escravos sobre a identidade de qualquer novo escravo que se junta à *piconet*. Para evitar múltiplas conexões entre o mesmo par de *piconets*, cada vez que um escravo cria ou se junta uma *piconet* temporária com um escravo de uma *piconet* diferente, ele contata o seu mestre para saber se a conexão para esta outra *piconet* já existe. Se esse não é o caso, o mestre informa aos outros escravos sobre a nova conexão inter-*piconet*.

BlueNet é descrito pelo exemplo a seguir. A Figura 2.3 ilustra rede descoberta a partir da Fase de Descoberta. Neste caso, não há necessidade de executar a Fase de Reabastecimento. É assumido que o número associado a cada nó representa o seu ID único. Na fase inicial 0 (zero), os nós 6, 11, 17, 21 e 22 decidem aleatoriamente a começarem a fazer *page* para os seus vizinhos. Todos os outros nós vão para o modo *page scan*. Assim que o nó 6 é bem-sucedido em adquirir o nó 13 como escravo, ele se move para a fase 1, junto com o nó 13. O mesmo acontece com os nós 11 e 30, que vão para a fase 1 (nó 11 é o mestre e o nó 30 é o seu escravo). Similarmente, os nós 17 e 21 adquirem os nós 36 e 20 como escravos, respectivamente. Os quatro nós todos se movem para a fase 1. O nó 6 alarga a sua *piconet* ao adquirir os nós 7 e 25 como escravos, e o nó 8 se junta à *piconet* do nó 17. Quando o nó 22 é bem-sucedido em fazer *page* para os nós 7, 13 e 30, ele se dá conta que esses nós já se juntaram a alguma outra *piconet*. Sendo um mestre isolado, o nó 22 se move para a fase 2. Nó 15 está esperando por um *page* de algum nó vizinho. Visto que um *page* não chega nele, depois de um tempo ele decide aleatoriamente se fica em modo *page scan* ou se alterna para o modo *page*. Quando ele finalmente vai para o modo *page* e consegue contatar com sucesso os seus vizinhos 20 e 36, e vendo que eles já são escravos de outros mestres, ele se move para a fase 2. A Figura 2.4 mostra as *piconets* originais [LI04], isto é, as *piconets* formadas pelos nós nas fases 1 e 2.

Os nós 15 e 22, que estão na fase 2, fazem *page* a todos os seus vizinhos para que conectem em *piconets* adjacentes. O nó 22, por exemplo, faz *page* aos vizinhos 7, 13 e 30, até que ele consiga contatar todos eles. Ao comunicar com o nó 30, o nó 22 pede para construir um *link* entre a *piconet* 22 e a *piconet* 11. O nó 30 repassa essa requisição para o seu mestre, o nó 11, que concorda em compartilhar o nó 30 como um escravo-ponte. Similarmente, 22 faz o mesmo pedido ao nó 7, e o nó 6 concorda em compartilhar o nó 7 como escravo-ponte. Quando o nó 22 faz o mesmo com o seu nó vizinho 13, o 13 pede ao seu mestre nó 6 pela permissão para se juntar à *piconet* 22, mas nesse caso, o nó 6 não concorda em compartilhar o nó 13 visto que a *piconet* 6 já está interconectada à *piconet* 22, e a especificação do protocolo exige que não existam *links* múltiplos entre *piconets* adjacentes. Similarmente, o nó 15 alarga a sua *piconet* para incluir os nós 20 e 36 como escravos, e portanto estabelecendo *links* para as *piconets* 21 e 17, respectivamente. Nós 15 e 22 saem da execução do protocolo neste momento.

Quando os nós já contataram todos os seus vizinhos, os mestres na fase 1 passam para a fase 3, junto com os seus escravos. Nesta fase, *piconets* adjacentes são interconectadas entre pontes inter-

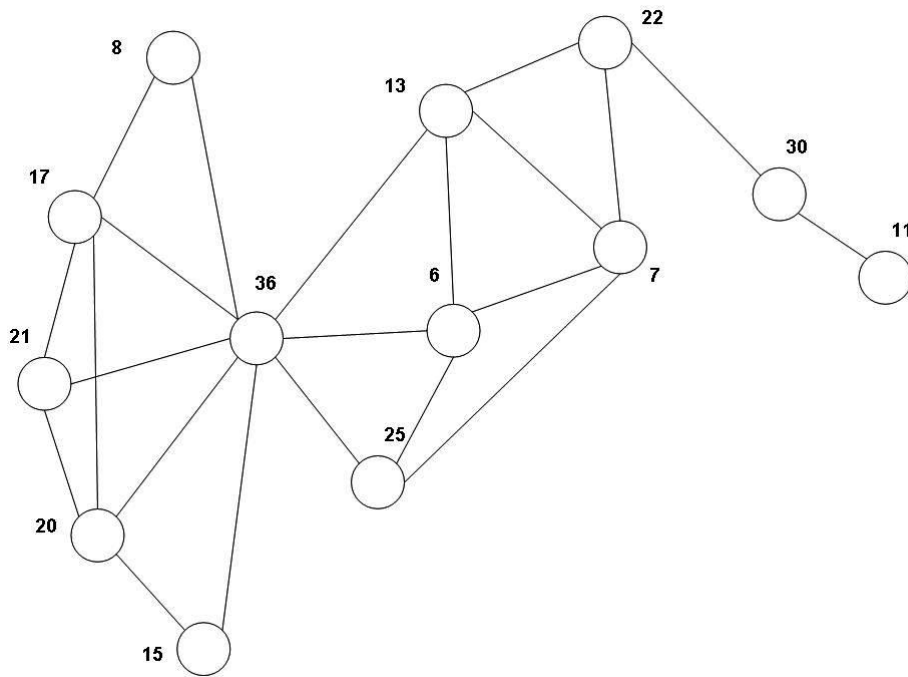


Figura 2.3: Rede descoberta a partir da fase de descoberta do *BlueNet*

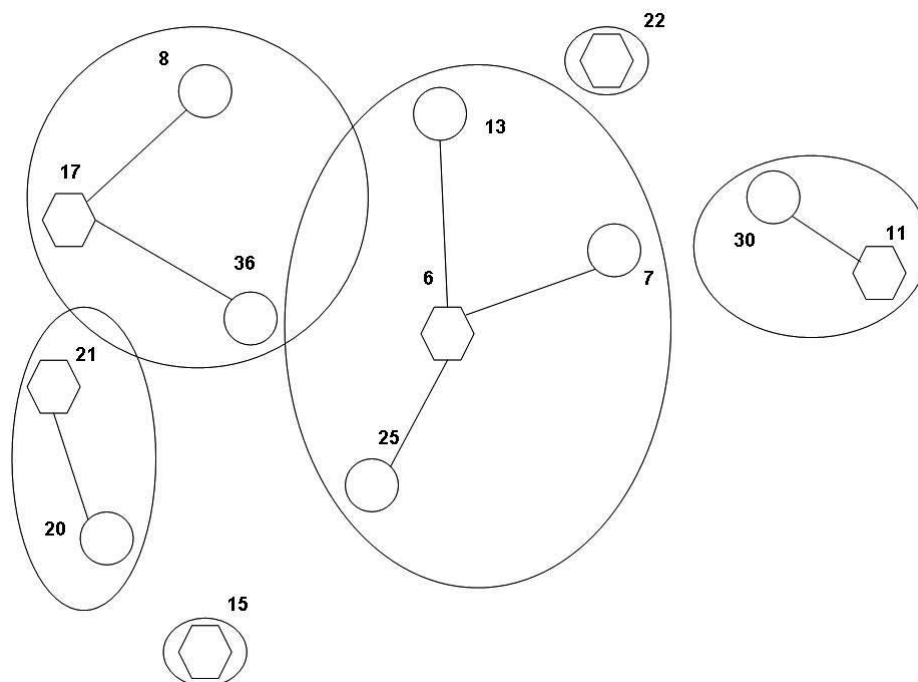


Figura 2.4: *Piconets* formadas nas fases 1 e 2 do *BlueNets*

mediárias. Nó 21 instrui o seu único escravo, o nó 20, a alternar entre os modos *page* e *page scan* para conseguir conectar todos os escravos de *piconets* adjacentes. Os nós 30 e 20 vão eventualmente estabelecer um *link* para interconectar as *piconets* 17 e 21, respectivamente. No caso ilustrado na

Figura 2.5, nó 20 se torna o mestre de uma *piconet* extra, incluindo o nó 36 como escravo. A *piconet* 25, que também inclui o nó 36 como escravo, é similarmente formada para interconectar as *piconets* 6 e 17. Nó 13 recebe notícia de que o nó 36 foi ligado à *piconet* 6 através do nó 25. Portanto, ele não contata o nó 36. A *BlueNet* resultante é ilustrada pela Figura 2.5, onde as *piconets* extras são ilustradas com linha pontilhada.

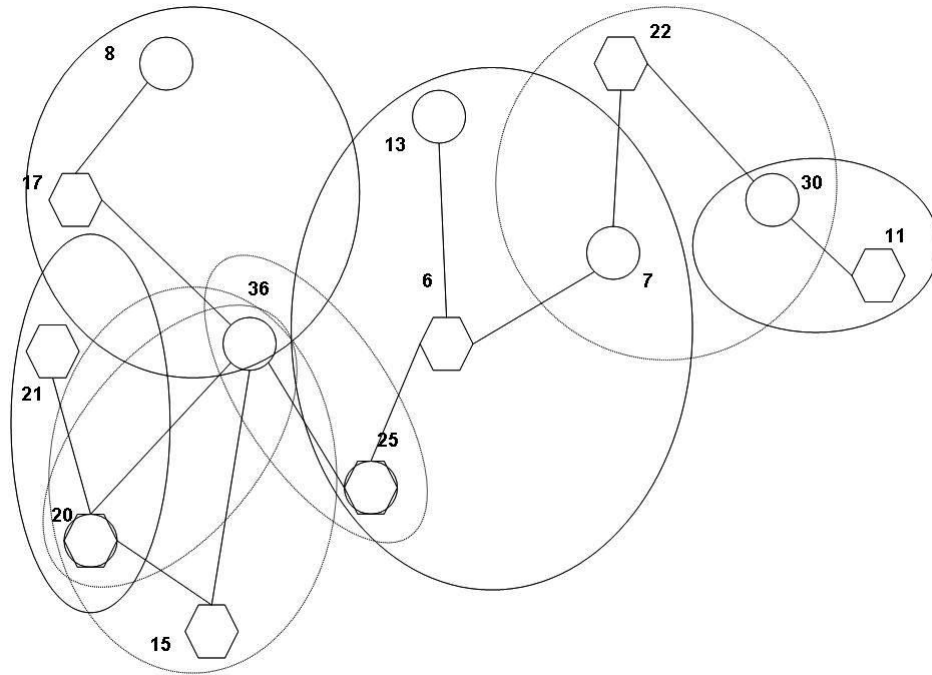


Figura 2.5: *Scatternet* formada pelo protocolo *BlueNets*

2.4 *BlueStars*

O protocolo apresentado em [PET03], *BlueStars*, procede, a partir da fase de descoberta de dispositivos, com as duas fases seguintes chamadas de Fase de Formação de *Piconets* e de Fase de Formação de uma *Scatternet* Conectada. Baseado em um peso localmente e dinamicamente computado, um número que expressa o quão adequado um nó é para se tornar o mestre, e no conhecimento do peso dos seus nós vizinhos obtido durante a fase de descoberta, cada nó decide se ele vai se tornar um mestre ou um escravo. Esta decisão é tomada em um nó dependendo da decisão dos seus vizinhos maiores, com peso maior que o seu, e então ele repassa essa decisão para os seus vizinhos menores. Em especial, um nó que decide ser um mestre ou é um nó *init* ou é um nó cujos todos vizinhos maiores decidiram serem escravos.

Um nó ao qual foi dito via *page* por um ou mais dos seus vizinhos maiores que eles são mestres este nó se torna escravo do primeiro mestre que deu *page* nele. Esta fase do protocolo leva à partição da topologia, resultante da fase de descoberta, em *piconets* que possuem um mestre e um número de

escravos que não é necessariamente limitado só pelo número de vizinhos do mestre. Observe que existem pares de mestres vizinhos.

Após esta fase, cada mestre procede para a seleção de dispositivos-ponte para interconectarem múltiplas *piconets* de forma que a *scatternet* resultante seja conectada. Para alcançar conectividade é necessário e suficiente que cada mestre estabeleça um caminho com (isto é, escolhe pontes para) todos os mestres que estão a dois ou três hops de distância [CHL99]. Mais precisamente, mestres estão a dois pulos de distância se eles podem ser interconectados por escravos-ponte (esta é a melhor maneira de interconectar *piconets* adjacentes). Mestres estão a três pulos de distância se eles não estão a dois pulos de distância e não tem um par de pontes intermediárias pelos quais eles possam se juntar. O conhecimento sobre quais nós são os mestres que estão a dois ou a três pulos de distância é alcançado durante a Fase de Formação de *Piconets*. Especificamente, cada nó *v* comunica o seu papel (e possivelmente a identidade e o peso do seu mestre) para todos os seus vizinhos menores e também para todos os seus vizinhos maiores que sejam escravos. Se um nó é um escravo, ele espera pelos vizinhos menores para comunicar a mesma informação. Desta forma, ao final da Fase de Formação de *Piconets* cada nó é ciente dos IDs de todos os seus vizinhos e do ID e do peso dos seus mestres, que é a informação necessária para a Fase de Formação de uma *Scatternet* Conectada.

2.5 Protocolo LSBS

O principal objetivo do protocolo Li-Stojmenovic /*BlueStars* (*LSBS*), que é uma combinação da construção Yao proposta em [LI04] e o *BlueStars*, é construir uma *scatternet* conectada onde cada *piconet* não tem mais do que sete escravos.

O protocolo assume que cada nó conhece a sua própria identidade, um peso computado dinamicamente que indica o quanto este nó é adequado para servir como mestre (como no *BlueStars*), e sua própria localização no plano (geralmente provida por um dispositivo GPS on-board, ou por qualquer sistema de posicionamento inercial que seja adequado). É assumido que, como resultado da fase de descoberta de dispositivos, um nó também sabe a identidade dos seus vizinhos, os seus pesos, e as suas localizações. Os dispositivos descobertos também trocam informação sobre os seus vizinhos (alcançando um conhecimento de vizinhança de dois pulos).

Na descrição do algoritmo dado em [LI04] é assumido que nós são espalhados no plano e que o grafo de rede resultante da fase de descoberta de dispositivos é conectada.

Dado que a fase de descoberta de dispositivos do *BlueStars* não produz uma rede conectada foi realizado nos experimentos de simulação uma fase extra descrita para o protocolo *BlueTrees* para que o protocolo funcione corretamente (reabastecimento). Observe-se que neste caso, já que o nó “*v*” está consciente da informação de localização dos seus vizinhos de dois pulos, apenas os nós que, de forma esperada, estão no alcance de transmissão de “*v*” precisam ser incluídos em “*Av*”. Isto reduz de forma consistente o número de nós que precisam ser paginados por um nó que está executando a Fase de reabastecimento do *BlueTrees*. O conhecimento da localização é então explorado para aplicar sobre

a topologia de reabastecimento algumas técnicas baseadas em geometria para reduzir o grau da rede para até “ k ” ≤ 7 . A construção Yao é executada em cada nó “ v ” e procede da seguinte maneira. O nó v divide o plano em torno dele em “ k ” ângulos iguais. Em cada ângulo, o nó “ v ” escolhe o vizinho mais próximo “ u ”. Na existência de empates os mesmos são quebrados de forma arbitrária.

Um *link* entre nós v e u sobrevive à fase de construção Yao se e somente se v escolheu u e vice-versa. Todos os outros *links* são deletados. Para tomar tal decisão os nós precisam trocar com os seus vizinhos a informação relativa aos nós que eles selecionaram. Apesar de alguns *links* serem deletados, a construção Yao garante a conectividade da topologia resultante [LI04] [YAO82].

Assim que uma topologia conectada com tal grau limitado foi obtida, o algoritmo *BlueStars* para formação de *scatternets* descrito anteriormente usa os pesos dos nós para selecionar os mestres, os escravos, e as pontes necessárias para formar uma *scatternet* conectada com limitação de grau

Considerando novamente a rede da Figura 2 como sendo a rede resultante da fase de descoberta de dispositivos e da fase de reabastecimento. O único nó com mais de 7 vizinhos é o nó 36. Portanto, ele executa o procedimento de construção Yao para descartar um dos seus 8 vizinhos. Assumindo que os nós 20 e 21 caem no mesmo de um dos 7 ângulos em que o nó 36 particionou o plano em torno dele, o resultado da fase de construção Yao é o cancelamento do *link* entre o nó 21 e o nó 36. A seguinte execução do *BlueStars* sobre a topologia Yao leva à mesma *scatternet* que nós iríamos obter com o *BlueStars* exceto pelo fato de que agora os nós 36 e 21 não são mais vizinhos. Portanto, sendo o nó 21 o maior na sua vizinhança, ele se torna um mestre e adquire o nó 20 como o seu escravo. Na fase final de formação de *scatternets* conectadas, o nó 20 é escolhido como escravo-ponte entre o mestre 21 e o mestre 36.

2.6 Considerações Finais sobre o Estado da Arte

Após o estudo dos trabalhos sobre *BlueTrees* [ZAR01], *BlueStars* [PET03], *BlueNet* [WAN02] e *LSBS Protocol*[LI04] e [BAS04], alguns aspectos importantes foram definidos como as propriedades desejáveis que os protocolos de formação de *scatternets* devem satisfazer. Estas propriedades são demonstradas na Tabela 2.1, onde cada item está especificado abaixo:

- Formação de *scatternets* conectadas, garante a conexão de todos os nós que estão dentro da área de formação da rede;
- Habilidade de funcionar em redes desconectadas, prevê meios de manter a rede em caso de desconexão de algum nó;
- Formação de *scatternets* com múltiplas rotas, se a rede formada possui mais de uma rota disponível para comunicação entre dois nós;
- Formação de *scatternets* com um número limitado de escravos por *piconet*, se as *piconets* formadas possuem um número limitado de escravos.

- Seleção de mestre baseada em recursos, se o protocolo usa características de cada nó, como por exemplo, poder computacional para a escolha do mestre.

Tabela 2.1: Propriedades de formação de scatternets

Propriedades	<i>BlueTrees</i>	<i>BlueNet</i>	<i>BlueStars</i>	<i>LSBS Protocol</i>
Formação de scatternets conectadas	×		×	×
Habilidade de funcionar em redes desconectadas		×	×	×
Formação de scatternets com múltiplas rotas		×	×	×
Formação de scatternets com um número limitado de escravos por piconet	×	×		×
Seleção de mestre baseada em recursos			×	×

Outro aspecto importante é que todos os protocolos dependem de uma fase de descoberta de dispositivo que precede a fase de conexão e de interconexão de *piconets* e nenhum dos protocolos tratam a saída (queda) e entrada de dispositivos na rede, fato que motivou a modelagem de um protocolo para aplicações como jogos onde o tempo de reconstrução total da rede impossibilita a utilização dos protocolos estudados.

Foi definido após o estudo das referências bibliográficas que o modelo deveria trabalhar sobre o *BlueStars* [PET03], por ser o mais completo sem a necessidade de utilização de GPS como é o caso do e *LSBS Protocol*[LI04]. O próximo capítulo detalha o modelo de protocolo desenvolvido no presente trabalho.

Capítulo 3

APRESENTAÇÃO DO MODELO PROPOSTO

Neste capítulo será exposto o modelo proposto que se desenvolve sobre o *BlueStars* trabalho apresentado em [PET03], com o objetivo de introduzir melhorias na propriedade de Capacidade de Recuperação, pois esta pode ser interessante para certas aplicações interativas, como jogos por exemplo.

Na próxima seção são descritos os conceitos básicos da tecnologia *Bluetooth* necessários para o entendimento do funcionamento do protocolo. A seção 3.2 descreve um estudo de caso onde o modelo proposto pretende sanar deficiências do *BlueStars*, devido ao mesmo, possuir uma Capacidade de Recuperação para saída de nós da rede. Na seção 3.3 está descrito o modelo do protocolo desenvolvido neste trabalho.

3.1 Conceitos Básicos da Tecnologia *Bluetooth*

Nesta seção estão expostos os conceitos básicos de *Bluetooth* para compreensão do modelo de protocolo. Em [BLU04] é descrito detalhadamente a Tecnologia *Bluetooth*.

Todos os dispositivos em uma *piconet* compartilham o mesmo canal, ou seja, uma sequência de pulso de frequência que é derivada do ID único e do relógio *Bluetooth* do mestre.

A comunicação de e para um dispositivo é sempre realizada através do mestre da *piconet* à qual o dispositivo pertence. Em particular, um esquema *Time-Division Duplex* (TDD) é empregado para comunicações dentro de uma *piconet*. As transmissões ocorrem em pares de "slots" de 625 microssegundos, sendo que o primeiro slot de cada par é usado para comunicação mestre-escravo e o segundo para a comunicação do escravo que foi consultado em direção ao mestre.

A formação de *piconets* é feita em dois passos. Primeiro descoberta de dispositivos e depois troca de informação para criar um canal de comunicação entre um candidato à escravo e um candidato à

mestre. De acordo com a especificação da tecnologia *Bluetooth* o primeiro passo é feito por *inquiry* e *inquiry Scan*, e o segundo por *page* e *Page Scan*.

Para que a descoberta de dispositivos aconteça, dois dispositivos vizinhos precisam estar em modos opostos: ou seja um precisa estar operando em modo *inquiry*) e o outro tem que ser o dispositivo que está querendo ser descoberto (executando em modo *inquiry scan*).

O inquisidor transmite pacotes de “ID de inquisição” pedindo para que os dispositivos vizinhos se identifiquem para prover informação de sincronização que será necessária para o estabelecimento do *link* mais tarde. Para minimizar o tempo de descoberta de dispositivos, a especificação *Bluetooth* define que pacotes de ID devem ser bem pequenos e que eles precisam ser transmitidos sobre as frequências de uma seqüência de pulso de frequência de *inquiry* ou *inquiry scan*, que é pré-definida, trocando de frequência em uma taxa alta (duas vezes um *slot*). Um dispositivo em *Inquiry Scan* pula entre as diferentes frequências de forma bem lenta (uma frequência a cada 1.28 segundos), e portanto aumentando a probabilidade de um *handshake* na mesma frequência de um inquisidor. Assim que um pacote de ID é recebido no dispositivo em *Inquiry Scan*, o dispositivo calcula um intervalo de tempo de *backoff* e continua a escutar. Apenas quando um pacote de ID é recebido após a fase de *backoff* é que o dispositivo em *inquiry Scan* vai enviar um pacote FHS contendo a sua identificação e a informação de sincronização (o seu relógio *Bluetooth*).

O procedimento de inquisição descrito acima leva a um conhecimento assimétrico de dois dispositivos vizinhos: a identidade do inquisidor não é conhecida no dispositivo que recebeu um pacote de ID de inquisição. Por outro lado, após uma resposta bem-sucedida do dispositivo em *inquiry Scan*, o inquisidor sabe a identidade e o relógio do vizinho que acabou de responder. Isto permite que o inquisidor v estime a seqüência de pulso de frequência usada pelo seu vizinho e portanto convide ele a se juntar à sua *piconet* como um escravo.

Esse convite é feito por meio das operações de *paging*. Para que dois dispositivos vizinhos u e v estabeleçam um *link*, um precisa estar em modo *Page*, por exemplo, nodo v, e o outro em modo *Page Scan*, nodo u.

Por definição o dispositivo que encontrasse em modo *Page* é o mestre. O nó v transmite um pacote de ID de *page* nas frequências de u, contendo o endereço de u. Quando u, que está em *Page Scan*, recebe o tal pacote, ele imediatamente envia uma confirmação. Neste ponto, v transmite a u um pacote FHS que tem toda a informação necessária para que u se sincronize na seqüência de pulsos de frequência de v. Finalmente, os dois dispositivos trocam toda a informação para estabelecer um *link* e uma *piconet* é formada com v sendo o mestre e u o escravo.

Pode acontecer que esse dispositivo u, que está em *Page Scan*, já é mestre de outra *piconet* e que ele poderia tornar v um dos seus escravos. Neste caso, uma vez que uma *piconet* foi estabelecida entre v e u, com v como o mestre, o escravo u pode pedir uma troca de papéis. Esta situação é tratada explicitamente pela especificação do *Bluetooth* e é executada através da troca de um pacote específico do protocolo de gerência de *link* (LMP - *Link Manager Protocol*) que instrui os dois dispositivos para trocarem para a seqüência de pulso de frequência do novo mestre.

Para conservar a energia dos dispositivos *Bluetooth*, modos de operação de baixo consumo de bateria foram incluídas nas especificações que permitem que nós de *Bluetooth* entrem em modo *Sleep* quando não estão envolvidos ativamente em uma comunicação. Pode ser utilizado no modelo esta característica para deixar que um mestre libere um escravo para que este possa executar operações relativas ao protocolo em outra *piconet*, como é feito no modelo *BlueStars* [PET03].

Um escravo que é posto no modo *Park* por seu mestre não pode ativamente ser envolvido em uma comunicação com este mestre. Entretanto, os escravos estacionados acordam periodicamente dentro de intervalos de tempo predefinidos para escutarem mensagens do seu mestre. O des-estacionamento, possivelmente de múltiplos dispositivos, é alcançado através da transmissão de um LMP *unparkProtocol Data Unit* (PDU) nessa janela de tempo. Este pacote carrega o ID dos dispositivos que devem ser des-estacionados e qual será os seus novos endereços de escravo ativos. Escravos estacionados podem causar o disparo de um pacote LMP PDU através do envio explícito de requisições durante janelas de tempo pré-allocadas.

Similarmente, dispositivos ativos podem pedir para serem estacionados (ou podem ser estacionados pelo seu mestre) através da troca de um pacote LMP *park* com o seu mestre.

3.2 Estudo de Caso

Após o estudo dos protocolos de formação de *scatternet* existentes, uma das desvantagens visualizadas, é que nenhum dos protocolos trata a saída ou entrada de dispositivos, na rede após esta última ser iniciada. Jogos digitais, são um modelo de aplicação, que deve atender esse aspecto, pois a cada saída de um dispositivo no decorrer de uma partida causaria uma nova fase de descoberta de dispositivos que, como já foi citado anteriormente, é a etapa que mais consome tempo na formação de uma *scatternet*.

O protocolo proposto visa atender as características descritas nessa seção. Inicialmente, o local onde jogos acontecem tem um espaço limitado e pré-determinado, como por exemplo uma sala de espera ou um vagão de trem. Com isso, os nós tendem a ficar concentrados formando assim um grafo de visibilidade quase completo, isto é, onde a maioria dos nós tem a possibilidade de estabelecer comunicação com todos os outros. Neste cenário, o protocolo *BlueStars* acarretaria um consumo de bateria elevado devido a utilização dos modos *Park* e *UnPark* [PET03], que seria necessário pois no *BlueStars* não há limite de conexões em uma *Piconet*.

No cenário de uso visualizado, os usuários da aplicação chegariam ao local e formariam uma rede de dispositivos portáteis tais como *notebooks*, telefones celulares e PDA's com a finalidade de, por exemplo, jogar um jogo. Imagina-se inicialmente, o uso do modelo para jogos "casuais", onde o tempo do uso da aplicação, isto é, da "partida" do jogo, ficasse, por exemplo, entre 20 e 30 minutos, mas podendo ser mais longo. Isso modifica um parâmetro muito importante, que é o algoritmo de escolha de pontes da *scatternet*, pois em todos os protocolos existe a preocupação com o consumo de energia dos nós-ponte, que é aproximadamente 20% maior que nos outros nós [BAS04].

Como citado anteriormente, no caso de desconexão da rede de um nó, seja por saída ou queda, os protocolos estudados teriam que executar novamente a descoberta de dispositivos, que levam no caso do *BlueStars*, por exemplo, por volta de 28s [BAS04].

Foi baseado nessas características que foi modelado o protocolo proposto neste trabalho.

3.3 Modelo Proposto

Nesta seção está a definição do modelo proposto que assim como o *BlueStars* [PET03] é dividido em três etapas: a etapa de descoberta de topologia, a etapa de formação de *Piconets* e a etapa de formação de *Scatternet*.

3.3.1 Fase de Descoberta de Dispositivos

Nesta etapa é descrito como cada par de dispositivos vizinhos obtém as informações sobre o ID um do outro usando os procedimentos de *Inquiry* e *Page*. Os procedimentos de *Inquiry* descritos nas especificações do *Bluetooth* indicam como um dispositivo no modo *Inquiry* pode fazer com que um dispositivo vizinho que está no modo *Inquiry scan* envie o seu ID e a sua informação de sincronização necessária para o estabelecimento de um *link*. Entretanto, nenhuma indicação é dada em como garantir que os dispositivos vizinhos estão em modos opostos de *inquiry*, que é a condição necessária para que eles se comuniquem. Além disso, o *broadcast* da mensagem de *inquiry* pela fonte não contem nenhuma informação sobre ela mesma. Assim, uma vez que dois dispositivos vizinhos terminam um *handshake* de *Inquiry*, apenas a fonte sabe a identidade do dispositivo que está em *Inquiry Scan*.

Para superar essas desvantagens e alcançar o conhecimento mútuo de IDs, foi usado assim como no *BlueStars Scatternet*, um mecanismo similar ao introduzido em [2]. Cada dispositivo alterna entre os modos *Inquiry* e *Inquiry Scan*, ficando em cada modo por um intervalo de tempo escolhido aleatoriamente.

Quando dois nós em modos de *Inquiry* opostos, ou seja *Inquiry* e *Inquiry Scan* fazem um *handshake*, eles criam uma *piconet* temporária que dura apenas o tempo necessário para que eles troquem os seus IDs.

Um dispositivo genérico v que executa o procedimento de descoberta, define um temporizador para que ele fique com o valor do intervalo tempo pré-definido pela fase de descoberta.

O dispositivo v então aleatoriamente entra ou no modo *Inquiry* ou no modo *Inquiry Scan*. Enquanto ele está em um dos modos, o dispositivo v realiza os procedimentos de inquirição descritos pelas especificações do *Bluetooth*.

Após uma fase de *Inquiry* (ou *Inquiry Scan*) ser completada o dispositivo troca para o outro modo (*Inquiry Scan* ou *Inquiry*). Como mencionado antes, para permitir que cada par de dispositivos vizinhos alcancem o conhecimento mútuo sobre os IDs uns dos outros.

O modelo requer ao dispositivo em modo *Inquiry* enviar um pacote FHS ou quando um dispositivo

em *Inquiry Scan* receba um pacote FHS, para que assim uma *piconet* temporária seja criada através de uma fase de *Page* e os dispositivos troquem seus IDs, juntamente com a informação de sincronização necessária para que mais comunicação ocorra. Assim que essa informação é trocada, a *Piconet* é dissolvida.

A eficácia do mecanismo descrito acima, para prover a informação mútua necessária para pares de dispositivos vizinhos, depende da idéia de que, ao alternar entre os modos *Inquiry* e *Inquiry Scan* e escolher aleatoriamente o comprimento de cada fase de *Inquiry* ou *Inquiry Scan*, tem-se uma alta probabilidade de que qualquer par de dispositivos vizinhos estejam em modos opostos por um tempo suficientemente longo, e portanto permitir que ambos os dispositivos descubram-se.

A Figura 3.1 mostra um cenário formado por 10 nós em um momento antes de rodar a primeira fase do modelo do protocolo, na Figura 3.2 mostra o mesmo cenário após a fase de descoberta de dispositivos, uma linha entre dois dispositivos indica que eles estão no alcance de transmissão um do outro.

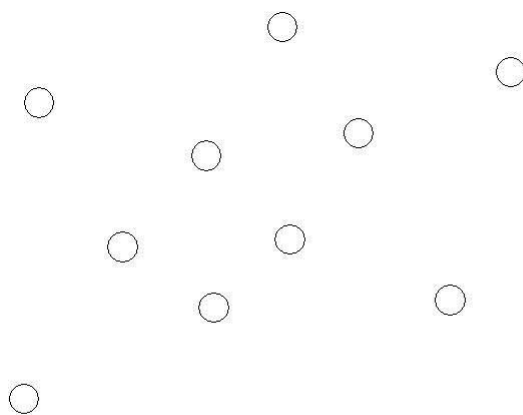


Figura 3.1: Cenário formado por 10 nós antes de qualquer formação de *Scatternet*

3.3.2 Fase de Formação de *Piconets*

Baseado nas informações recolhida na fase anterior, ou seja, o ID, e a informação da sincronização dos vizinhos descobertos, cada dispositivo executa um algoritmo localmente. A regra seguida por cada dispositivo é a seguinte: Um dispositivo v decide se será um mestre ou um escravo dependendo da decisão feita pelo vizinhos com peso maior. O peso pode ser definido de acordo com vários critérios, como por exemplo a capacidade de processamento e a quantidade de energia da bateria disponível. No modelo e testes adotou-se apenas pelo ID. Em especial, v se torna o escravo do primeiro mestre entre seus vizinhos maiores que enviou um *Page* para ele e o convidou para se juntar à sua *piconet*. Caso nenhum vizinho maior convide v , o próprio v transforma-se em um mestre.

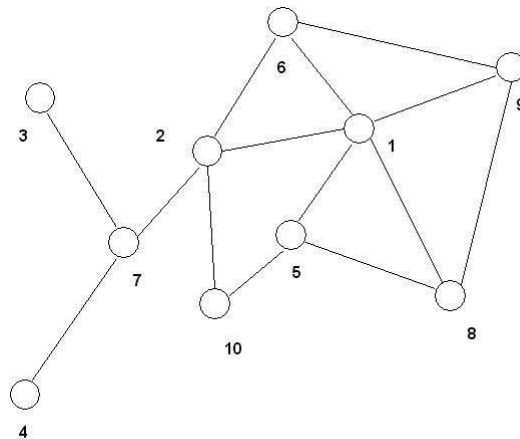


Figura 3.2: Cenário da Figura 3.1 após a fase de descoberta de dispositivos

Uma vez que um dispositivo decidiu seu papel, comunica a todos seus vizinhos (menores) de modo que possam também fazer sua própria decisão. Vamos chamar os nós que têm o peso maior dentre os seus dispositivos nó inicial da vizinhança. Dado o ordenamento total correspondente, há sempre pelo menos um nó inicial. Os nós iniciais são os dispositivos que iniciam a fase da formação de *Piconets*. Eles serão mestres. Estes são os únicos dispositivos que vão para o modo *Page* imediatamente depois da fase da descoberta de dispositivos. Todos os outros dispositivos vão para o modo *Page Scan*. Os dispositivos escolhidos como nó inicial são ilustrados como triângulos na Figura 3.3.

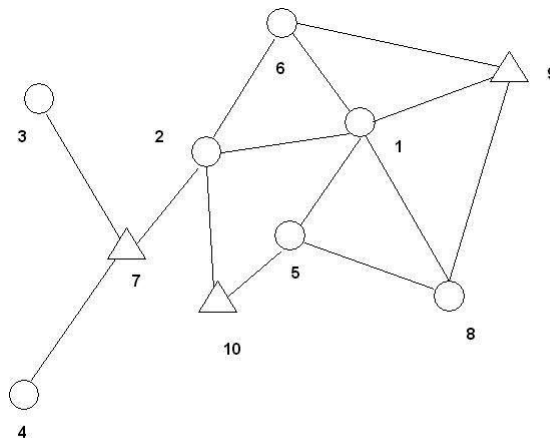


Figura 3.3: Nós iniciais escolhidos para o cenário da Figura 3.2

O dispositivo genérico v fica no modo *Page Scan* até que ele possa receber um *Page* de todos os seus vizinhos maiores. Como mencionado anteriormente, nesta fase, se pelo menos um vizinho maior é um mestre, então v se juntará à *piconet* do primeiro mestre que fizer um *Page* nele. Caso contrário, o v se tornará um mestre ele próprio. Em qualquer um desses casos, assim que todos os vizinhos

maiores fizeram *Page* nele, o *v* troca para o modo *Page* e faz um *Page* em um por um desses vizinhos que ainda não estão cientes do seu papel.

Apenas os dispositivos selecionados como nó inicial irão para o modo *Page* e vão começar a paginar os seus vizinhos menores um por um. Os parâmetros do *Page* são a identidade do dispositivo paginado *u*, a identidade do dispositivo paginador *v*, o papel do Paginador (se é mestre ou escravo), e, no caso em que o dispositivo paginador *v* é um escravo, a identidade do dispositivo que é o mestre do *v*. No caso de *v* ser um mestre, esse parâmetro é irrelevante e pode ser setado automaticamente para apontar para o próprio *v*. Todos os dispositivos não selecionados para serem nós iniciais vão para o modo *Page Scan*.

O procedimento de gravar o papel de um dispositivo *u* inclui toda a informação da sincronização e de endereçamento, que permitem a *v* estabelecer uma comunicação com *u* mais tarde, se necessário. Além disso, se o nó *u* for um escravo, a identidade do mestre do *u* é também gravada.

Ao receber um *Page* de um dispositivo *u*, o dispositivo *v* começa verificando se este foi um *Page* de um vizinho maior ou de um menor. Um *Page* de algum vizinho menor é necessário para recolher a informação usada mais tarde no procedimento de seleção de pontes. No caso em que o *Page* é de um vizinho maior *u*, *v* verifica se *u* é mestre. Se é mestre, e *v* não é parte de nenhuma *piconet* ainda, então o *v* se junta à *piconet* do *u*. Se, ao contrário, o dispositivo *v* já está em uma *piconet*, então o *v* informa ao dispositivo *u* sobre isso, também comunicando o ID de seu mestre. O dispositivo *v* prossegue então para verificar se todos seus vizinhos maiores já fizeram *Page*. Se este não for o caso, ele passa então a esperar outro *Page*, portanto, parando a execução do procedimento.

Quando paginado com sucesso por todos seus vizinhos maiores, o dispositivo *v* sabe se ele já se juntou à *piconet* de um mestre maior ou não. No primeiro caso, o dispositivo *v* é o escravo do mestre maior que o paginou primeiro. No último caso, o dispositivo *v* próprio está indo ser um mestre. Em todo o caso, o dispositivo *v* vai para o modo *page* e comunica sua decisão primeiramente a todos seus vizinhos menores e então também a seus vizinhos maiores que são escravos.

Neste momento, um mestre *v* se retira da execução desta fase do protocolo.

Se o dispositivo *v* for um escravo, ele retorna ao modo *Page Scan* e espera os *Pages* de todos seus vizinhos menores. Certamente, alguns dos vizinhos menores de um escravo podem não ter decidido ainda o seu papel no momento em que recebem um *Page* de um vizinho maior. Assim que um dispositivo fizer uma decisão sobre o seu papel, conseqüentemente ele faz *Page* para os seus vizinhos maiores e comunica se é um mestre ou um escravo, junto com o ID do seu mestre (se ele decidiu ser escravo). Esta troca de informação é necessária para executar a fase seguinte de seleção de pontes para que se obtenha uma *scatternet* conectada.

O funcionamento da fase de formação da *Piconets* é ilustrado pelo exemplo a seguir:

Considere a rede *Bluetooth* ilustrada na Figura 3.2, onde uma conexão entre dois dispositivos indica que dois nós se descobriram mutuamente durante a fase de descoberta de topologia.

Dado que são os dispositivos com o ID maior em sua vizinhança, os dispositivos 10, 9 e 7 somente eles são nós iniciais (ilustrados como triângulos na Figura 3.3). Eles iniciam o protocolo no modo

Page e começam a fazer *Page* nos dispositivos vizinhos. Todos os outros nós começam no modo *Page Scan*. O dispositivo 7 faz *Page* com sucesso nos dispositivos 3 e 4, que se transformarão em escravos na piconet 7 resultante (foi seguido a nomenclatura típica de *bluetooth* para identificar uma *piconet* pelo seu mestre). A *Piconet 9* é formada por seu mestre, o dispositivo 9, e por todos seus vizinhos: dispositivos 6, 1, 8. O Mestre 10 faz um *Page* com sucesso nos dispositivos 2 e 5.

A Figura 3.4 mostra as *piconets* resultantes da Fase de formação de *Piconets*.

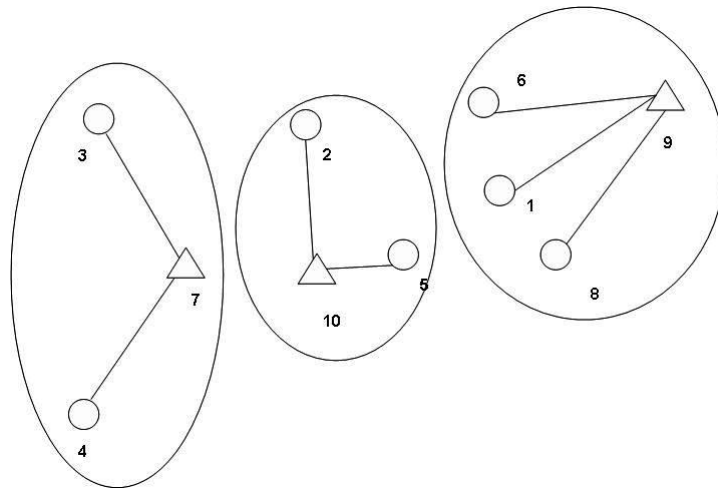


Figura 3.4: *Piconets* resultantes da fase de *Piconets*

Todos os dispositivos são sempre capazes de distinguir entre um *Page* enviado por um dispositivo na fase um, onde o *Page* é usado para estabelecer uma *piconet* temporária para atingir o conhecimento simétrico de topologia, e um *Page* da segunda fase que é baseado no papel do parâmetro. Na fase de descoberta de topologia, o papel do parâmetro é setado como "nenhum", enquanto na fase de formação das *BlueStars* ele é sempre setado como mestre ou escravo visto que os dispositivos enviam *Page* apenas após terem decidido os seus papéis.

As operações de protocolo desta fase dependem dos procedimentos padrões de *Page* do *Bluetooth*. Entretanto, os procedimentos *Page* e *Page Scan* usados no modelo assim como no *BlueStars* [PET03] supõem a possibilidade de trocas de informações adicionais, tal como o papel do dispositivo e, para os escravos, o ID dos seus mestres. Estas informações não podem ser incluídas no pacote de FHS que é o pacote trocado nos procedimentos padrões de *Page*.

A proposta feita no modelo *BlueStars* é adicionar um *Protocol Data Unit* (PDU) de LMP, incluindo campos onde possa ser inserido o papel do dispositivo que está enviando a mensagem e o ID do seu mestre. Isso proporciona uma troca mais fácil da informação necessária para a formação de *scatternets*.

Naturalmente, sempre que um escravo se junta a uma *piconet* não-temporária, uma formação completa de *piconet* precisa ser feita, e depois disso o escravo é posto no modo *Park* para permitir que ele prossiga com a operação do protocolo, por exemplo, executando o *Page* ele mesmo quando for necessário.

3.3.3 Fase de Formação de *Scatternets*

Nesta etapa estão as principais modificações propostas no modelo. Essas modificações tem como objetivo tornar o *BlueStars* seguro na presença de eventos de entrada e saída de nós da rede móvel.

A finalidade da terceira fase do protocolo é interconectar as *piconets* vizinhas selecionando dispositivos que farão a ponte entre as *piconets* de modo que a *scatternet* resultante, esteja conectada.

As tarefas principais realizadas por esta fase do protocolo são, portanto, a seleção de pontes para ligação das *Piconets*, e a interconexão dessas.

Dois mestres são chamados de mestres vizinhos se houver ou um caminho de dois pulos entre eles, com o nó intermediário sendo o escravo de um deles, também chamado de ponte escrava, ou há um caminho de três pulos que atravessa dois de seus escravos, chamados de pontes intermediárias. Por exemplo, na Figura 3.4, os mestres 9 e 10 são mestres vizinhos visto que eles podem ser interconectados através de várias possibilidades de pontes intermediárias: 5 e 8, 5 e 1, 2 e 6 e 2 e 1.

Um mestre é chamado de mestre inicial, se tiver o maior peso entre todos os seus mestres vizinhos. Conseqüentemente, o conjunto de mestres resultante da fase de formação de *Piconets* é dividido em dois conjuntos: os dispositivos mestres iniciais e os dispositivos que não são mestres iniciais. Referindo-se outra vez à Figura 3.4, o mestre inicial é o dispositivo mestre 10. Os 2 mestres restantes são mestres não iniciais. A conectividade da *scatternet* é garantida pelo seguinte teorema [PET03]:

Dado os *piconets* resultantes da fase da formação de *Piconets*, a ocorrência de uma *Scatternet* conectada é garantida se cada mestre estabelecer conexões multi-pulo a todos os seus mestre vizinhos.

Esse Teorema define um critério para selecionar pontes que garantam a conectividade da *scatternet* resultante. Para cada dois mestres, os seus escravos nos caminhos com dois pulos e com três pulos entre eles vão ser pontes. Se não há mais de um caminho de interconexão entre dois mestres (como entre nas *piconets* 10 e 9, na Figura 3.4), eles podem decidir ficar com apenas um escravo-ponte ou um par de pontes intermediárias entre eles ou então manter múltiplas interconexões. No protocolo *BlueStars* [PET03] fica estabelecido que apenas uma rota deve ser selecionada, enquanto no modelo proposto todas as rotas serão utilizadas.

A seleção de pontes é realizada localmente em cada mestre baseado nas informações coletadas até o fim da fase de formação de *Piconets*:

1. o ID dos vizinhos de um pulo de seus escravos;
2. o ID dos mestres dos vizinhos de um pulo dos seus escravos (coletados a partir dos seus próprios escravos) e
3. o ID dos mestres dos escravos vizinhos que não se juntaram à *piconet* dele.

Com essas informações o modelo seleciona qual será a rota primária, ou seja, por onde as mensagens irão trafegar até que a rota seja perdida. No caso de problemas com essa rota primária uma nova rota que já esta estabelecida mas não é utilizada passa a ser a rota de comunicação entre dois

mestres vizinhos. Para a seleção quando forem mestres vizinhos de dois pulos, o maior escravo-ponte é escolhido como a ponte que formará a rota primária. No caso de mestres vizinhos de três pulos, o par de pontes intermediárias selecionado é aquele no qual a soma dos ID deles é a maior. Por exemplo na Figura 3.4 a rota primária entre os mestres vizinhos 9 e 10 seria através das pontes intermediárias 5 e 8, no caso de problemas com essa rota, o modelo proposto passaria a utilizar a segunda maior soma que no caso seria 2 e 6.

Assim que a seleção de ponte é realizada, o protocolo é capaz de estabelecer todas as conexões e de estabelecer todas as novas *piconets*, para obter uma *scatternet* totalmente conectada.

Todo o mestre *v* inicia verificando se ele é um mestre ou não. Se ele é um mestre inicial, ele instrui cada um dos seus escravos ponte e pontes intermediárias *u* para irem para o modo *Page* e para fazerem *Page* quando é necessário, nos dois casos citados a seguir.

Nos mestres vizinhos para os quais *u* foi selecionado como o escravo ponte. Neste caso, assim que *u* se torna o mestre de um mestre vizinho *t*, eles realizam uma troca de papéis de forma que *u* também se torna um escravo na *piconet* de *t*. Neste caso, nenhuma nova *piconet* é formada e o escravo entre *v* e *t* é então um escravo nas duas *piconets*.

Às suas pontes intermediárias, que foram selecionadas para interconectar *v* com os seus mestre vizinhos *t* de três pulos. Neste caso, *u* se torna também um mestre de uma *piconet* cujo escravo é também um escravo de *t*, isto é, uma nova *piconet* é criada para fazer a ligação entre as *piconets* dos dois mestres.

O mestre inicial *v* pode então ele mesmo ir para o modo *page* para recrutar, para a sua *piconet*, alguns destes nós vizinhos que tenham se juntando à alguma outra *piconet*, de forma que estes nós podem ser pontes para os seus mestres originais.

Um nó *v* que não é um mestre inicial instrui todos os seus escravos *u*, que são pontes para mestre vizinhos maiores, para irem para o modo *Page Scan* e para esperarem que os nós especificados façam *page* neles. Então, *v* verifica para identificar se tem escravos ponte de mestre vizinho maiores aos quais ele tem que se interconectar. Se este é o caso, *v* espera para receber um *page* desses nós. Após as rotas para mestres vizinhos maiores serem desta forma estabelecidas, *v* começa a configurar conexões para mestres vizinhos menores. Para este propósito, ele age como se ele fosse um mestre inicial.

A Figura 3.5 mostra a *scatternet* resultante da conexão das *Piconets* da Figura 3.4 quando ao final do protocolo *BlueStars*. A Figura 3.6 mostra a *scatternet* resultante quando o modelo proposto é aplicado também sobre as *piconets* da Figura 3.4, onde as linhas tracejadas são as rotas auxiliares e as linha contínuas são as rotas principais.

3.3.4 Problema: Queda de Ponte

Como foi citado anteriormente o modelo visa resolver para um determinado grupo de aplicações (descrito na seção 3.1) o problema da saída ou queda de uma ponte da rede. No protocolo *BlueStars*

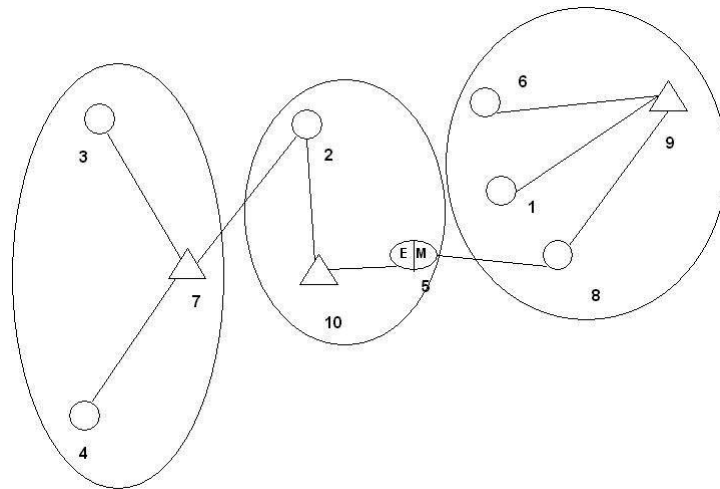


Figura 3.5: *Scatternet* resultante da terceira fase do *BlueStars*

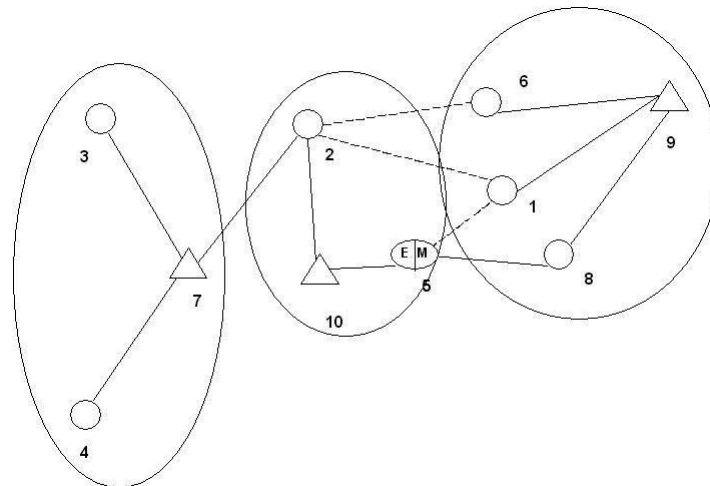


Figura 3.6: *Scatternet* resultante da terceira fase de formação *Scatternet*

o caso da queda ou saída de uma ponte é desconsiderado. O comportamento do protocolo nesse caso seria a desconexão das *piconets* da qual essa ponte pertencia. Por exemplo na *scatternet* da Figura 3.5, no caso do nó 8 desconectar-se o que resultaria seria uma *scatternet* formada pelas *piconets* 10 e 7 e a *piconet* 9 (formada pelo mestre 9 e pelos dispositivos 1 e 6) excluída da rede *Bluetooth* como mostra a Figura 3.7. Neste caso para que todos os nós restantes voltem a fazer parte da *scatternet* é necessário que o protocolo seja novamente executado desde o início, ou seja, desde a fase de descoberta de dispositivos. Esse caso, segundo [PET03], levaria uma perda de 28s, aproximadamente.

O modelo de protocolo proposto como mostrado na seção anterior leva em consideração todas as rotas possíveis entre duas *piconets*, sendo assim no caso da queda do nó 8 da *scatternet* da Figura 3.6, o procedimento do protocolo proposto seria o seguinte: como no estudo de caso que o modelo se propõe em resolver a troca de mensagem é constante, os nós que são mestres do nó 8 (no caso, os nós 9 e 5) percebem rapidamente a saída ou queda deste dispositivo. Então, o nó 5 avisa para o seu

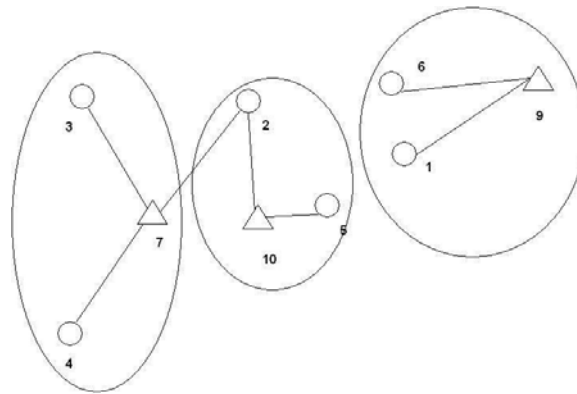


Figura 3.7: *Scatternet* e piconet isolada resultante da falha do nó 8

mestre inicial (que é o nó 10) a desconexão do dispositivo que o ligava com o outro nó inicial 9 (isto é, o nó 8).

A partir deste ponto, os nós 10 e 9 realizam todas as trocas de mensagens a partir da segunda rota de sua lista, ou seja, a fase de formação de *scatternets* e voltam a se conectar através da rota auxiliar formada pela ponte intermediária formada pelos nós 2 e 6. O nó 2 então se torna o mestre da piconet formada pelos nós 2 e 6. A Figura 3.8 demonstra como seria a rede da Figura 3.4 após a queda do nó 8 e se a *scatternet* estivesse executando o protocolo proposto ao invés do *BlueStars*. Para essa troca de rota o tempo estimado fica em torno de 3s, com base no tempo que é definido para a terceira fase do protocolo *BlueStars*.

Portanto, o protocolo proposto é mais tolerante à queda ou saída de nós ponte da *scatternet*, isto é, no protocolo proposto, a queda ou saída de um nó ponte acarreta um atraso de até 3s na comunicação das *piconets* unidas pelo nó ponte ou por uma ponte intermediária. Já no caso do *BlueStars*, o atraso pode ser de até 28s e envolve uma reconstrução total da *scatternet*, ou seja, um atraso significativamente maior.

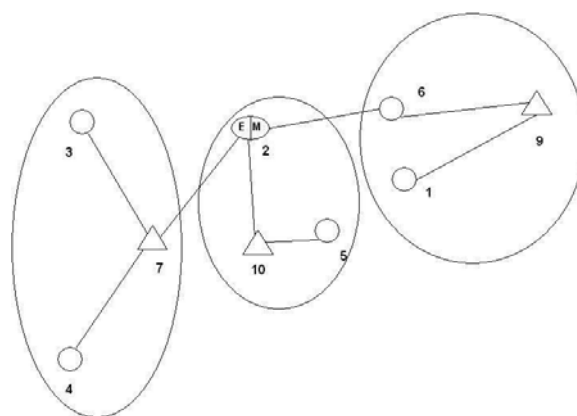


Figura 3.8: *Scatternet* resultante do protocolo proposto após a falha do nó 8

No próximo capítulo são encontrados os resultados das simulações feitas e as comparações com o protocolo *BlueStars*.

Capítulo 4

ESTRATÉGIAS PARA IMPLEMENTAÇÃO DO MODELO

4.1 Projeto de SoC para Formação de *Scatternets*

O presente trabalho visava se inserir no contexto do projeto *Brazil-IP*, que envolve várias universidades brasileiras no desenvolvimento de uma plataforma de prototipação de sistemas *wireless* denominada *Fênix*. Dentro do projeto, cada universidade é responsável pelo projeto e implementação de um IP específico. O conjunto de IPs é testado e validado na Plataforma *Fênix*. A Figura 4.1 apresenta a organização interna da *Fênix*, salientando os dez núcleos (*IP cores*) componentes desta, bem como a interconexão destes via a rede de interconexão Hermes. A partir da Figura 4.1, nota-se que os blocos constituintes da *Fênix* podem ser divididos em nove núcleos processadores e um núcleo de comunicação (Hermes), todos interconectados através do núcleo Hermes.

A plataforma *Fênix* é uma boa opção validação do trabalho. Essa plataforma que é composta do microprocessador *RISC Leon (SPARC compliant)*, módulos de decodificação áudio/vídeo MP3/MPEG4, módulos de comunicação tais como USB e *Bluetooth*, todos estes interligados através da rede *Hermes*. A rede intrachip Hermes é a contribuição da PUC-RS ao projeto, sendo desenvolvida pelo grupo GAPH da FACIN. Esta plataforma encontra-se prototipada em FPGAs Xilinx e Altera, e vem sendo a base de diversos trabalhos de grupos de pesquisa da FACIN, e das demais universidades parceiras.

Conforme ilustrado na Figura 4.1, a *Fênix* contém um controlador *Bluetooth*. Esse controlador é um núcleo IP Desenvolvido pelos parceiros da USP. De forma a possibilitar a formação de uma *scatternet*, protocolo proposto, é necessário que o núcleo *Bluetooth* possua capacidade para controle de dois rádios. Assim, um nó poderá, simultaneamente, ser mestre em uma *piconet*, e escravo em outra. Essa alteração no núcleo *Bluetooth* da *Fênix* é responsabilidade dos desenvolvedores do mesmo, e assim que for realizada, a *Fênix* poderá vir a ser o primeiro dispositivo no mundo a possuir a capacidade

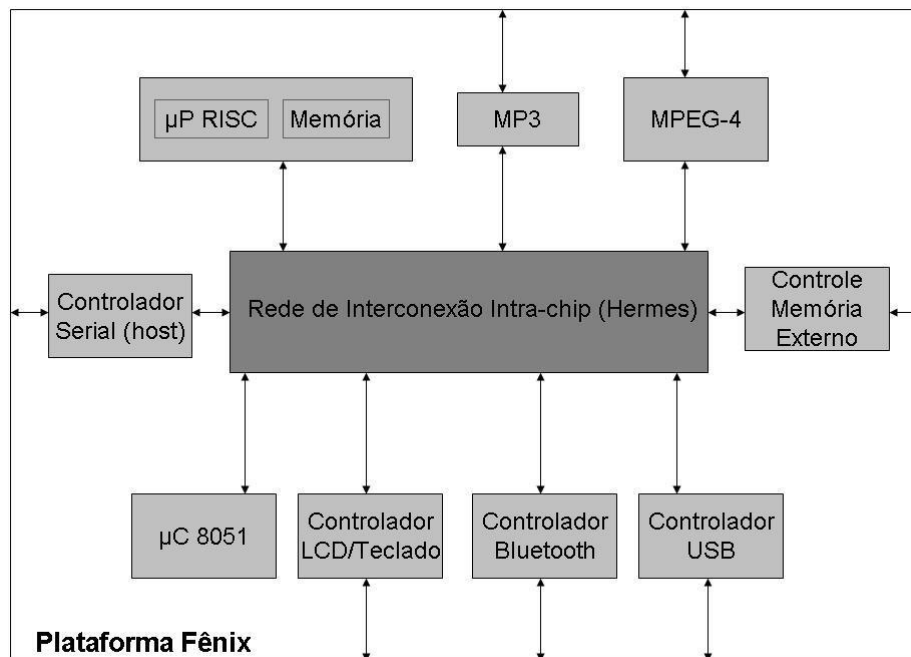


Figura 4.1: Diagrama de blocos da estrutura interna da Plataforma *Fênix*, apresentando cada um dos 10 núcleos constituintes da plataforma e sua interconexão. Conexões externas são apresentadas para os IPs pertinentes.

para suporte a formação de scatternets, conforme definido na especificação *Bluetooth* [BLU04].

O protocolo proposto nessa dissertação foi projetado e desenvolvido visando prover essa capacidade a *Fênix*. No momento da escrita dessa dissertação, o núcleo *Bluetooth* da *Fênix* ainda não possuía implementada a capacidade de controlar dois rádios simultaneamente, e a validação da proposta foi realizada por intermédio de simulação, conforme descrito a seguir.

4.2 Simulação do Modelo de Protocolo Proposto

Inicialmente, foi utilizado o simulador ns-2 [FAL06] para modelagem e validação do protocolo proposto. O ns-2 é um simulador de redes de computadores, com maleabilidade em relação aos protocolos de rede, suportando simulações de protocolos em larga escala e é usado também como um emulador de rede podendo fazer interação com uma rede real. O ns-2 possui um visualizador gráfico para animações da simulação (*Network Animation - NAM*), *timers* e escalonadores, prevê suporte aos protocolos TCP e UDP, contendo modelos para cálculos numéricos, ferramentas de plotagem (o *xgraph*) e vários geradores de tráfego. Possui módulos como o interpretador para tcl (*tcl/tk*) que é a interface com o usuário, o tcl com orientação ao objeto (*Otcl*), a implementação de classes para Tcl (*tlcl*), classes do simulador (ns-2), bibliotecas (*cweb* e *SGB*), geradores de topologia (*bt-itm*, *gt-itm*, *sgb2-ns*) e ferramenta para compreensão de arquivos. Além disso, O ns-2 traz consigo um código aberto onde é permitido a inclusão de novos módulos ou a alteração dos que já existem, além de ser

gratuito, sua utilização pode ser em diferentes plataformas como: Unix, linux, SunOS, FreeBSD e Solaris. Devido à sua flexibilidade, é tido como “padrão” para realização de testes e validação de protocolos de rede.

Porém, o ns-2 não possui capacidade para implementação de simulações de redes formadas com a tecnologia *Bluetooth*, devido a falta das especificações *Bluetooth* neste simulador, e as extensões disponíveis não possibilitavam as modificações necessárias como, por exemplo, as mensagens de *Page* que devem carregar o ID do mestre de um nó para seus vizinhos.

Devido a essas dificuldades associadas ao ns-2, foi desenvolvido um simulador, baseado na modelagem descrita no capítulo 3.

O simulador desenvolvido tem como objetivo obter resultados referentes ao tempo de indisponibilidade de uma *scatternet* (para comunicação normal da aplicação) devido à execução de operações de manutenção da mesma que são realizadas pelos protocolos de formação e de manutenção da rede *ad hoc*. Mais especificamente, o simulador compara o tempo de indisponibilidade médio gasto com a re-execução da fase de descoberta de dispositivos, da fase de formação de *piconets*, e da fase de formação de *scatternets*, tanto do protocolo *BlueStars* original, quanto do protocolo proposto por este trabalho, lembrando que o protocolo modificado, proposto, possui uma fase de formação de *scatternet* diferente da do *BlueStars* original.

O simulador é uma aplicação de linha de comando, implementada em Java, utilizando apenas as bibliotecas padrão. O simulador possui como entradas, os seguintes parâmetros:

1. Número de iterações: O simulador irá repetir várias vezes a simulação e, ao final, apresentará uma média do resultado obtido em todas as iterações
2. Número de nós: O simulador irá posicionar aleatoriamente, no ambiente simulado, a quantidade de nós especificada neste parâmetro
3. Tempo de simulação (em segundos): Cada iteração irá representar uma sessão de uso da aplicação (por exemplo, de um jogo) que durará o tempo especificado (o padrão são 1200 segundos, ou seja, 20 minutos)
4. Probabilidade de queda ou saída de nó (por segundo): A cada segundo de tempo de uso da rede que for simulado, o simulador irá utilizar este parâmetro para decidir, para cada nó, se ele irá falhar (cair ou sair do alcance da *scatternet*).
5. Largura do ambiente (plano 2D) em metros
6. Altura do ambiente (plano 2D) em metros
7. Algoritmo a ser utilizado (B: *BlueStars* original, ou M: *BlueStars* modificado). O simulador executa várias iterações sobre um único algoritmo e, portanto, para comparar resultados, é necessário executá-lo separadamente para cada algoritmo

8. Probabilidade de descoberta de nó em alcance: Este parâmetro, com valor padrão em 0.93, representa a chance de que dois nós, no alcance de transmissão um do outro, se descubram na fase de descoberta de dispositivos. Este parâmetro é retirado dos resultados obtidos em [BAS04]

Uma iteração do simulador possui as seguintes etapas:

1. *InitScenario()*: Posiciona, de forma aleatória (com gerador de números aleatórios distribuídos uniformemente no intervalo), a quantidade de nós especificada no parâmetro 2, em uma área 2D especificada pelos parâmetros 5 e 6.
2. *Discovery()*: Simula a fase inicial de descoberta de dispositivos, montando o grafo de visibilidade, de forma que alguns pares de nós, em alcance de transmissão uns dos outros, possam acabar não tornando-se visíveis uns aos outros (dependendo do parâmetro 8).
3. *ElectInits()*: Marca os nós que serão nós iniciais, segundo a terminologia do *BlueStars*
4. *FormPiconets()*: Utilizando os nós iniciais, forma as *piconets* iniciais, o que é especificado pela fase de formação de *piconets* do *BlueStars*
5. *FixIsolatedNodesAndFlagInitMasters()*: Uma etapa auxiliar, que precede a implementação da terceira fase (de formação de *scatternets*). Este método identifica nós que acabaram isolados devido a não terem recebido um *Page* de um mestre vizinho (nó maior), e não possuírem vizinhos menores que já não sejam escravos de outros mestres. Após identificados, estes nós viram mestres de *piconets* vazias, o que permite a sua inclusão na *scatternet* posteriormente através das pontes que serão criadas. Após, todos os mestres são marcados como "mestres iniciais", o que é essencial para que a fase de formação de *scatternets* funcione como especificado nos artigos do *BlueStars*.
6. (Alternativa 1) *ModifiedScatternet()*: Executa a terceira fase, ou seja, de formação de *scatternets*, que é proposta neste trabalho. Esta implementação, como discutido, irá alocar mais pontes entre duas *piconets* adjacentes (caminhos alternativos) para que a *scatternet* resultante seja mais tolerante à queda ou saída de nós.
7. (Alternativa 2) *BlueStarsScatternet()*: Executa a terceira fase, ou seja, de formação de *scatternets*, que é proposta pelo artigo do *BlueStars* [PET03].
8. *SimulateDrops()*: Este método finalmente irá executar uma iteração interna para cada segundo de tempo em que a rede é usada, utilizando o parâmetro 4 para determinar a probabilidade de um nó simulado cair. Quando isto ocorre, dependendo do algoritmo selecionado (*BlueStars* original ou modificado), um "tempo de indisponibilidade" diferente será acumulado para a iteração de fora do simulador.

Após todas estas iterações, é feita uma média do “tempo de indisponibilidade” obtido em cada iteração, e isto é apresentado como o principal dado de saída do simulador. Como visto anteriormente, o simulador permite a variação do tamanho do ambiente, do número de nós, da probabilidade de queda de um nó (por segundo) e do algoritmo utilizado, que foram os parâmetros variados durante os experimentos realizados. Os experimentos e os resultados obtidos são apresentados e analisados no Capítulo 5.

É importante ressaltar que o simulador faz uma simplificação que, porém, não deve alterar os resultados. Na fase de formação de *scatternets* do *BlueStars* original, escravos-ponte que unem dois mestres são escolhidos pelo maior peso. Isto é, se dois mestres possuem, ao final da fase de formação de *piconets*, dois nós, com pesos, por exemplo, 5 e 10, que estão no alcance de visibilidade de ambos (por exemplo, o nó 5 é escravo de um, e o nó 10 é escravo de outro nó, mas não necessariamente), então o nó 10 será o escolhido para ser a ponte entre estes dois mestres, pelo peso ser maior. Isto é feito, pois o *BlueStars* foi projetado para dar pesos maiores para nós com maiores capacidades. O simulador, por outro lado, assume que todos os nós são idênticos (rede homogênea), e a quantidade energia é suficiente para que a sessão (relativamente curta) da aplicação seja terminada sem problemas. Desta forma, o simulador utiliza como peso apenas o identificador único dos nós. Portanto, a escolha de um nó de peso menor como escravo-ponte (ou para mestre de uma ponte intermediária) não afetam os resultados obtidos. O capítulo seguinte mostra os resultados obtidos nas simulações realizadas.

Capítulo 5

EXPERIMENTO E ANÁLISE DOS RESULTADOS

Para avaliar a redução do tempo médio de indisponibilidade da *scatternet* obtida com o novo protocolo proposto, foram realizados experimentos com o simulador descrito no Capítulo 4.

Utilizou-se uma duração fixa de sessão de 20 minutos (tempo sugerido no Capítulo 3), e variou-se o tamanho do ambiente em três configurações. Primeiramente, utilizou-se uma dimensão de 5 x 5 metros, que garante que todos os nós estão no alcance de transmissão uns dos outros uma vez que o alcance do *Bluetooth* é de 10 metros. Considera-se que alguns ambientes terão esta característica, por exemplo, uma sala de reuniões ou sala de espera. Após, foi utilizada uma dimensão de 24 x 4 metros, simulando o formato de, por exemplo, um vagão de trem ou outro veículo de transporte coletivo, que foi um dos cenários de uso visualizados. Por fim, foi considerada uma dimensão de 20 x 20 metros, representando um ambiente mais amplo. Também foi variado o número de nós presentes nos ambientes, nos valores de 10, 15, 20, 25 e 30 nós, que representam quantidades aproximadas de pessoas (usuários) nos ambientes que o modelo vislumbra.

As tabelas possuem as seguintes colunas. A primeira coluna indica o número de nós, ou seja, mostra a variação do número de nós nas redes dos experimentos. A segunda coluna BS (*BlueStars*) e terceira coluna MP (Modelo Proposto) mostram o tempo total de indisponibilidade da rede, ou seja, o tempo real gasto em manutenção (neste caso, reconstrução) da rede, ao invés de comunicação útil para a aplicação, que fica impossibilitada de funcionar durante este intervalo de tempo para cada um dos protocolos. A quarta coluna BS e a quinta coluna MP indicam o número de nós que falharam no tempo da aplicação, ou seja, mostra para situação simulada o número de quedas ou saída de nós na simulação proposta. A sexta coluna indica o número de nós isolados da rede após a queda de algum nó mestre ou nó ponte. Esta coluna só é referente ao protocolo proposto, pois ele não reconstrói a rede sempre que existe uma desconexão como é o caso do protocolo *BlueStars*. E por último a coluna que indica o número de quedas (ver item 8 da seção 4.2), ou seja, a soma dos nós que caíram e os

que ficaram fora da rede devido ao fato de terem ficado isolados da *Scatternet* conectada no protocolo modificado.

A Tabela 5.1 demonstra os resultados das simulações em uma rede de dimensões 5x5 quando existe uma variação no número de nós entre os valores de 10 e 30 levando em conta que uma ponte só poderia participar de duas *piconets*. Nesta tabela nota-se que a média aritmética de tempo em que a rede fica indisponível para mil iterações utilizando-se do protocolo proposto é muito reduzida se comparado com o tempo de indisponibilidade da rede que roda sobre o *BlueStars*. Por exemplo para 20 nós o tempo de indisponibilidade é de 25,67 segundos enquanto para o modelo proposto nesse trabalho fica em 4,98 segundos, ou seja 80% de economia de tempo indisponível em relação ao protocolo *BlueStars*.

Tabela 5.1: Simulação: área=5x5, nó pode participar de no máximo 2 *piconets* e tempo da aplicação igual a 20 minutos variando o número de nós

Nós	Ind. BS	Ind. MP	Falhos BS	Falhos MP	Isolados MP	Total Quedas MP
10	14,64	0,72	1,61	1,52	0,85	2,38
15	18,07	1,42	2,47	1,61	1,48	3,85
20	25,67	4,98	3,28	4,92	3,1	1,82
25	32,28	5,51	3,99	3,86	2,23	6,07
30	31,52	3,44	4,79	4,71	1,66	6,37

As tabelas 5.2 e 5.3 possuem apenas um parâmetro diferente da tabela anterior, que é o tamanho do ambiente em que é executado o experimento. Ou seja, na Tabela 5.2 o ambiente é de 24x4 metros, enquanto na Tabela 5.3 é de 20x20 metros.

Tabela 5.2: Simulação: área=24x4, nó pode participar de no máximo 2 *piconets* e tempo da aplicação igual a 20 minutos variando o número de nós

Nós	Ind. BS	Ind. MP	Falhos BS	Falhos MP	Isolados MP	Total Quedas MP
10	15,01	1,44	1,49	1,51	0,84	2,35
15	20,16	3,54	2,27	2,32	1,11	3,43
20	24,88	5,52	3,13	3,14	1,63	4,78
25	26,55	7,01	3,7	3,81	2,23	6,05
30	27,12	7,55	4,53	4,6	2,57	7,18

Pode ser visualizado a partir das tabelas 5.2 e 5.3 que, com o aumento do tamanho do ambiente, as quedas causam um tempo de indisponibilidade maior. Sendo assim como citado anteriormente o protocolo proposto funciona positivamente para o estudo de caso em que o mesmo visa atender, ou seja, aplicações onde o grafo de visibilidade seja “praticamente” completo, isso ocorre quando há uma grande concentração ou densidade de nós independente da área total considerada.

As tabelas 5.4, 5.5, e 5.6 possuem as mesmas variações de tamanho e número de nós das tabelas anteriores, o parâmetro variado nessas tabelas é o número de *piconets* a qual um nó pode pertencer,

Tabela 5.3: Simulação: área=20x20, nó pode participar de no máximo 2 *piconets* e tempo da aplicação igual a 20 minutos variando o número de nós

Nós	Ind. BS	Ind. MP	Falhos BS	Falhos MP	Isolados MP	Total Quedas MP
10	15,09	2,73	1,33	1,51	0,63	2,14
15	17,15	4,61	2,02	2,27	0,86	3,14
20	19,62	5,41	2,83	3,02	1,37	4,39
25	17,35	8,19	3,44	3,84	1,9	5,74
30	20,2	8,54	3,75	4,68	2,34	7,03

Tabela 5.4: Simulação: área=5x5, nó pode participar de no máximo 3 *piconets* e tempo da aplicação igual a 20 minutos variando o número de nós

Nós	Ind. BS	Ind. MP	Falhos BS	Falhos MP	Isolados MP	Total Quedas MP
10	14,7	0,22	1,62	1,55	0,77	2,32
15	19,03	0,05	2,44	2,41	0,42	2,84
20	25,24	1,48	3,15	1,6	3,24	4,85
25	31,14	1,68	3,92	4	1,51	5,51
30	32,43	0,89	4,73	4,77	0,94	5,71

Tabela 5.5: Simulação: área=24x4, nó pode participar de no máximo 3 *piconets* e tempo da aplicação igual a 20 minutos variando o número de nós

Nós	Ind. BS	Ind. MP	Falhos BS	Falhos MP	Isolados MP	Total Quedas MP
10	15,28	1,41	1,57	1,56	0,63	2,19
15	20,88	1,42	2,38	2,31	0,88	3,19
20	23,4	2,81	3,11	3,15	1,36	4,52
25	25,83	2,98	3,77	3,97	1,67	5,65
30	29,51	3,36	4,67	4,84	1,77	6,62

Tabela 5.6: Simulação: área=20x20, nó pode participar de no máximo 3 *piconets* e tempo da aplicação igual a 20 minutos variando o número de nós

Nós	Ind. BS	Ind. MP	Falhos BS	Falhos MP	Isolados MP	Total Quedas MP
10	15,22	2,47	1,3	1,41	0,48	1,89
15	18,04	2,4	2,02	2,36	0,63	3
20	18,18	3,23	2,67	3,19	1,09	4,28
25	20,69	3,79	3,62	4	1,36	5,36
30	21,82	4,19	4,15	3,79	1,94	6,8

que passa de 2 para 3. Nos gráficos das figuras 5.5, 5.6 e 5.7 pode ser melhor visualizado o comportamento das simulações das tabelas apresentadas. Os gráficos mostram a variação do tempo de indisponibilidade (eixo y) pela variação do número de nós na simulação (eixo x). Nos gráficos o

tempo de duração nas três simulações é de 20 minutos, existe uma variação dos tamanhos dos ambientes que são os mesmos das tabelas, ou seja, 5x5, 24x4 e 20x20 e mostram o funcionamento de redes *Bluetooth* utilizando o protocolo *BlueStars* e utilizando o modelo proposto com um nó podendo ser ponte para 2 e 3 *piconets*. As figuras 5.2, 5.3 e 5.4 mostram redes formadas a partir de uma rede *Bluetooth* após a fase de descoberta de dispositivos visualizada na Figura 5.1. A Figura 5.2 demonstra a rede *BlueStars* formada. Na Figura 5.3 pode ser visualizado uma rede formada com o protocolo proposto com um nó podendo ser ponte entre no máximo 2 *piconets*, enquanto na Figura 5.4 pode ser visto a rede formada com o protocolo proposto com um nó podendo ser ponte entre no máximo 3 *piconets*.

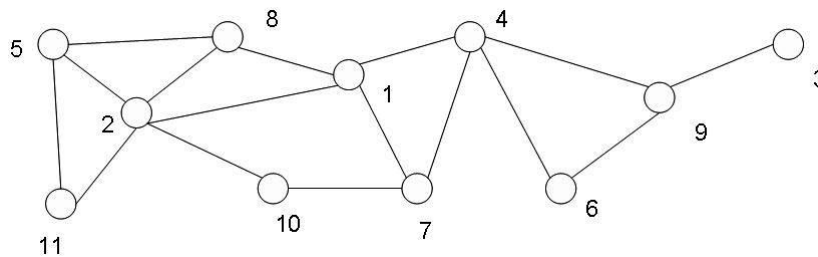


Figura 5.1: Grafo de visibilidade de uma rede *Bluetooth* após a fase de descoberta de dispositivos

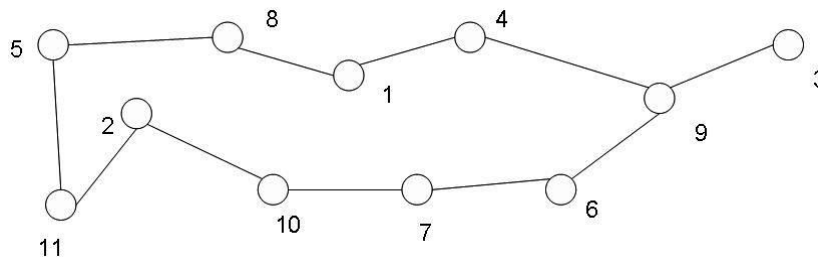


Figura 5.2: Rede *BlueStars* formada sobre a rede mostrada no grafo de visibilidade da Figura 5.1

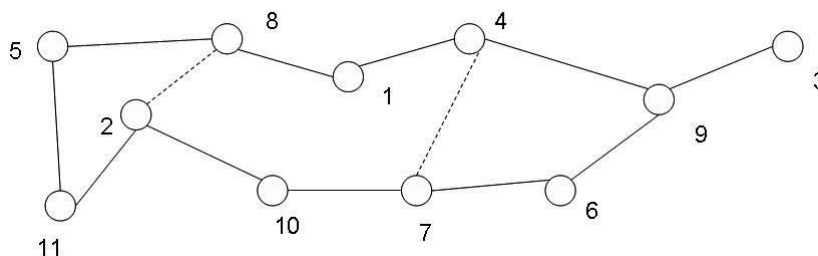


Figura 5.3: Rede do modelo proposto formada sobre a rede mostrada no grafo de visibilidade da Figura 5.1 com um nó fazendo parte de no máximo 2 *piconets*

Através dessas tabelas pode-se confirmar as constatações anteriores. Mas é possível visualizar que com o aumento do número de *piconets* ao qual o nó pode ser ponte há uma diminuição considerável do tempo em que a rede fica indisponível para o modelo proposto, além de uma diminuição do número

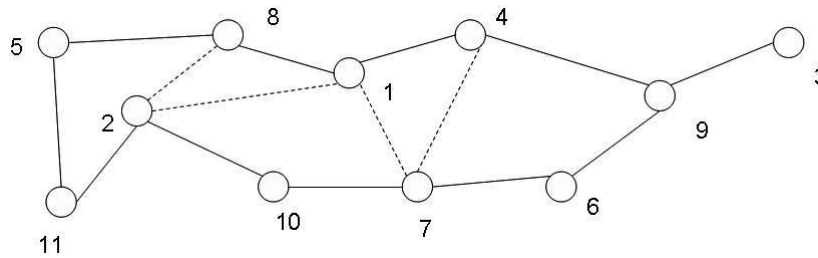


Figura 5.4: Rede do modelo proposto formada sobre a rede mostrada no grafo de visibilidade da Figura 5.1 com um nó fazendo parte de no máximo 3 *piconets*

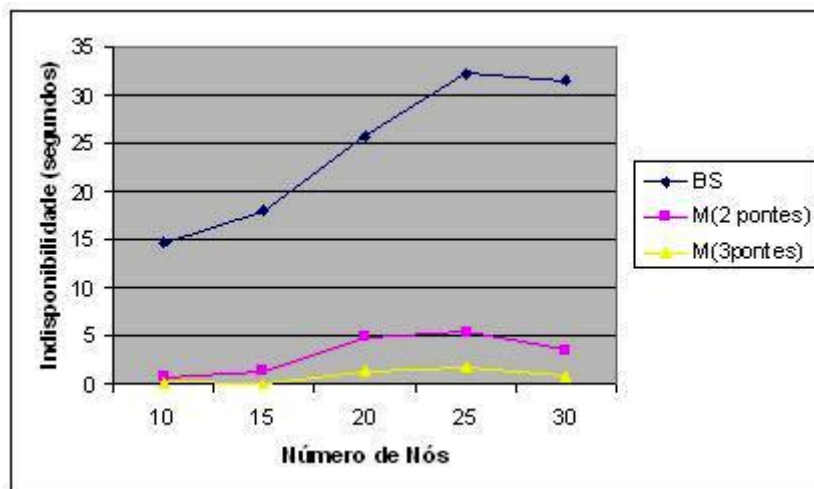


Figura 5.5: Gráfico de tempo de indisponibilidade por variação do número de nós com área=5x5 e tempo da aplicação igual a 20 minutos (tabelas 5.1 e 5.6)

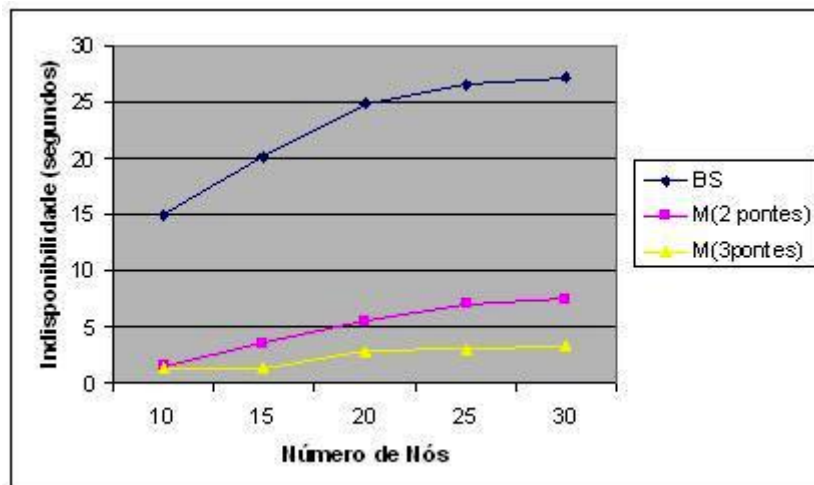


Figura 5.6: Gráfico de tempo de indisponibilidade por variação do número de nós com área=24x4 e tempo da aplicação igual a 20 minutos (tabelas 5.2 e 5.7)

total de quedas de nós da rede. Exemplos: na tabela 5.1 para 30 nós, com o nó sendo ponte em apenas duas *piconets*, a rede fica indisponível por 3,44 segundos e possui 6,37% de queda de nós em média,

enquanto na tabela 5.4 para 30 nós com um nó podendo participar de até 3 *piconets* a simulação mostra que o tempo indisponível é de 0,89 segundos e que a média de queda de nós cai para 5,71%.

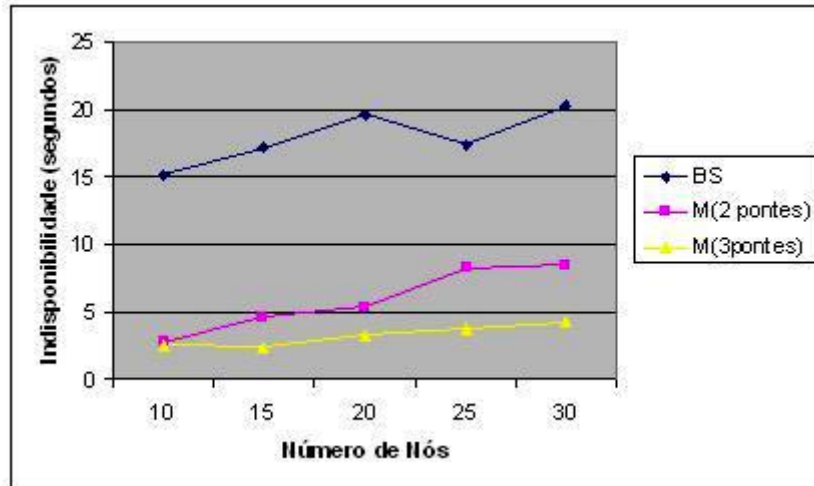


Figura 5.7: Gráfico de tempo de indisponibilidade por variação do número de nós com área=20x20 e tempo da aplicação igual a 20 minutos (tabelas 5.3 e 5.8)

Uma observação importante deve ser feita sobre o número de *piconets* a qual um nó pode fazer parte. Segundo [PET03], um nó começa a apresentar problemas como perda de pacotes e consumo de energia elevado quando um nó passa a fazer parte de 4 ou mais *piconets*, por isso as simulações foram desenvolvidas com um nó fazendo parte de 2 ou 3 *piconets*.

As tabelas 5.7 e 5.8 possuem a primeira coluna diferente das tabelas mostradas anteriormente. Nesta coluna é mostrada a variação de tempo (10, 20, 30, 60, 120 minutos) para três simulações. Os parâmetros para cada simulação são os seguintes: a tabela 5.7 possui um tamanho de ambiente de 5x5 metros, com cada ponte podendo fazer parte de no máximo 2 *piconets* e com o número de nós igual a 15. Na tabela 5.8 o ambiente é simulado com tamanho 24x4 metros, com cada nó participando de no máximo 3 *piconets* e com o número de nós 20. Nessas tabelas foi constatado que o número de nós que ficam fora da rede em aplicações, que possuem com um tempo maior em relação ao que o protocolo proposto foi projetado, é bastante elevado. Nesse ponto um aspecto precisa ser constatado, o modelo proposto neste trabalho, descarta os nós que após a queda de um mestre ou nó ponte não podem mais fazer parte da rede, ou seja, ficam fora da grafo de visibilidade dos nós que formam a *scatternet* conectada. Esta escolha foi feita com o objetivo de atender as aplicações como jogos, que possuam um tempo de duração reduzido, pois seria interessante para essas aplicações manterem o máximo de usuários que iniciaram a aplicação sem pausas e interrupções no tempo total da aplicação. Os valores foram obtidos a partir de pesquisa com consulta a programadores de jogos. Nessa consulta foi definido que entre 20 e 30 minutos seria o tempo ideal para uma partida "casual", 10 minutos um tempo curto e 120 minutos um tempo muito elevado para esse modelo de jogo.

Neste capítulo foram demonstrados os resultados encontrados para as simulações feitas sobre o protocolo proposto e o protocolo *BlueStars*. Todas as especificações usadas no simulador estão

baseadas no artigo sobre o *BlueStars* [PET03]. Como visto, foram simuladas diversas variações nas propriedades importantes para formação de uma *scatternet*.

Tabela 5.7: Simulação: área=5x5, nó pode participar de no máximo 2 *piconets* e número de nós igual a 15 variando o tempo da aplicação

Tempo	Ind. BS	Ind. MP	Falhos BS	Falhos MP	Isolados MP	Total Quedas MP
10	8,34	0,56	1,28	1,25	0,81	2,06
20	19,31	1,93	2,46	2,31	1,58	3,89
30	26,05	2,99	3,47	3,24	2,2	5,45
60	49,83	6,66	6,17	5,45	3,45	8,9
120	83,13	10,7	9,85	7,65	4,49	12,14

Tabela 5.8: Simulação: área=24x4, nó pode participar de no máximo 3 *piconets* e número de nós igual a 20 variando o tempo da aplicação

Tempo	Ind. BS	Ind. MP	Falhos BS	Falhos MP	Isolados MP	Total Quedas MP
10	12,23	1,04	1,6	1,72	0,76	2,49
20	23,38	3,31	3,07	3,22	1,31	4,54
30	35,79	5,35	4,51	4,5	1,74	6,24
60	71,69	11,82	7,96	7,52	3,16	10,69
120	113,87	18,34	12,6	10,85	4,82	15,68

Os resultados mostram que o modelo proposto em situações como a do estudo de caso especificado possui uma melhoria significativa no tempo de indisponibilidade da rede. Deve ser salientado que os resultados apresentados são as médias aritméticas para mil simulações em cada uma das propriedades apresentadas. Em um caso específico quando existe a possibilidade de reconstrução da rede em pelo menos três oportunidades, o tempo gasto pelo protocolo existente é de 84 segundos, o que é muito elevado para a aplicação proposta. Pelos resultados apresentados o protocolo proposto possui um ganho de 80% na média de tempo de indisponibilidade com relação ao *BlueStars*, para a classe de aplicação definida, e o tempo em que a rede ficaria indisponível seria de no máximo 16,8 segundos. Esse tempo para o protocolo proposto seria menor que uma reconstrução da rede para o protocolo existente que é de 28 segundos.

Capítulo 6

CONCLUSÕES

O presente trabalho objetivou a modelagem de um protocolo de formação de redes *ad hoc* para implementação em hardware na plataforma *Fênix*. Devido a indisponibilidade do SoC para validação do protocolo na *Fênix*, foram realizadas atividades de simulação. Para essas atividades foi utilizada inicialmente a ferramenta ns-2 e a seguir foi desenvolvido um simulador especialmente para esse fim.

Após as simulações obteve-se um protocolo para estabelecimento de uma rede *ad hoc wireless multihop* de dispositivos *Bluetooth*. Esse protocolo assegura uma descoberta local de topologia apropriada, permite que os dispositivos se auto-organizem em *piconets*, e permite a interconexão das *piconets* formadas em uma única *scatternet* conectada. O protocolo proposto é baseado no protocolo *BlueStars*, um dos protocolos estudados no início do trabalho e analisado no Capítulo 2.

As fases do protocolo são descritas, levando em conta a tecnologia *Bluetooth* conforme a última versão liberada das especificações. A *scatternet* obtida tem trajetos múltiplos entre cada par de nós e a operação do protocolo não necessita que todos os dispositivos estejam cada um no alcance de transmissão do outro. Isto é bom para a aplicação-alvo do trabalho, que são jogos casuais em dispositivos móveis interconectados por uma rede *ad hoc Bluetooth*, como discutido no estudo de caso apresentado no Capítulo 3.

No Capítulo 4, foi descrita a implementação do simulador, utilizado nos experimentos e nos resultados obtidos dos mesmos, apresentados no Capítulo 5. O Capítulo 5 mostra os resultados para simulações com a variação de propriedades como: tamanho do ambiente, tempo da aplicação, número de *piconets* da qual um nó pode fazer parte e número de nós no ambiente. As simulações visaram obter resultados referentes ao tempo de indisponibilidade de uma *scatternet* devido à execução de operações de manutenção da mesma que são realizadas pelos protocolos de formação e de manutenção da rede *ad hoc*.

Através dos resultados obtidos na simulação foi concluído que existe a viabilidade de utilização do modelo proposto em ambientes de tamanho reduzido ou ambientes com um número elevado de nós ocupando a mesma área, ou seja, formando assim um grafo de visibilidade "quase" completo. Para esses ambientes o tempo de indisponibilidade é em média 80% menor nas redes que tem sua

formação com o protocolo proposto, se comparado com as redes que se formam com o *BlueStars*.

Uma desvantagem constatada nas simulações é que em aplicações realizadas por um tempo maior que o projetado inicialmente, que é na ordem de 20 minutos, como por exemplo 120 minutos (tabelas 5.7 e 5.8) o número de nós que ficam fora no decorrer da aplicação é muito grande. Isso acontece devido, como já explicado anteriormente, devido ao modelo proposto neste trabalho descartar os nós que após a queda de um mestre ou nó ponte não poderem mais fazer parte da rede, ou seja, ficam fora da grafo de visibilidade dos nós que formam a *scatternet* conectada. Para solução deste problema pode ser implementada alguma técnica que possibilite a entrada de nós na rede depois das três fases do protocolo terem sido executadas. Essas técnicas não foram expostas ou aplicadas ao modelo proposto devido a simulação ter sido feita sobre o protocolo *BlueStars* e esse trabalho [PET03] não atender a essas necessidades. Sendo assim as especificações para esse tipo de operação não estavam a disposição o que impediu que fossem simuladas essas situações, para que os resultados da simulação fossem reais.

Como trabalho futuro existe a possibilidade de criação de uma extensão *Bluetooth* para o software ns-2 com código livre, facilitando assim o desenvolvimento de melhorias para este e outros protocolos de formação de *scatternets*. Com isso poderiam ser executadas simulações como a entrada de nós, que como mencionado anteriormente pode melhorar os resultados do protocolo proposto se incorporado ao mesmo. Levando a diminuição do número de nós que ficam fora da aplicação quando o tempo não é reduzido, ou seja, na ordem de 20 ou 30 minutos.

Um projeto para um trabalho futuro, é a realização de uma pesquisa para disponibilizar ao protocolo a entrada de nós em uma *scatternet* já formada, essa possibilidade não foi explorada no modelo proposto, pois a aplicação de jogos, conforme pesquisa realizada, normalmente não suporta a entrada de usuários no decorrer da aplicação. E, além disso, pode ser difícil a inclusão desta característica. Por exemplo, o protocolo *BlueStars* só suporta a inclusão de novos nós na rede quando uma nova fase de reconstrução da mesma é iniciada, o que normalmente só acontece quando um nó falha ou sai do alcance da rede. Para suportar a inclusão automática de novos dispositivos na *Scatternet* do jogo, seria necessário que, por exemplo, os dispositivos já presentes no jogo entrassem, periodicamente, no modo *Inquiry*. Isso pode prejudicar, de alguma forma, a comunicação normal do dispositivo com a *scatternet*, ou talvez aumentar significativamente o consumo de bateria. Isto não foi explorado neste trabalho e fica como possibilidade de trabalho futuro.

E por último a possibilidade do protocolo ao iniciar verificar a capacidade de processamento e a energia disponível no dispositivo e atribuir pesos. Esses pesos seriam repassados nas mensagens junto com o ID, possibilitando assim melhorias na escolha de mestres e pontes.

REFERÊNCIAS BIBLIOGRÁFICAS

- [AKY05] I. F. Akyildiz, X. Wang, W. Wang. “Wireless mesh networks: a survey”. *Computer Networks*, vol. 47-4, Março 2005, pp.445–487.
- [BAA02] S. Baatz, C. Bieschke, M. Frank, C. Köhl, P. Martini, C. Scholz. “Building Efficient Bluetooth Scatternet Topologies from 1-Factors”. In: *International Conference on Wireless and Optical Communications*, 2002, 6p.
- [BAS04] S. Basagni, R. Bruno, G. Mambriani, C. Petrioli. “Comparative performance evaluation of scatternet formation protocols for networks of bluetooth devices”. *Wireless Networks*, vol. 10-2, Março 2004, pp. 197–213.
- [BLU04] Bluetooth Special Interest Group. “Bluetooth Specification Version 2.0”. Capturado em: <http://www.bluetooth.com/Bluetooth/Technology/Building/Specifications/>, Fevereiro 2007.
- [BRU05] R. Bruno, M. Conti, E. Gregori. “Mesh networks: commodity multihop ad hoc networks”. *IEEE Communications Magazine*, vol. 43-3, Março 2005, pp. 123–131.
- [BUL04] N. Bulusu, J. Heidemann, D. Estrin, T. Tran. “Self-configuring localization systems: Design and Experimental Evaluation”. *Transactions in Embedded Computing Systems*, vol 3-1, Fevereiro 2004, pp. 24–60.
- [CHI03] C. F. Chiasserini, M. A. Marsan, E. Baralis, P.Garza. “Towards Feasible Topology Formation Algorithms for Bluetooth-based WPANs”. In: *Proceedings of the 36^a Annual Hawaii International Conference on System Sciences*, 2003, 10p.
- [CHL99] I. Chlamtac, A. Faragó. “A new approach to the design and analysis of peer-to-peer mobile networks”. *Wireless Networks*, vol. 5-3, Maio 1999, pp. 149–156.
- [COR90] T. H. Cormen, C. E. Leiserson, R. L. Rivest. “*Introduction to Algorithms*”. Cambridge: The MIT Press, 1990, 1^a edição, 1048p.
- [FAL06] K. Fall, K. Varadhan. “The ns Manual (Formerly ns Notes and Documentation)”, Capturado em: http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf, Fevereiro 2006.

- [HIL04] J. Hill, M. Horton, R. Kling, L. Krishnamurthy. “The platforms enabling wireless sensor networks”. *Communications of the ACM*, vol. 47-6, Junho 2004, pp. 41–46.
- [HSI06] T. Hsieh, R. Chang. “Bluegon: a polygon-shaped scatternet formation algorithm for Bluetooth: Research Articles”. *Wireless Communications & Mobile Computing*, vol. 6-4, Junho 2006, pp. 439–451.
- [LAW03] C. Law, A. K. Mehta, K. Siu. “A new Bluetooth scatternet formation protocol”. *Mobile Networks and Applications*, vol. 8-5, Outubro 2003, pp. 485–498.
- [LI04] X. Li, I. Stojmenovic, Y. Wang. “Partial Delaunay Triangulation and Degree Limited Localized Bluetooth Scatternet Formation”. *IEEE Transactions on Parallel and Distributed Systems*, vol. 15-4, Abril 2004, pp. 350–361.
- [MAR02] M. A. Marsan, C. Chiasserini, A. Nucci, G. Carello, L. De Giovanni. “Optimizing the Topology of Bluetooth Wireless Personal Area Networks”. In: 21^a Annual Joint Conference of the IEEE Computer and Communications, 2002, 8p.
- [MAR04] K. Martinez, J. K. Hart, R. Ong. “Environmental Sensor Networks”. *IEEE Computer*, vol. 37-8, Agosto 2004, pp. 50–56.
- [PER04] E. G. Perera, V. Sivaraman, A. Seneviratne. “Survey on Network Mobility Support”. *Mobile Computing and Communications Review*, vol. 8-2, Abril 2004, pp 7–19.
- [PET03] C. Petrioli, S. Basagni, I. Chlamtac. “Configuring BlueStars: Multihop Scatternet Formation for Bluetooth Networks”. *IEEE Transactions on Computers*, vol. 52-6, Junho 2003, pp. 779–790 .
- [RUI03] L. B. Ruiz, J. S. Nogueira, A. A. F. Loureiro. “Functional and Information Models for the MANNA Architecture”. In: *Proceedings of the 5th French Conference on Networks and Services Management*, 2003, 15p.
- [SAL01] T. Salonidis, P. Bhagwat, L. Tassiulas, R. O. LaMaire. “Distributed Topology Construction of Bluetooth Personal Area Networks”. In: 20^a Annual Joint Conference of the IEEE Computer and Communications Societies, 2001, 10p.
- [SUB06] A. P. Subramanian, M. M. Buddhikot, S. Miller. “Interference aware routing in multi-radio wireless mesh networks”. In: 2^o IEEE Workshop on Wireless Mesh Networks, 2006, 9p.
- [SZE04] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, D. Estrin. “Habitat monitoring with sensor networks”. *Communications of the ACM*, vol. 47-6, Junho 2004, pp. 34–40.

- [THO04] B. Thorstensen, T. Syversen, T. Walseth. “Electronic shepherd - a low-cost, low-bandwidth, wireless network system”. In: Proceedings of the 2^a international conference on Mobile systems, applications, and services, 2004, 10p.
- [WAN02] Z. Wang, R. J. Thomas, Z. Haas. “Bluenet - A New Scatternet Formation Scheme”. In: Proceedings of the 36^a Annual Hawaii International Conference on System Sciences, 2002, 9p.
- [WEL05] P. McDermott-Wells. “What is Bluetooth?”. IEEE Potentials, vol 23-5, Dezembro 2004-Janeiro 2005, pp. 33–35.
- [WOO04] A. Woo, S. Madden, R. Govindan. “Networking support for query processing in sensor networks”. Communications of the ACM, vol. 47-6, Junho 2004, pp. 47–52.
- [YAO82] A. C. Yao. “On Constructing Minimum Spanning Trees in k-Dimensional Spaces and Related Problems”. Technical Reports, Stanford University, 1982, 16p.
- [ZAR01] G. V. Zaruba, S. Basagni, I. Chlamtac. “Bluetrees-scatternet formation to enable Bluetooth-based ad hoc networks”. In: IEEE International Conference on Communications, 2001, 5p.