

EDUARDO HABIB BECHELANE MAIA

**PROPOSTA DE UM NOVO ALGORITMO DE
ROTEAMENTO PARA REDES DE SENSORES SEM FIO E
O ESTUDO DE UMA TÉCNICA PARA SE PROVER QOS
NESTAS REDES**

Belo Horizonte
30 de janeiro de 2006

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**PROPOSTA DE UM NOVO ALGORITMO DE
ROTEAMENTO PARA REDES DE SENSORES SEM FIO E
O ESTUDO DE UMA TÉCNICA PARA SE PROVER QOS
NESTAS REDES**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

EDUARDO HABIB BECHELANE MAIA

Belo Horizonte
30 de janeiro de 2006



UNIVERSIDADE FEDERAL DE MINAS GERAIS

FOLHA DE APROVAÇÃO

Proposta de um novo algoritmo de roteamento para Redes de Sensores Sem Fio e o estudo de uma técnica para se prover QoS nestas redes

EDUARDO HABIB BECHELANE MAIA

Dissertação defendida e aprovada pela banca examinadora constituída por:

Ph. D. ANTONIO ALFREDO FERREIRA LOUREIRO – Orientador
Universidade Federal de Minas Gerais

M. Sc. DANIEL CÂMARA – Co-orientador
Universidade Federal de Minas Gerais

Ph. D. RAQUEL APARECIDA DE FREITAS MINI
Pontifícia Universidade Católica de Minas Gerais

Ph. D. LINNYER BEATRYS RUIZ
Universidade Federal de Minas Gerais

Ph. D. GERALDO ROBSON MATEUS
Universidade Federal de Minas Gerais

Belo Horizonte, 30 de janeiro de 2006

Resumo

Este trabalho apresenta duas contribuições principais. A primeira propõe um novo e eficiente algoritmo de roteamento para Redes de Sensores Sem Fio chamado Inter Cluster Routing Algorithm (ICA) baseado no algoritmo LEACH (Low-Energy Adaptive Clustering Hierarchy) [1]. O ICA foi desenvolvido para redes de sensores que coletam informações periodicamente durante o tempo de vida da rede. Para estes ambientes a longevidade da rede é normalmente um fator importante. Os resultados de simulação mostram que o ICA, quando comparado ao LEACH e ao LEACH-C [2], apresenta não somente um tempo maior de vida da rede como também um maior número de pacotes transmitidos e um consumo mais homogêneo de energia na rede. A segunda contribuição deste trabalho trata do estudo de uma técnica para melhorar os parâmetros de QoS da rede para os pacotes prioritários. Os resultados das simulações mostram que utilizando esta técnica reduzimos a perda de pacotes prioritários em até 68,5%. Este resultado é apenas 14,36% pior do que o resultado ótimo. Demonstra-se ainda, através de experimentos com um tráfego heterogêneo, que a técnica apresentada é capaz de tratar cada pacote de acordo com seus requisitos específicos de QoS.

Abstract

This work presents two main contributions. First, it shows the Inter Cluster Routing Algorithm (ICA), a new and efficient routing algorithm for wireless sensor networks based on the Low-Energy Adaptive Clustering Hierarchy (LEACH) [1] algorithm. The proposed algorithm, when compared with LEACH and LEACH-C [2], presents not only higher network lifetime but also a higher number of delivered messages. The energy maps also show that ICA presents a better energy distribution and degradation, when compared to LEACH and LEACH-C [2]. ICA is indicated to sensor networks that collect data periodically and in which the network lifetime is an important parameter. The second contribution of this work is the Overlay for WireLess sensor networks (OWL) algorithm. We show that the use of overlay networks on Wireless Sensor Networks (WSN) can increase the Quality of Service (QoS) of such networks at a low cost. In our experiments we observe that the use of overlay networks over WSNs can decrease the packet loss of priority packets, in 68.5% being only 14.36% higher than the best possible result. Experiments with heterogeneous traffic in terms of QoS requests show that OWL can handle such traffic and treat each packet in accordance to its QoS requests. In this way we can have a cheap way to increase the QoS over Wireless Sensor Networks using, basically, the already existent protocols and structure, even for networks with packets with different QoS requests.

A Deus, meus pais, irmãos, Ellen, amigos e professores.

Agradecimentos

Meus agradecimentos especiais ao professor Antônio Alfredo Ferreira Loureiro por ter me orientado neste trabalho. Agradeço pelo incentivo, paciência, dedicação e exemplo. Agradeço ainda por ter me incentivado a ingressar no mestrado e pela compreensão e suporte indispensáveis durante toda a minha vivência na área. Sou honrado em tê-lo tido como orientador.

Agradecimentos especiais a Daniel Câmara, por sua constante disposição, orientação, incentivo, disponibilidade e amizade em todos os momentos. Agradeço, ainda, pela confiança mais uma vez depositada no meu trabalho, tendo em vista que foi meu co-orientador também na Disciplina Projeto Orientado em Computação II.

Agradeço ao professor Geraldo Robson Mateus por acreditar em mim ao me acolher no meu primeiro estágio e, após isso, ter se disponibilizado a ser meu orientador na Iniciação científica. Foi com você que eu comecei a desenvolver uma enorme curiosidade pela busca do novo. Obrigado também pelos seus conselhos. Foi graças a eles que eu comecei a pensar em ingressar no mestrado.

Meus agradecimentos ao Synergia por ter me dado o suporte financeiro necessário para vencer mais esta etapa de minha vida.

À professora Linnyer Beatrys Ruiz pela co-orientação na disciplina de graduação Projeto Orientado em Computação I. Foi a partir do estudo feito nesta disciplina que conseguimos a base para desenvolver a primeira parte deste trabalho.

Agradecimentos especiais ao Departamento de Ciência da Computação e à UFMG pela excelência nos seus cursos de graduação e pós-graduação.

A todos os meus amigos, entre eles André, Stenio, Issa, Emanuel, Wagner, Gislaine, Piti, Carol, Paula, Xande, Gleiton, Gilson, Juneo e João, pelas conversas e pelo apoio.

À minha família, aos meus amigos e todos aqueles que várias vezes abandonei, mas jamais

esqueci nestes dois anos de muito trabalho.

Aos meus irmãos e a Ellen pelo constante apoio e carinho.

Agradecimentos especiais e imensuráveis a meus pais que me proporcionaram tudo do bom e do melhor. Gostaria de deixar claro que só cheguei onde estou porque vocês lutaram muito e não mediram esforços para que eu já começasse na frente. Amo vocês.

A Deus, que iluminou meu caminho e colocou essas pessoas extraordinárias em minha vida.

Sumário

1	Introdução	1
1.1	Visão Geral	1
1.2	Roteamento	2
1.3	Qualidade de Serviço	3
1.4	Contribuições	3
1.4.1	ICA	4
1.4.2	Overlay for WireLess networks	4
1.5	Organização	4
2	Trabalhos Relacionados	6
2.1	Algoritmos de Roteamento	6
2.1.1	Roteamento em redes cabeadas	6
2.1.2	Roteamento em redes ad hoc	7
2.1.3	Roteamento em RSSFs	8
2.2	Qualidade de Serviço	15
2.2.1	QoS em redes cabeadas	15
2.2.2	QoS em redes sem fio	17
2.2.3	QoS em RSSFs	19
2.3	Contribuições desse trabalho	21
3	ICA - Inter Cluster Algorithm	23
3.1	Introdução	23
3.2	Fase de formação dos grupos	25
3.3	Fase de coleta de dados	27

3.4	ICA pseudo-código	35
4	Análise e simulação do ICA	37
4.1	Ambiente utilizado	37
4.2	Cenários de simulação	37
4.3	Resultados	39
4.4	Observações sobre o ICA	47
5	Overlay for WireLess networks (OWL)	50
5.1	Utilizando RONS em RSSFs	51
5.2	Arquitetura da rede	54
5.3	Overlay em RSSFs	54
5.4	Pseudo-código	55
5.4.1	Nós sensores	56
5.4.2	ERB	57
5.5	Custos	58
5.5.1	Custo de espaço	58
5.5.2	Custo de processamento	59
5.5.3	Custo de transmissão	60
6	Análise e simulação da técnica OWL	61
6.1	Ambiente utilizado	61
6.2	Cenário de simulação	61
6.3	Resultados	62
7	Conclusões e trabalhos futuros	71
	Referências Bibliográficas	74

Lista de Figuras

2.1	Exemplo de agrupamento no LEACH	12
2.2	Princípio do funcionamento do PEGASIS	14
2.3	Exemplo do funcionamento do protocolo proposto em [3]	18
3.1	Imagem de dois <i>rounds</i> consecutivos. Em (a) temos a formação de grupos em um <i>round</i> e em (b) temos a formação de grupos no <i>round</i> seguinte	27
3.2	Fase de formação dos grupos	28
3.3	Fase de coleta de dados	29
3.4	Diagrama das mensagens trocadas entre a ERB e um líder	31
3.5	Diagrama das mensagens trocadas entre um líder e um nó sensor	32
3.6	Diagrama das mensagens do ICA para a comunicação entre líderes	32
3.7	Cenário em que o líder do grupo 1 se recusa a servir de ponte para o grupo 2	33
3.8	Diagrama das mensagens do ICA durante a negação de serviço	34
4.1	Dados transmitidos	40
4.2	Dados transmitidos no tempo	41
4.3	Tempo médio de desligamento do primeiro nó	41
4.4	Tempo médio de vida da rede	42
4.5	Mapa de Energia para o LEACH na rede 5. (a) no tempo 100s e (b) no tempo 200s. O eixo Z representa a energia restante na rede.	43
4.6	Mapa de Energia para o LEACH-C, na rede 5 (a) no tempo 100s, (b) no tempo 200s e (c) no tempo 300s	44
4.7	Mapa de Energia para o ICA, na rede 5, nos tempos (a) 100s, (b) 300s, (c) 600s, (d) 900s, (e) 1000s	45
4.8	Dados transmitidos para as redes 4 e 5 com a ERB na posição (50,175)	46

4.9	Tempo médio de desligamento do primeiro nó com a ERB na posição (50,175) . .	47
4.10	Tempo médio de vida da rede com a ERB na posição (50,175)	48
4.11	Mapa de Energia para o ICA na rede 4 no tempo t=600s, com a ERB na posição (50, 175)	49
5.1	Alcance de transmissão de diferentes nós sensores	53
5.2	Mesmo cenário da figura 5.1 mas mostrando a rede <i>overlay</i> formada pela conexão entre os nós	53
5.3	Esquema genérico da mensagem	58
5.4	Esquema da mensagem prioritária	59
5.5	Esquema da mensagem não prioritária	59
6.1	Pacotes prioritários totais perdidos	63
6.2	Pacotes prioritários totais perdidos com o tráfego heterogêneo	64
6.3	Número não acumulativo de pacotes perdidos	66
6.4	Energia consumida quando os três tipos de pacotes coexistem na rede	67
6.5	Atraso médio dos pacotes prioritários para o cenário padrão	68
6.6	Atraso médio dos pacotes prioritários caso o parâmetro de maior prioridade seja o atraso dos pacotes	68
6.7	Atraso médio dos pacotes prioritários versus Tempo	69
6.8	Atraso médio quando os três tipos de pacotes coexistem na rede	70

Lista de Tabelas

4.1	Densidade das redes simuladas	38
4.2	Parâmetros das simulações	39
4.3	Variância dos dados transmitidos	40
4.4	Variância do tempo de desligamento do primeiro nó	42
4.5	Variância do tempo de vida da rede	42
4.6	Variância dos dados totais transmitidos em uma rede com a ERB na posição (50,175)	46
4.7	Variância do tempo de desligamento do primeiro nó na rede em que a ERB está na posição (50,175)	47
4.8	Variância do tempo de vida na rede em que a ERB está na posição (50,175) . . .	48
6.1	Pacotes prioritários perdidos	64
6.2	Variância dos pacotes prioritários perdidos	65
6.3	Atraso dos pacotes prioritários	65
6.4	Variância no atraso dos pacotes prioritários	65

Capítulo 1

Introdução

Este trabalho concentra-se no problema do roteamento em Redes de Sensores Sem Fios (RSSFs). Serão apresentadas duas contribuições principais: Na primeira, demonstra-se um novo algoritmo de roteamento para RSSFs e, na segunda, avalia-se uma técnica que permite prover qualidade de serviço em Redes de Sensores Sem Fio.

1.1 Visão Geral

Rede de Sensor Sem Fio é um tipo especial de rede ad hoc. Contudo alguns recursos dessas redes como banda, energia e processamento são mais escassos do que em uma rede ad hoc. Como as RSSFs podem estar localizadas em locais de difícil acesso é necessário que elas sejam tolerantes a falhas para que a necessidade de manutenção seja minimizada [4].

Além dos nós sensores essas redes possuem um ou mais nós sorvedouros, responsáveis por receber as informações sensoriadas. Os elementos computacionais dessas redes estão normalmente interligados através de meios não guiados e os nós sensores possuem quantidade de energia, capacidade computacional e memória extremamente limitados [5]. Pode existir, nas RSSFs, uma grande variação nas características básicas das redes tais como número, capacidade e autonomia dos elementos interligados.

Como exemplos de aplicações para redes de sensores pode-se citar a monitoração de ambientes e aplicações militares. Na monitoração de ambientes, por exemplo, o sensoriamento da temperatura de uma floresta seria de grande valia para detectar incêndios em estágio inicial. Estes sensores, ao serem depositados, coletariam dados sobre temperatura, umidade relativa

do ar, concentração de gás carbônico, dentre outros, formariam uma rede sem fio e poderiam se comunicar com a rede fixa informando a ocorrência de incêndios. Isto poderia preservar florestas, animais e vidas humanas. Além disso, como o incêndio seria detectado em estágio inicial, poderia haver uma economia de dinheiro no combate ao incêndio e no reflorestamento das áreas queimadas. Aplicações de segurança como a monitoração da estrutura de edifícios em caso de terremoto, ou de um desastre similar, também podem ser úteis.

1.2 Roteamento

Para que os dados monitorados pelos nós sensores se propaguem até a Estação Rádio Base (ERB), é necessário que sejam estabelecidas rotas entre a origem e o destino. Os protocolos que têm como objetivo determinar uma rota entre a origem e o destino são chamados de protocolos de roteamento. Existem diversas propostas de algoritmos de roteamento para redes cabeadas [6].

Em redes ad hoc o roteamento é, assim como nas redes cabeadas, uma área de pesquisa bastante ativa. Muitos dos algoritmos de roteamento propostos para redes ad hoc [7] se baseiam, de alguma forma, em algoritmos desenvolvidos para redes cabeadas. Um grande problema a ser levado em consideração ao se desenvolver protocolos de roteamento para redes ad hoc é a própria natureza da comunicação sem fio. A banda é bastante limitada e a quantidade de erros é alta. O meio sem fio é suscetível a interferências de outras fontes de energia. Uma outra limitação é que a maioria dos nós possuem recursos restritos de energia. Desta forma, algoritmos de roteamento eficientes devem levar em consideração estas limitações no momento do seu desenvolvimento.

As RSSFs se assemelham às redes sem fio pelo fato de o caminho entre a origem e o destino ser, na maioria das vezes, *multi-hop*. Entretanto, os requisitos dessas redes diferem em alguns aspectos das redes sem fio [8]. Em primeiro lugar, a forma típica de comunicação em uma RSSF é unidirecional no sentido dos nós sensores para a ERB. Segundo, como os dados de nós vizinhos podem se referir a fenômenos comuns, pode haver redundância de dados. Terceiro, normalmente os nós sensores possuem pouca ou nenhuma mobilidade. Finalmente, como principal restrição, a energia disponível para os nós é bastante limitada. Estas diferenças devem ser levadas em consideração no momento de se desenvolver um protocolo de roteamento

para as RSSFs.

No capítulo 3 será apresentada a proposta de um algoritmo que atende a esses requisitos e é mais eficiente do que outros algoritmos semelhantes.

1.3 Qualidade de Serviço

Um importante fator a ser considerado durante o projeto de novos protocolos para redes de sensores é a possibilidade de se prover Qualidade de Serviço (QoS). Como os nós sensores são bastante suscetíveis a falhas, seria importante poder transmitir dados com QoS nessas redes.

Para prover QoS em uma rede de comunicação, é necessário a análise e a manutenção de rotas que satisfaçam restrições de QoS como, por exemplo, atraso, banda mínima, custo de energia e perda de pacotes. Entretanto, para que essas restrições sejam satisfeitas são necessárias a obtenção e o gerenciamento de dados sobre os enlaces da rede. Porém, esses dados podem ser de difícil obtenção.

As redes *overlay*, que são redes lógicas construídas sobre redes físicas, têm sido utilizadas para prover garantias de QoS em redes estruturadas [9][10]. Nas redes *overlay*, os nós agem de forma cooperativa e repassam os pacotes para um caminho que difere do roteamento padrão. Várias aplicações de redes *overlay* podem ser encontradas em redes P2P [11], serviços multicast [12], redes de distribuição de conteúdo [13] e *Resilient Overlay Networks* (RONs) [9].

1.4 Contribuições

Este trabalho se concentra no problema do roteamento em RSSFs. Nossas principais contribuições são:

- O algoritmo de roteamento ICA (*Inter Cluster Algorithm*) que é eficiente no consumo de energia e pode fazer agregação de dados.
- A proposta da técnica OWL (*Overlay for WireLess Networks*) para prover QoS em RSSFs com a utilização de Redes *Overlay* (RONs).

1.4.1 ICA

No capítulo 3 será apresentado um novo e eficiente algoritmo de roteamento para redes de sensores sem fio chamado *Inter Cluster Routing Algorithm* (ICA). O ICA é baseado no *Low-Energy Adaptive Clustering Hierarchy* (LEACH) [2] e foi desenvolvido para redes de sensores que coletam informações periodicamente durante o tempo de vida da rede. Para este tipo de cenário, a longevidade da rede é, normalmente, um fator de extrema importância. Os resultados de simulações mostram que o ICA quando comparado ao LEACH e ao LEACH-C [2] apresenta não somente um tempo maior de vida da rede como também um maior número de pacotes transmitidos e uma distribuição mais homogênea do consumo de energia na rede. Os resultados dessa contribuição foram publicados em [14].

1.4.2 Overlay for WireLess networks

No capítulo 5 será apresentado o *Overlay for WireLess networks* (OWL). O OWL é uma técnica que permite prover QoS em RSSFs. Mostraremos usando simulações que o uso de redes *overlay* em RSSFs pode incrementar a Qualidade de Serviço destas redes a um custo baixo. Observa-se nos experimentos realizados que o uso de redes *overlay* em RSSFs pode diminuir a perda de pacotes prioritários em até 68,5% sendo, esse resultado, somente 14,5% pior do que o melhor resultado possível. Será apresentado também o resultado de experimentos com tráfegos com diferentes requisitos de QoS demonstrando que a técnica apresentada é capaz de tratar cada pacote de acordo com seu requisito de QoS. Mesmo para o caso de haver pacotes com diferentes requisitos de QoS trafegando pela rede argumenta-se que o custo de implementação desta técnica é baixo, se comparado com a melhora na Qualidade de Serviço obtida pela rede.

1.5 Organização

O restante deste trabalho está organizado da seguinte maneira: O capítulo 2 apresenta os trabalhos relacionados ao nosso trabalho. O capítulo 3 descreve um novo algoritmo de roteamento que é a primeira contribuição deste trabalho. No capítulo 4 é feita a análise e a simulação do algoritmo proposto no capítulo 3. No capítulo 5 propomos a utilização de redes *overlay* como forma de suportar QoS em RSSFs. O capítulo 6 apresenta os resultados das

simulações demonstrando a viabilidade de utilizar a técnica proposta no capítulo 5 bem como uma cuidadosa discussão destes resultados. Para finalizar, no capítulo 7, apresentamos as conclusões e possíveis trabalhos futuros.

Capítulo 2

Trabalhos Relacionados

Temos três áreas principais em termos de trabalhos relacionados ao nosso trabalho: roteamento, qualidade de serviço e redes *overlay*. Nas próximas seções citaremos alguns destes trabalhos.

2.1 Algoritmos de Roteamento

Para que os dados que trafegam por uma rede cheguem ao seu destino é necessário que se descubram rotas entre a origem e o destino. Os protocolos responsáveis por determinar estas rotas são chamados de protocolos de roteamento.

2.1.1 Roteamento em redes cabeadas

Muitos dos algoritmos de roteamento desenvolvidos para redes sem fio se baseiam em algoritmos para redes cabeadas. Esses algoritmos podem ser classificados de diversas maneiras. Uma divisão comum são os algoritmos baseados em vetor de distâncias (*Distance Vector Routing Protocols*) e os algoritmos baseados no estado do enlace (*Link State Routing Protocol*) [15].

Os algoritmos baseados em vetor de distâncias mantêm tabelas de roteamento com informações que permitem determinar o próximo nó em uma comunicação e a distância para qualquer nó da rede. A atualização da tabela de rotas é feita pelo recebimento, por um nó, das tabelas dos seus vizinhos. Ele compara, então, a sua tabela de rotas com os dados de seus vizinhos e verifica se algum de seus vizinhos conhecem uma rota de custo menor. Se isso ocorrer o nó atualiza a sua própria tabela de rotas.

Os algoritmos baseados no estado do enlace são melhores para redes em que os enlaces da rede não variam muito (redes estáveis), pois nestes algoritmos os nós não enviam a difusão periódica de toda a sua tabela de roteamento. No início de seu funcionamento os nós trocam informações com seus vizinhos sobre suas tabelas de roteamento. Entretanto, a partir deste momento, a troca de informações de roteamento só ocorre quando há uma mudança no custo de um enlace e, mesmo assim, os dados transmitidos se referirem somente ao enlace que teve seu custo modificado.

Uma outra classificação comum dos algoritmos de roteamento é a diferenciação entre protocolos pró-ativos ou reativos. Nos protocolos pró-ativos cada nó possui, em sua tabela de roteamento, informações referentes a possíveis destinos. Estas informações são coletadas durante o funcionamento normal da rede. Assim, quando for necessário enviar um pacote a um determinado nó de destino, a rota já é conhecida e pode ser imediatamente utilizada. Os protocolos reativos, em contrapartida, atuam sob demanda: a determinação de uma rota para um destino só é feita quando é enviada uma mensagem para esse nó.

2.1.2 Roteamento em redes ad hoc

Um dos primeiros protocolos de roteamento desenvolvidos para redes ad hoc foi o OLSR (*Optimized Link State Routing*) [16]. O OLSR é um protocolo pró-ativo sendo uma otimização do *Link State* para redes fixas. Entretanto, quando a topologia da rede não é estável, este protocolo apresenta um baixo desempenho. Isso ocorre devido à disseminação de informações da topologia, que necessita ser feita frequentemente causando um *overhead* de comunicação.

No DSDV (*Highly Dynamic Destination-Sequenced Distance-Vector Routing*) [17], cada nó possui uma tabela com todos os possíveis destinos dentro da rede e o número de saltos até cada um deles. Cada entrada nessa tabela possui o número de sequência determinado pelo nó de destino. Com esse número é possível diferenciar rotas antigas de rotas novas evitando a formação de *loops*. A rota utilizada é sempre a rota com o número de sequência mais recente. Caso duas rotas possuam o mesmo número de sequência, é utilizada a rota com menor número de saltos (menor métrica).

A utilização do DSDV em redes sem fio pode ser problemática. Isso porque, as mensagens periódicas necessárias para a manutenção das rotas não somente congestionam a rede como também podem acabar com o limitado suprimento de energia do nó.

O DSR (*Dynamic Source Routing Protocol*) [18], por outro lado, não envia mensagens periódicas para a manutenção das rotas. Nele, o nó origem determina toda a seqüência dos nós por onde passará o pacote. Assim, não é necessário o envio de mensagens periódicas para trocar informações de roteamento. Isso reflete em economia de banda e, conseqüentemente, de energia. O problema em utilizar esse algoritmo em RSSFs é que ele considera que o diâmetro da rede estará entre 5 a 10 nós. Entretanto, em uma rede com milhares de nós, isso seria praticamente impossível. Outro problema de utilizar esse algoritmo é que a determinação de toda a rota pelo nó origem exige a utilização de *flooding* para encontrar o caminho até o destino e isso, em uma RSSF com milhares de nós, seria inviável.

O AODV (*Ad-Hoc On-Demand Distance Vector*) [19] é um exemplo de algoritmo reativo. Somente quando uma rota é necessária que ela é criada. Assim, ele não mantém todas as rotas continuamente preocupando-se apenas com as rotas que estão em uso em um dado momento. Quando há a necessidade de uma nova rota, é disseminada pela rede uma requisição de rota que é repassada de nó em nó. Quando o destino ou algum nó que possui a rota para o destino é contactado, ele responde à requisição da rota. Entretanto, como a maioria dos algoritmos propostos para redes ad hoc, o AODV não considera as severas restrições computacionais, de banda e de energia das RSSFs.

A maioria do protocolos de roteamento propostos para redes ad hoc foram desenvolvidos para funcionar em redes que requisitam dados a um nó específico e sem muita redundância de dados e nós.

2.1.3 Roteamento em RSSFs

Os protocolos de roteamento para RSSFs têm que ser projetados para serem tolerantes a falhas de nós individuais enquanto minimizam o consumo de energia. Como a banda é limitada e tem de ser dividida entre todos os sensores, os protocolos de roteamento têm de ser capazes, usando colaborações locais, de reduzir os requerimentos por banda [1]. Além disso, os ambientes em que os nós necessitam operar, são muito dinâmicos com mudanças rápidas dos parâmetros físicos. Alguns dos parâmetros que podem variar dependendo da aplicação são:

- Energia disponível;
- Posição (se os nós são móveis);

- Alcance de transmissão dos sensores;
- Tarefas a serem executadas.

Os protocolos de roteamento desenvolvidos até o momento para redes cabeadas e redes ad hoc não são apropriados para RSSFs pelas seguintes razões:

1. Em geral, as RSSFs são centradas em dados, ou seja, ao contrário das redes sem fio tradicionais onde os dados são requisitados a um nó específico, em RSSFs os dados são requisitados baseados em certos atributos [20].
2. Em contraste com as redes ad hoc, em redes de sensores quase todas as aplicações possuem um fluxo de dados de múltiplas fontes para um nó sorvedouro particular [8].
3. Os nós sensores possuem sérias restrições de energia e processamento [21].
4. Os requisitos para as redes de sensores mudam de acordo com a aplicação. Pode existir uma aplicação que exija que os dados sejam enviados para a ERB o mais rápido possível, não importando a quantidade de energia gasta (aplicações militares), enquanto que, outra aplicação, pode perfeitamente suportar o envio periódico de dados (medição da temperatura de uma cidade com o objetivo de armazenar a temperatura média da cidade ao longo do ano).
5. Cada bit transmitido pode consumir tanta energia quanto a execução de dezenas ou centenas de instruções [22]. Assim, as vezes é melhor processar o dado localmente antes de transmití-lo.
6. Como os dados coletados pelos sensores são baseados em um fenômeno comum, é possível que parte dos dados possuam alguma redundância. Assim, ao invés de enviar os dados de cada nó separadamente, pode ser interessante que dados similares sejam agregados diminuindo o número de bits a serem enviados pela rede [21] [23].
7. Durante o funcionamento da rede os sensores podem mudar o seu papel ou função. Assim, por exemplo, em uma rede hierárquica um sensor pode ser líder de grupo em um determinado momento e em outro momento ele pode voltar a ser um sensor comum.

Pode-se concluir, com base nos seis itens acima, que as redes de sensores necessitam de protocolos de roteamento que sejam específicos para as aplicações, centradas em dados, capazes de agregar dados e otimizar o consumo de energia. Além disso, é importante observar que as RSSFs podem ter comportamentos completamente diferentes apenas mudando o tipo de dados que elas se propõem a coletar. Desta forma o projeto de protocolos para RSSFs deve considerar o tipo de dados que ela coleta e como eles serão processados e transmitidos.

Nas próximas subseções faremos uma descrição de alguns dos principais algoritmos de roteamento propostos para RSSFs.

2.1.3.1 Difusão direcionada

Intanagonwivat et. al [24] apresentam um algoritmo de roteamento centrado em dados para RSSFs chamado Difusão Direcionada. Nesse algoritmo a seleção das rotas é baseada em informações providas pela aplicação para a rede através do nó de monitoração. Somente os nós que possuírem os interesses requisitados retornam o dado desejado. Isto faz com que se gaste menos energia, pois bons caminhos são selecionados empiricamente.

Segundo Willing, Karl, e Wolisz [25] a difusão direcionada confia excessivamente na difusão de informações pela rede. Conseqüentemente, o desempenho final do protocolo pode ser influenciado fortemente pela eficiência ou não dessa operação elementar. Uma das maneiras mais diretas e simples de realizar uma transmissão difusão é a simples utilização da inundação onde cada nó na rede envia cada mensagem nova a todos seus vizinhos. Mas esta aproximação é ineficiente nas redes de sensores sem fio onde a semelhança significativa entre os nós vizinhos de dois sensores adjacentes pode conduzir a um grande número de re-transmissões desnecessárias. A difusão de transmissões de vizinhos que monitoram o mesmo evento resulta em disputas adicionais do canal de comunicação e no desperdício da largura de banda, causando, conseqüentemente, um aumento do consumo de energia dos nós. Em [26], é discutido o problema de identificar o conjunto de nós que deverão reenviar uma mensagem de forma que a cobertura da rede seja mantida e o número de difusões seja diminuído.

2.1.3.2 Multi

Em [27] foi proposto o Multi. O Multi é um algoritmo adaptativo e híbrido de disseminação de dados pela rede que adapta seu funcionamento de forma autônoma conforme variações da

rede. O Multi é baseado em em dois algoritmos de disseminação: O SID (Source-Initiated Dissemination), que é um algoritmo reativo onde o processo de disseminação se inicia a partir da origem dos dados, e EFTree (Earliest-First Tree), que é um algoritmo pró-ativo que constrói e mantém uma árvore para a disseminação de dados de toda a rede.

2.1.3.3 PROC

Em [28] é proposto o PROC, um protocolo cujo objetivo é a disseminação contínua de dados permitindo que a aplicação reconfigure o PROC em tempo de execução. Através de simulações é demonstrado que o PROC aumenta o tempo de vida da rede em até 12% e, ao mesmo tempo, apresenta uma degradação suave no desempenho quando o número de nós da rede aumenta.

2.1.3.4 Diffuse

Em [29] é argumentado que as limitações do alcance dos rádios dos nós e a necessidade de economia de energia em RSSF faz com que os dados dessas redes sejam normalmente transmitidos nó a nó, em uma topologia de árvore, até chegar à ERB (multi-hop). Assim, Nakamura et al. propuseram o Diffuse. O Diffuse permite a detecção do momento em que a topologia por onde os dados são disseminados deve ser reconstruída, baseando-se somente na medição do tráfego. É demonstrado em [29] que a abordagem apresentada evita as reconstruções desnecessárias de rotas.

2.1.3.5 TEDD

Em [30] é apresentado o TEDD (Trajectory and Energy-Based Data Dissemination). O TEDD é um novo algoritmo de disseminação de dados cuja idéia principal é a utilização de uma combinação de conceitos de disseminação de dados com informações fornecidas pelo mapa de energia. O funcionamento do TEDD é dividido em duas partes principais. Na primeira parte um algoritmo gera uma trajetória para os dados evitando os nós com pouca energia. Na segunda parte, um novo mecanismo de envio de pacotes repassa os pacotes na direção da ERB. É demonstrado nesse trabalho, através de simulações, que o gasto de energia com disseminação de dados pode ser concentrado somente nos nós com mais energia enquanto que os nós com pouca energia utilizam sua energia somente para sensoriar o ambiente.

2.1.3.6 LEACH (Low-Energy Adaptive Clustering Hierarechy)

A proposta que iremos apresentar no capítulo 3 deste trabalho foi inspirada, principalmente, em um algoritmo hierárquico proposto para RSSFs: o LEACH [1]. Como faremos a comparação de nossos resultados com os resultados do LEACH, faremos uma descrição mais detalhada desse algoritmo.

Heinzelman [2] apresenta um algoritmo de roteamento hierárquico chamado LEACH. O LEACH é um protocolo auto-organizável e adaptativo de agrupamento que utiliza randomização dos nós líderes para distribuir a energia gasta nos sensores da rede.

Conforme podemos observar na figura 2.1, no LEACH os nós se organizam em grupos locais com um nó agindo como uma ERB local ou líder de grupo. Se os líderes forem escolhidos a priori e fixados ao longo do tempo de vida da rede, como um algoritmo hierárquico tradicional, os nós sensores que forem escolhidos para serem líderes irão esgotar sua energia mais rapidamente, pois o nó líder gasta mais energia do que os nós sensores não líderes já que ele envia os dados do grupo para a estação base (que geralmente está distante da maioria dos nós da rede). Por isso, o LEACH altera aleatoriamente os líderes a fim de não acabar com a energia de um único nó na rede. O gasto de energia será, então, melhor distribuído entre os nós e isso acabará por atrasar o aparecimento de áreas descobertas na rede. O LEACH faz também a agregação de dados similares visando diminuir o número de dados enviados à ERB.

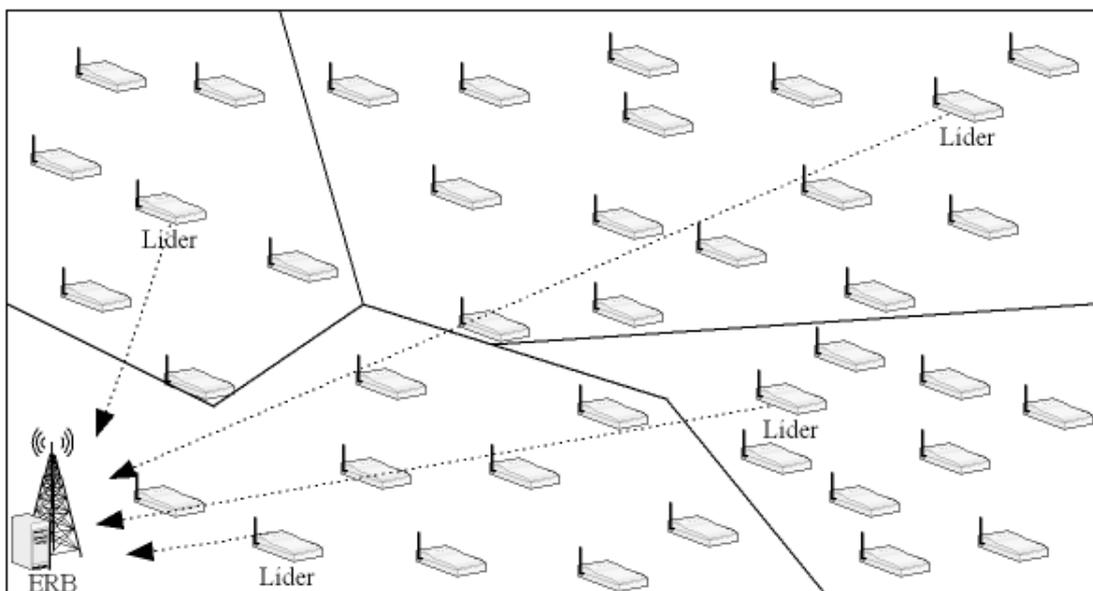


Figura 2.1: Exemplo de agrupamento no LEACH

Um sensor se elege líder de um grupo local com uma certa probabilidade. Estes líderes enviam uma mensagem para todos os outros sensores da rede informando que eles são possíveis líderes. Cada nó decide, então, qual líder ele deseja seguir escolhendo pertencer àquele grupo em que o consumo de energia na comunicação com o líder será menor [1].

Depois que todos os nós já estão organizados em grupos, cada líder cria uma agenda para cada nó em seu grupo. Isto permite que todos os nós que não sejam líderes tenham os seus componentes de rádio desligados durante todo o tempo, sendo religados somente durante a transmissão, minimizando assim a energia dissipada individualmente em cada nó. Quando o líder possui todos os dados dos nós no seu grupo, ele agrega e comprime estes dados e os envia à ERB.

Para distribuir melhor o gasto de energia entre os vários nós, os líderes não são fixos. Eles são escolhidos randomicamente em intervalos regulares. Assim, um conjunto S de nós podem ser eleitos líderes no tempo t_1 mas no tempo $t_1 + d$ um novo conjunto de nós será eleito. Cada nó toma a decisão de se tornar líder independente dos outros nós na rede e nenhuma negociação extra é feita para determinar quais nós serão os líderes da rede.

O sistema pode, a priori, determinar o número ótimo de grupos da rede. Isso irá, entretanto, depender de diversos parâmetros como a topologia da rede e os custos relativos da comunicação versus computação. Testes realizados em [1] mostram que para uma rede de 100 nós distribuídos aleatoriamente e dimensão $100m \times 100m$, o melhor número de líderes é um valor equivalente a 5% do número de nós da rede.

Um dos problemas do LEACH é que, como os líderes enviam os dados de seu grupo diretamente para a ERB, os líderes que estão mais distantes gastam mais energia no envio destes dados à ERB, o que diminuirá o seu tempo de vida.

2.1.3.7 LEACH-C

O LEACH-C [2] é uma variação do LEACH [1] que centraliza as decisões de formação dos grupos. A maior vantagem da abordagem centralizada é a criação mais eficiente de grupos distribuindo melhor os mesmos pela rede. Cada nó, na fase de inicialização da rede, envia sua posição geográfica e energia disponível para a ERB. Baseada nesta informação a ERB determina os grupos de forma centralizada. Quando os grupos e seus líderes são determinados, a ERB envia uma mensagem que contém o identificador do líder que cada nó deve se associar.

Após isso os nós agem como se utilizassem o LEACH original.

2.1.3.8 PEGASIS

Lindsey e Raghavendra propuseram em [23] o PEGASIS, um protocolo de roteamento baseado no conceito de correntes. Nessa proposta cada nó troca informações apenas com os vizinhos mais próximos formando uma corrente entre os nós e apenas um nó é escolhido a cada momento para transferir as informações coletadas ao nó sorvedouro. Os dados são repassados pela corrente com a utilização de um *token*. Assim, um nó só transmite os dados quando ele estiver com o *token*. Portanto, o número de trocas de mensagens será baixo e a comunicação será realizada entre nós próximos uns dos outros. Espera-se, com isso, que a energia gasta seja menor, se comparada a outros protocolos que requerem muitas trocas de mensagens para eleger líderes e formar grupos e protocolos em que os nós constantemente trocam mensagens com o sorvedouro de forma direta (assume-se que o sorvedouro geralmente encontra-se distante de todos os nós). Isto implica em um tempo de vida maior para cada nó e um consumo menor da largura de banda da rede. A figura 2.2 ilustra um exemplo de como é feita a comunicação no PEGASIS.

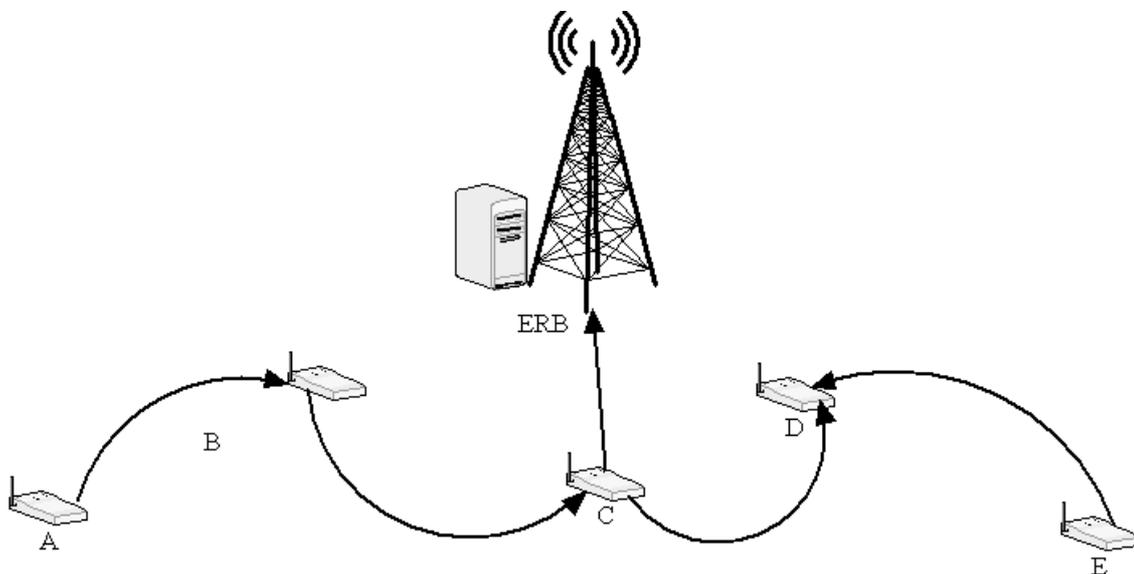


Figura 2.2: Princípio do funcionamento do PEGASIS

Apesar desse algoritmo conseguir economizar energia encaminhando os pacotes para os nós mais próximos ao invés de encaminhá-los diretamente para a ERB, existe o problema do atraso na transmissão dos dados gerado pela formação da corrente e pela passagem do *token*

pela rede. Embora o *token* evite a colisão na transmissão de dados entre os nós, ele acaba aumentando o atraso no recebimento dos dados pela ERB. Outro problema desse protocolo é que, durante um *round*, somente um nó envia os dados de toda a rede para o nó sorvedouro. Dependendo do tamanho da rede, isto fará com que este nó fique sem energia rapidamente.

2.2 Qualidade de Serviço

A Internet provê, atualmente, uma grande diversidade de serviços. Para prover aplicações de tempo real ou para prover uma qualidade de serviço melhor do que o tradicional melhor esforço, é necessário o suporte a Qualidade de Serviço.

QoS pode ser definida de várias maneiras [31]. Na camada de aplicação, QoS pode ser definida como a qualidade percebida pelo usuário/aplicação enquanto que na camada de rede QoS pode ser definida como a qualidade do serviço oferecida às aplicações/usuários [31]. A RFC 2836 [32] define QoS como um conjunto de exigências a serem atendidas quando se transporta um pacote da origem ao destino. Em [33] QoS é definida como a quantidade ótima de nós transmitindo dados de forma a maximizar o tempo de vida da rede sem que ocorra perda na resolução dos dados. Em [34] roteamento com qualidade de serviço diz respeito à determinação e à manutenção de rotas que satisfazem algumas restrições de QoS.

2.2.1 QoS em redes cabeadas

Muitos algoritmos que provêm QoS na Internet foram propostos na última década. As propostas da IETF são o Intserv [35] e o DiffServ [36]. Estas propostas oferecem a possibilidade de prover novos serviços com garantia de atraso, perda e priorização de pacotes. O IntServ provê Qualidade de Serviço através da reserva de recursos. O Diffserv provê QoS através da diferenciação de classes de serviços e garante os recursos mínimos para cada classe de serviço.

A utilização de um algoritmo como o Intserv em RSSFs é, do nosso ponto de vista, inviável pois os recursos das redes de sensores já são bastante restritos para serem reservados. O Diffserv, por sua vez, normalmente modifica a forma de escalonar os pacotes a serem transmitidos. Com isso, pode ser necessário que as mensagens de menor prioridade sejam armazenadas para que as mensagens de maior prioridade sejam transmitidas. Entretanto, o hardware atualmente disponível para RSSFs possui pouca memória disponível. Assim, dificilmente seria possível

armazenar as mensagens de menor prioridade. Com isso podemos concluir que fornecer garantias rígidas de recursos para determinadas classes de serviços é incompatível com o atual estado da arte das RSSFs.

As redes *overlay* estão propostas como uma melhoria do nível de QoS da rede. As redes *overlay* [37], [38], [39] e [40], constroem uma topologia virtual sobre a topologia física da rede. Entretanto, a função primária da rede *overlay*, difere muito de aplicação para aplicação. O desenvolvimento de sistemas que utilizam redes *overlay* para prover novas funcionalidades para a rede é motivado pela dificuldade de se implementar as políticas de QoS em todos os nós da rede. Com isso, muitas aplicações *overlay* foram desenvolvidas para redes cabeadas [9] [10] [11] [41].

Em [9] foi proposta a técnica RON (*Resilient Overlay Networks*) com o objetivo de aumentar a robustez e a disponibilidade das rotas na Internet. Conseqüentemente, aumenta-se também a confiança na transmissão dos dados. Quando um nó da rede começa a falhar, a rede *overlay* provê a estrutura necessária para que essa falha seja contornada. A técnica RON propõe formas de fazer com que as aplicações cooperem entre si para aumentar a confiabilidade e o desempenho da rede. O RON monitora os nós da rede e posteriormente decide se os pacotes devem ser encaminhados diretamente para a Internet ou indiretamente via outro nó *overlay*, melhorando assim as métricas da rede. Com isso a técnica RON consegue prover um roteamento com um melhor desempenho do que o tradicional melhor esforço pois permite que um pequeno grupo de nós na Internet detectem rotas "quebradas" ou com desempenho degradado.

Foram propostos ainda, alguns algoritmos que provêm QoS para comunicações multicast na Internet. Segundo Shavitt, Chen e Nahrstedt [42] os algoritmos que tradicionalmente propõem QoS em transmissões multicast são computacionalmente "pesados" [43], [44] ou impõem um custo de comunicação para conectar um novo membro na árvore de transmissão [44].

Em [45] é proposta uma arquitetura onde, através de uma rede *overlay*, obtêm-se métricas de qualidade dos enlaces da rede. Com isso propõe-se o ALSW (*Application Layer Switching*), um algoritmo que tem a habilidade de utilizar rotas alternativas para prover uma melhor Qualidade de Serviço. O ALSW é capaz de tomar decisões baseadas em várias métricas de QoS como o menor caminho, atraso e vazão. Assim, demonstra-se que o simples re-roteamento do tráfego para enlaces sub-utilizados já melhora a Qualidade de Serviço. Propõe-se também

em [45] o ALST (*Application Layer Striping*) que, além de possuir todas as habilidades do ALSW, possui a capacidade de utilizar múltiplos caminhos alternativos. Tanto no ALSW quanto no ALST ocorre uma monitoração da rede e periodicamente cada nó atualiza, através da troca de mensagens, o custo de todos os enlaces da rede. Através da análise do custo desses enlaces, consegue-se obter o caminho ótimo. Entretanto, esses protocolos necessitam de mensagens adicionais para repassar para todos os nós *overlay* uma tabela com os custos de todos os enlaces da rede.

2.2.2 QoS em redes sem fio

A tendência é que os usuários de comunicações sem fio irão demandar da mesma qualidade de serviço disponível correntemente nas redes cabeadas. Existem vários algoritmos que provêm QoS para redes ad hoc.

Em [3] e [46] são propostas redes sem fio sobre redes celulares. O objetivo é aproveitar as vantagens das redes celulares e das redes ad hoc. Dentre as vantagens pretendidas com a utilização desta estrutura encontram-se:

- melhora da robustez e da escalabilidade do sistema;
- balanceamento de carga;
- extensão da cobertura de uma célula da rede celular;
- prover uma maior conectividade usando uma comunicação multi-hop;
- reduzir o consumo de energia do nó;
- aumentar a capacidade da rede.

Em [3] dois nós podem se comunicar diretamente, sem que os dados passem pela ERB. A figura 2.3 descreve melhor esta proposta. Nela, a ERB pode dividir o envio dos dados para um nó C, enviando parte do tráfego diretamente para o nó C e a outra parte através os nós A e B. Esses nós A e B entrariam, então, em contato com o nó C, repassando os dados enviados pela ERB. Como a comunicação entre os nós seria feita em frequência diferente da comunicação com a ERB, estas comunicações não sofreriam interferência. Porém, da forma como foi feita a proposta, seria necessário que o nó possuísse dois canais de rádio diferentes.

Um para se comunicar com a ERB e outro para se comunicar com outro nó. Isso tornaria o hardware mais caro e, conseqüentemente, dificultaria a utilização desse algoritmo em uma rede com milhares de nós.

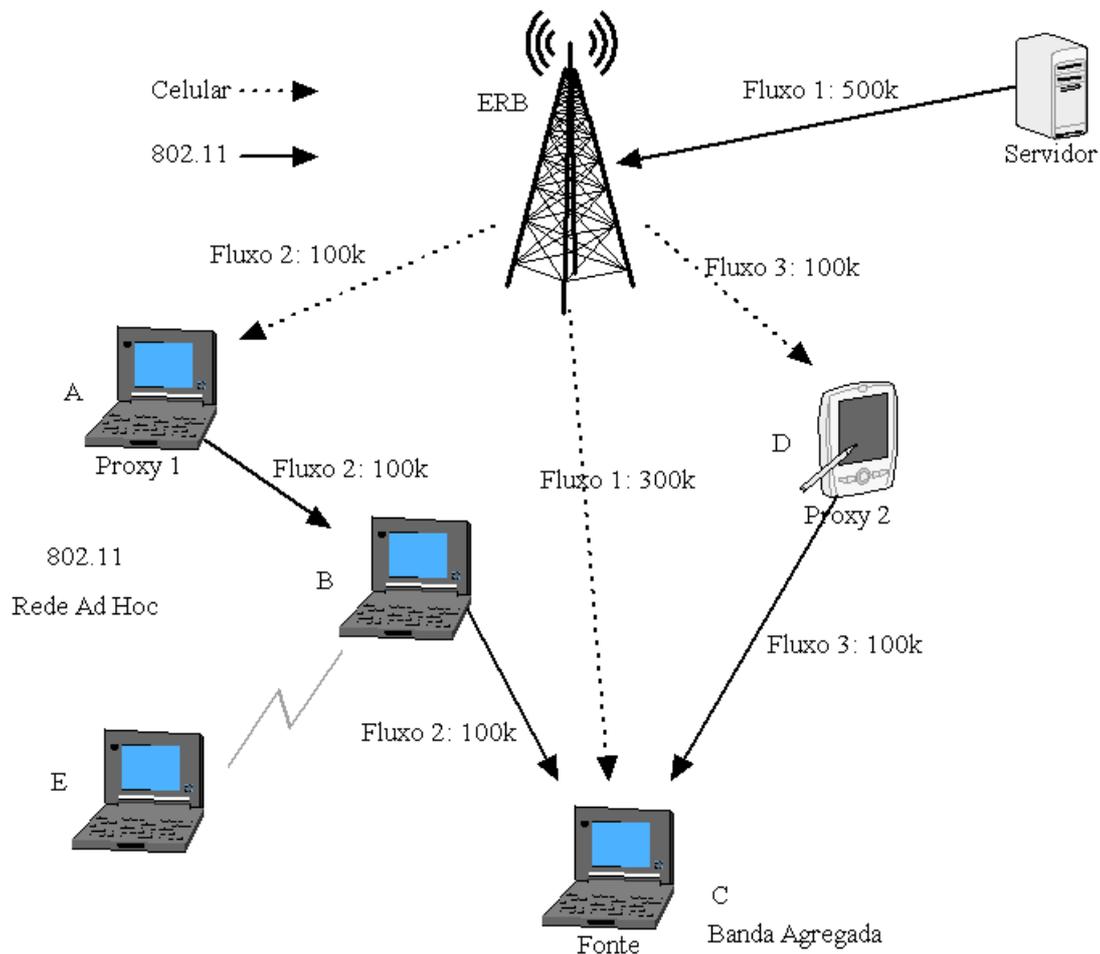


Figura 2.3: Exemplo do funcionamento do protocolo proposto em [3]

Em [47] os nós móveis se comunicam com outros nós móveis. Na arquitetura proposta existem três níveis de hierarquia: nós terminais, nós intermediários e nós de egresso. São os nós terminais que possuem requisitos especiais de QoS. Os nós intermediários apenas repassam aos nós de egresso os dados enviados pelos nós terminais. Nós de um mesmo nível na hierarquia não se comunicam. Os pontos de egresso se comunicam com os nós intermediários e os nós intermediários não possuem restrições de banda sendo que eles normalmente ficam próximos da ERB. Cada nó descobre os parâmetros de QoS na comunicação com seus vizinhos no nível superior da hierarquia. Assim, consegue-se repassar os pacotes pelo enlace que garante a melhor qualidade de serviço na visão local do nó. A grande desvantagem dessa abordagem é

que, como é tomada uma decisão local, pode-se acabar enviando o pacote por um caminho que, apesar de parecer melhor, não atende aos requisitos de QoS. Isso poderia ser melhorado se o nó egresso também auxiliasse na determinação dos níveis de QoS dos vizinhos em um nível superior.

Os trabalhos citados anteriormente que tentam prover QoS, por terem sido desenvolvidos para redes cabeadas ou para redes ad hoc, não levam em consideração as severas restrições das RSSFs.

2.2.3 QoS em RSSFs

Muito trabalho tem sido feito em diversas áreas de RSSFs visando o desenvolvimento de novos protocolos, economia de energia e localização dos nós sensores. Entretanto, prover Qualidade de Serviço em RSSFs é uma área de pesquisa desafiadora [34] e ainda pouco explorada. Isso ocorre em grande parte porque os princípios básicos das RSSFs são diferentes das redes convencionais [31].

Existem algumas propostas de algoritmos de roteamento com Qualidade de Serviço para RSSFs [48], [49], [50], [51]. Entretanto, as propostas existentes para RSSFs são dispendiosas do ponto de vista computacional [48], precisam de nós sensores mais poderosos e do envio de mensagens especiais para a determinação dos parâmetros de QoS dos enlaces [49], estão mais preocupadas com a qualidade fornecida pelo meio de acesso [50], ou ainda, se preocupam mais com a qualidade de serviço no nível da aplicação [51].

Entre os desafios de se prover QoS em RSSFs destacam-se os seguintes:

1. topologia dinâmica das redes de sensores [52];
2. grande quantidade de sensores [52];
3. natureza não confiável dos nós sensores. Estudos recentes demonstram que 20% dos nós em uma comunicação sem fio falham [53];
4. capacidade limitada de processamento dos nós;
5. alto índice de "quebra de rotas" devido à falhas dos nós [53] [54];
6. grandes restrições computacionais e de energia [53] [54].

Apesar de existirem diversas propostas visando a garantia da qualidade de serviço em redes convencionais de computadores e para redes ad hoc, elas não se adaptam ao ambiente de RSSF pois a maior parte destas propostas se baseiam em reserva de recursos [52] [55] [56] [57].

O primeiro algoritmo a levar em consideração o provimento de QoS em RSSFs foi o SAR (*Sequential Assignment Routing*) [48]. A decisão de roteamento no SAR é baseada em três fatores: recursos de energia, QoS e a prioridade de cada pacote. Para evitar que a falha de um único nó comprometa a entrega dos pacotes, é utilizada uma árvore com múltiplas rotas. Caso haja falha em algum nó, é feita uma restauração local da rota. Entretanto, é montada uma árvore para cada parâmetro de QoS da aplicação. Toda vez que algum nó falha, é necessário que toda a árvore seja reconstruída. Embora a existência de uma árvore para cada parâmetro de QoS e a reconstrução local das rotas garanta a tolerância à falhas, existe o problema do *overhead* da manutenção de todas as árvores [58]. Tentar prover um nível fixo de QoS é difícil devido às limitações atuais dos sensores.

Em [49] foi apresentada uma proposta visando-se a utilização de redes *overlay* para se prover QoS em RSSFs. Nessa proposta a rede *overlay* deve ser formada sobre uma rede hierárquica e os nós líderes devem ser mais poderosos e definidos na inicialização da rede formando uma rede *overlay* entre eles. Além disso, os nós *overlay* devem possuir dois rádios distintos. De tempos em tempos, os nós líderes enviam um pacote *overlay* para a ERB. É através desse pacote que são coletadas as informações relativas à qualidade de serviço de um enlace. Essas informações são, então, disseminadas aos líderes da rede. A técnica que será proposta no capítulo 5 não se aplica unicamente a redes hierárquicas e não tem a necessidade da existência de nós mais poderosos na rede. Além disso, as métricas de QoS são coletadas através de dados fornecidos pelos pacotes não prioritários não havendo a necessidade do tráfego na rede de pacotes cujo único objetivo é a coleta de dados de QoS nos enlaces da rede.

O GridKit [51] é um arcabouço flexível que suporta diversas aplicações em um grid. Esse arcabouço pode funcionar em redes de sensores. Entretanto, esse trabalho é focado em fornecer diversos serviços para a aplicação como, por exemplo, segurança, descobrimento e gerenciamento de recursos. A proposta que será apresentada no capítulo 5 difere deste trabalho porque nosso foco está no roteamento dos pacotes e nossa principal preocupação é encontrar a melhor rota até o nó sorvedouro que atenda aos parâmetros de QoS da aplicação.

Boutaba et al. [50] propuseram a utilização de uma rede de sensores, organizada em uma rede *overlay*, para prover um melhor acesso a uma rede ad hoc sem fio. Para isso, é proposta uma arquitetura de duas camadas onde a rede de sensores atua como uma camada de controle de uma rede padrão IEEE 802.11. A proposta que será apresentada no capítulo 5 difere desse trabalho já que a preocupação em [50] é em permitir um acesso justo ao meio não existindo uma preocupação específica com QoS.

Em [59] é apresentado um estudo do desempenho de algoritmos de roteamento para disseminação contínua de dados na rede mediante a ocorrência de falhas. As maiores causas das falhas dos nós são apresentadas e divididas em categorias de acordo com sua extensão e persistência. Com isso, foi possível extrair aspectos comuns às diversas falhas, simplificando, assim, a avaliação dos protocolos. Este trabalho se preocupa com o estudo do desempenho dos algoritmos de roteamento em cenários em que há falhas na rede. Não há, neste trabalho, uma preocupação em contornar essas falhas.

2.3 Contribuições desse trabalho

No capítulo 3 desse trabalho apresentamos o Inter Cluster Routing Algorithm (ICA). O ICA é um novo e eficiente algoritmo de roteamento para RSSFs, baseado no Low-Energy Adaptive Clustering Hierarchy (LEACH) [1]. O ICA foi desenvolvido para redes de sensores que coletam informações periodicamente durante o tempo de vida da rede. Para esses ambientes a longevidade da rede é normalmente um fator de extrema importância. Os resultados de simulações mostram que o ICA quando comparado ao LEACH e ao LEACH-C [2] apresenta não somente um tempo maior de vida da rede como também um maior número de pacotes transmitidos e uma distribuição mais uniforme da energia na rede.

No capítulo 5 apresentamos a técnica OWL que provê QoS em RSSFs através da utilização de redes *overlay*. No nosso trabalho, focaremos em uma melhora da Qualidade de Serviço do roteamento observando os parâmetros de QoS fornecidos pela aplicação. O objetivo de se utilizar redes *overlay* em redes de sensores é prover um acesso sem fio mais flexível [47].

Neste trabalho iremos nos concentrar em aplicações que não exigem um nível fixo de QoS, mas sim que possam desempenhar suas funções mesmo que a rede não possa garantir o nível desejado de QoS. Assim, a técnica OWL tem como objetivo atender ao nível de

QoS requisitado da melhor maneira possível. Para tanto, definimos dois tipos de pacotes: pacotes prioritários e pacotes não prioritários. Pacotes não prioritários são pacotes que, caso sejam perdidos, não ocasionariam um grande transtorno nem para a aplicação nem para o funcionamento da rede. Pacotes prioritários são pacotes que necessitam de algum nível de QoS.

No restante deste texto, deve-se entender por roteamento com qualidade de serviço o problema de selecionar um caminho para os pacotes prioritários, melhor que o caminho padrão. Essa escolha deve ser baseada nos requerimentos de QoS de cada pacote. Para tanto, utilizamos os pacotes não prioritários para coletar métricas de QoS das rotas da rede. Com isso, seremos capazes de prover um melhor serviço para os pacotes prioritários. Assim, a quantidade de pacotes prioritários recebidos pela ERB acaba aumentando porque o tráfego transportado tende a aumentar. Uma vantagem da utilização de pacotes não prioritários na coleta de métricas da rede é que, com isso, não é necessário que nenhum pacote adicional para a coleta de dados trafegue pela rede.

Por fim, como as aplicações determinam a forma de funcionamento das RSSFs, é natural que a aplicação também aponte quais os seus requisitos de QoS e que isto seja utilizado no roteamento de mensagens dentro das RSSFs. Acreditamos também que as aplicações podem ter distintos modos de operação. Assim, o ideal é codificar a qualidade requerida no pacote pois desta forma uma mesma aplicação, em diferentes momentos, pode utilizar diferentes formas para rotear pacotes.

Capítulo 3

ICA - Inter Cluster Algorithm

Neste capítulo iremos apresentar o *Inter Cluster routing Algorithm* (ICA). Esse algoritmo é a proposta de um novo algoritmo de roteamento para RSSFs. Esse algoritmo apresenta uma distribuição mais homogênea do gasto de energia em RSSFs e um aumento do número de dados transmitidos até a ERB.

3.1 Introdução

Um dos objetivos das redes de sensores é o monitoramento eficiente e seguro em áreas em que a presença humana é de difícil acesso ou representa riscos à integridade física. Como a reposição de bateria em nós localizados nestas regiões é difícil, as redes de sensores devem visar a economia de energia gasta pela rede. Uma das formas de se fazer isso é analisar os dados sensoriados e fazer a agregação dos mesmos. Muitas vezes os dados de sensores geograficamente próximos são bastante semelhantes. Isto difere as RSSFs das redes tradicionais onde os dados requisitados por nós próximos geralmente não se correlacionam. Assim, é importante que o algoritmo de roteamento utilizado nas RSSFs possa utilizar a redundância de dados de sensores vizinhos como forma de economizar energia.

O ICA (*Inter Cluster Routing Algorithm*) é uma nova proposta de algoritmo de roteamento que visa minimizar os gastos de energia em RSSFs. O ICA é baseado no LEACH. O LEACH é um algoritmo hierárquico, sendo que, segundo Santi, Younis e Fahmy [60], as redes hierárquicas aumentam a escalabilidade e a eficiência no consumo de energia entre os nós. Isso porque, nessas redes, os dados de um grupo podem ser processados pelo líder do grupo,

e os dados dos nós de um grupo podem ser agregados de tal forma a manter a informação contida nos pacotes originais [2]. Assim, uma quantidade menor de dados é repassada ao líder mais próximo e, conseqüentemente, uma quantidade menor de dados é transmitida à ERB. Com a agregação de dados, reduzimos o consumo de energia em todas as etapas até a chegada dos dados na ERB. As redes hierárquicas apresentam um gasto adicional para a formação dos grupos. Acredita-se que este gasto adicional seja compensado pela maior eficiência do consumo de energia apresentada por essas redes.

O ICA foi idealizado para aumentar o tempo de vida da rede e, conseqüentemente, o número de pacotes recebidos pela ERB. Os nós são organizados em grupos com um nó do grupo atuando como líder. Todos os nós que não são líderes enviam seus dados para o líder do seu grupo. O nó líder do grupo recebe os dados de todos os nós de seu grupo, faz a agregação dos dados e retransmite para outro líder na direção da ERB ou na falta deste para a própria ERB. Portanto, os nós líderes normalmente consomem mais energia do que um nó não líder. Se o líder fosse fixo desde o início da operação da rede, ele provavelmente ficaria sem energia mais rapidamente e começariam a aparecer áreas sem monitoramento na rede, pois além dos dados de sensoriamento do próprio líder, a rede ficaria sem conhecer os dados sensorizados por todos os nós daquele grupo. Por isso, assim como o LEACH, o ICA faz um rodízio dos líderes de grupo. Para isso, o funcionamento do algoritmo é dividido em *rounds*. Cada *round* é dividido em duas fases: a fase de formação de grupos e a fase de coleta de dados.

No ICA, a comunicação dentro do grupo é feita usando o padrão TDMA. Como o número de nós esperado por líder é relativamente baixo, o líder é capaz de fazer um agendamento das comunicações evitando colisões. A comunicação entre dois líderes e entre os líderes e a ERB é feita usando o padrão CSMA. O CSMA foi escolhido pois não há um gerente global para os nós líderes e o alcance entre os nós líderes é limitado devido ao custo da comunicação. Sendo assim, a utilização do CSMA foi a maneira utilizada para organizar essa transmissão. Desta forma justificamos a utilização de dois padrões distintos pois um deles é mais indicado para a comunicação dentro do grupo e outro é mais indicado para a comunicação entre líderes de grupo.

No início do processo a estação rádio base envia uma difusão para todos os nós informando sua posição geográfica. Após isso os nós sabem a posição geográfica da ERB e as suas próprias posições. Os nós são, então, agrupados utilizando regras semelhantes às do LEACH.

Entretanto, no momento da formação dos grupos ocorre também a construção das rotas entre dois líderes. Esta é a fase de formação dos grupos. Após a formação dos grupos, inicia-se a fase Coleta de dados. Nessa fase, os líderes recolhem os dados de seu grupo e os repassam aos líderes mais próximos (ou para a ERB).

3.2 Fase de formação dos grupos

A fase de formação de grupos do ICA é baseada no LEACH [1]. Assim, um número predeterminado de nós são eleitos como possíveis líderes. A decisão de se tornar um líder é feita escolhendo-se um número aleatório entre 0 e 1. Se o número gerado for menor do que um limite $T(n)$ então o nó se tornará um líder no *round* corrente. O limite $T(n)$ é dado pela expressão [2]:

$$T(n) = \frac{k}{(1-k \times (r \bmod (\frac{N}{k})))}, \text{ se } n \in G$$

$$T(n) = 0 \text{ caso contrário,}$$

onde K é a porcentagem desejada de líderes, r é o *round* corrente e G é o conjunto de nós que não foram líderes nos últimos $\frac{1}{k}$ *rounds*. Usando este limite, cada nó será líder em algum ponto dos $\frac{N}{k}$ *rounds*. Durante o *round* 0 ($r=0$), cada nó tem a probabilidade k de se tornar um líder de grupo. Os nós que são líderes em um *round* x , não podem ser líderes novamente nos próximos $\frac{N}{k}$ *rounds*. A probabilidade de um nó que ainda não foi líder se tornar líder de grupo vai aumentando, uma vez que existem menos nós que podem ser elegíveis. Depois de $\frac{N}{k} - 1$ *rounds*, $T=1$ para qualquer nó que ainda não se tornou líder ainda e depois de $\frac{N}{k}$ *rounds* todos os nós podem ser líderes de novo.

Os nós que não são líderes ligam seus receptores durante esta fase para escutar todas as mensagens. Após o nó se eleger líder para o *round* corrente ele faz um difusão com uma mensagem informando para toda a rede que ele é um nó líder. Isto é feito com o protocolo MAC CSMA [61]. Essa mensagem possui apenas o ID do nó e um cabeçalho informando que esta mensagem é uma mensagem de informação.

Cada nó que não é líder decide então a qual grupo ele se associará neste *round*. Para tomar esta decisão o nó analisa a potência dos sinais recebidos e escolhe pertencer ao grupo cujo líder enviou uma mensagem com o sinal mais forte. Isso é feito supondo que a propagação do sinal é simétrica o que implica que o caminho percorrido pelo sinal será o mesmo nos dois

sentidos. O nó escolhido será, normalmente, o nó mais próximo a não ser que exista algum obstáculo causando interferência na comunicação entre os dois nós próximos. Se isso ocorrer, a transmissão será feita com o nó mais distante mas que, em termos de potência do sinal, é o que resulta em uma economia de energia maior.

Quando um líder envia mensagens informando que ele é líder de grupo, ele fica escutando a rede. Ao receber o anúncio de outro líder que esteja na direção da ERB, e caso a potência do sinal recebido seja mais forte do que a potência do sinal de algum outro vizinho armazenado anteriormente (caso exista), ele armazena a frequência de comunicação do novo vizinho, sobrescrevendo a frequência armazenada anteriormente. Assim, os líderes sabem onde está o nó líder de sinal mais forte na direção da ERB. É através desse líder que os dados serão enviados para a ERB.

Quando um nó decide a qual grupo ele quer pertencer, ele informa ao líder que fará parte do grupo. Esta comunicação também é feita utilizando o protocolo CSMA. Essa mensagem contém o ID do nó, o ID do líder ao qual ele vai se associar e um cabeçalho informando que é uma mensagem de associação. A potência de transmissão do nó é, então, ajustada para que ele gaste a menor quantidade possível de energia na comunicação com o líder.

Os líderes de grupo repassam uma agenda TDMA a cada um dos nós de seu grupo informando o momento em que o nó pode enviar seus dados ao líder. Com isso, é assegurado que não ocorrerá nenhuma colisão entre os nós de um grupo. Assim, garante-se também que cada nó transmitirá seus dados pelo menos uma vez em cada *round*. O nó pode decidir enviar mais de uma mensagem ou não. A decisão dependerá se ele terá dados novos a serem enviados.

Podemos ver na figura 3.1 uma divisão de grupos em dois *rounds* consecutivos. Nesta figura, os nós em forma de triângulo são líderes de grupo e os outros nós são sensores comuns.

Um diagrama resumindo passo-a-passo como é a formação dos grupos está mostrado na figura 3.2.

É necessário ressaltar que a eleição aleatória de líderes é válida em casos em que todos os nós começam com a mesma quantidade de energia. Assim, tenta-se distribuir o gasto de energia fazendo-se a rotação de líderes. Se alguns nós possuem mais energia do que outros, eles podem ser pré-definidos como líderes do grupo.

A fase de formação de grupos descrita acima possui basicamente três diferenças em relação à fase de formação de grupos descrita no LEACH: A primeira é que nesta fase ocorre também

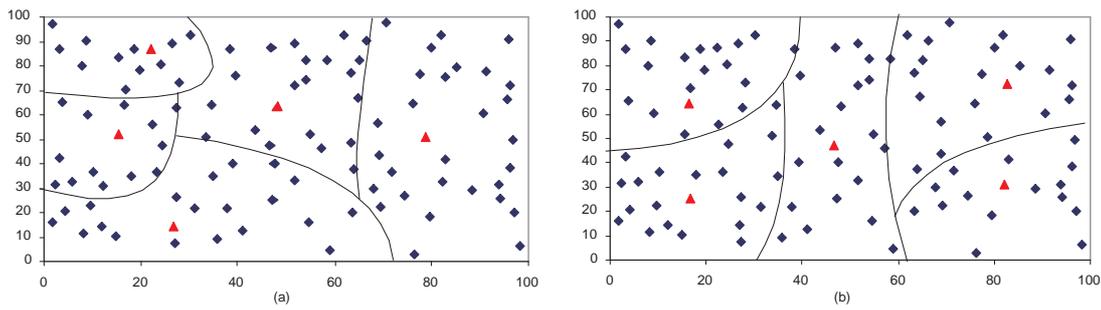


Figura 3.1: Imagem de dois *rounds* consecutivos. Em (a) temos a formação de grupos em um *round* e em (b) temos a formação de grupos no *round* seguinte

o estabelecimento das rotas entre os líderes de grupos. A rota para a ERB passa pelo vizinho mais próximo na direção da ERB e é por onde os nós líderes encaminharão seus dados na próxima fase. A segunda diferença foi a limitação do alcance dos nós implementada para prover uma maior escalabilidade do algoritmo. No LEACH, durante todo o funcionamento do algoritmo, o nó líder alcança toda a rede. Isto, mesmo sendo útil em alguns casos, pode causar problemas de escalabilidade. À medida que o número de nós da rede vai aumentando há uma necessidade natural de se aumentar o número de líderes da rede. Porém, caso os líderes alcancem toda a rede, o número de colisões entre as mensagens enviadas pelos líderes também aumenta. Em um determinado momento, as colisões são tantas que o LEACH não consegue formar nenhum grupo. Com isso, nenhum dado é entregue à ERB. Assim, no ICA o alcance dos nós após a fase de formação de grupos é limitada. A terceira diferença foi a adição de um atraso aleatório no anúncio de líder. Isso foi feito para evitar que vários líderes sejam anunciados ao mesmo tempo. Os nós que não conseguirem se associar a nenhum grupo irão se tornar líder e ajustar o alcance do seu rádio para se comunicar com o líder mais próximo.

Após a fase de formação dos grupos, os nós iniciam o envio de dados para a ERB. Esta fase foi chamada de fase de coleta de dados.

3.3 Fase de coleta de dados

No LEACH os líderes de grupo enviam as mensagens diretamente para a ERB. Com isso os nós mais distantes acabam consumindo mais energia e, conseqüentemente, ficam sem energia antes dos nós que estão próximos à ERB. No ICA tentamos minimizar esse problema transmitindo

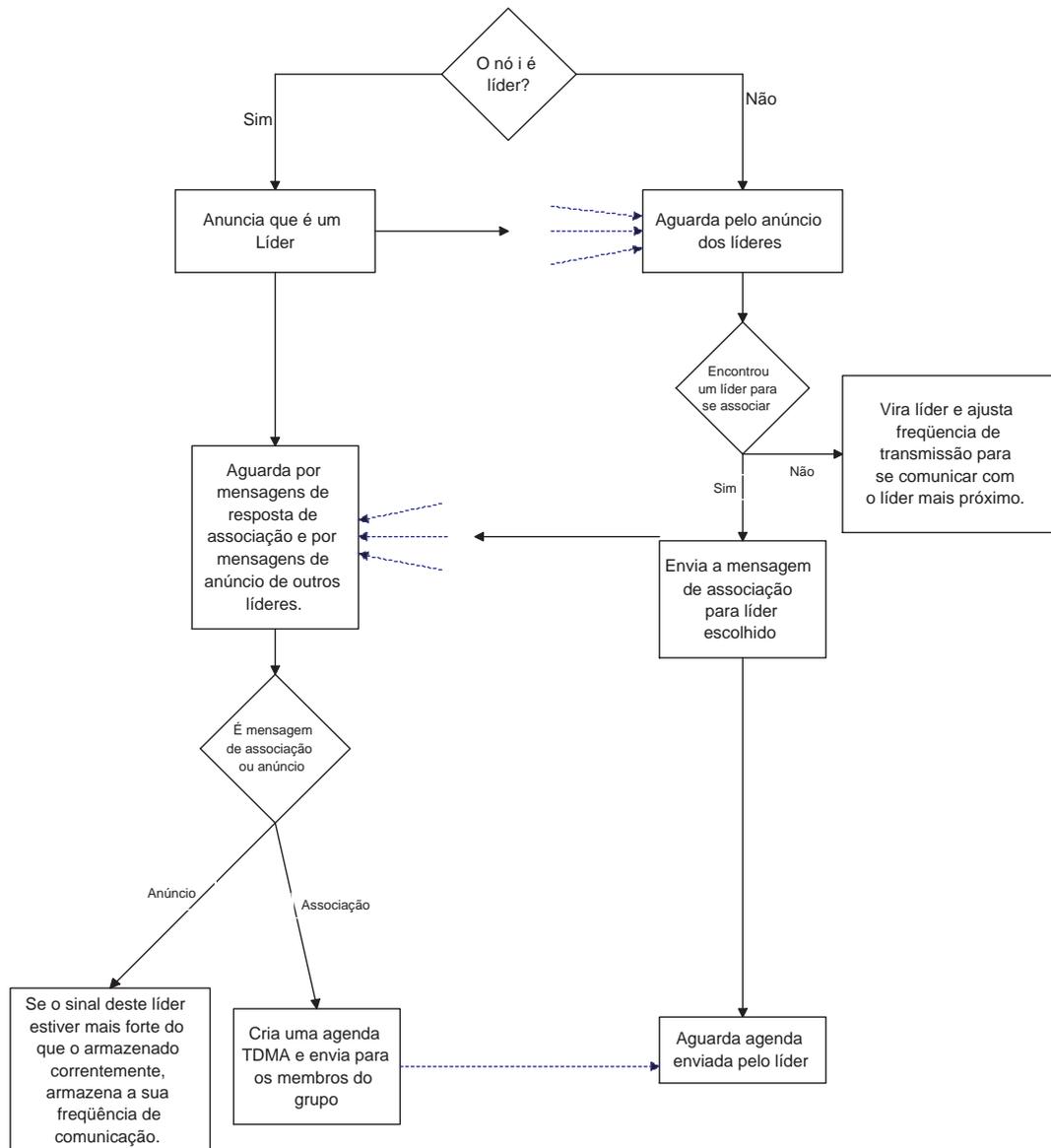


Figura 3.2: Fase de formação dos grupos

as mensagens para o líder mais próximo na direção da ERB.

Essa fase é dividida em *frames*. Um *frame* é o tempo em que todos os nós do grupo já enviaram dados (ou tiveram chance de enviar mas não enviaram porque não tinham dados novos) para o líder. Assim, o líder já pode encaminhar esses dados repassando-os a outro líder. Após o fim de um *frame* todos os nós do grupo podem, novamente, enviar mensagens para a ERB. Isto se repete até que o *round* termine.

Os nós líderes têm que manter seu rádio ligado durante todo o *round* enquanto que os nós que não são líderes só ligam o rádio no momento em que vão transmitir os dados. A figura 3.3 descreve a fase em que os dados dos nós são coletados.

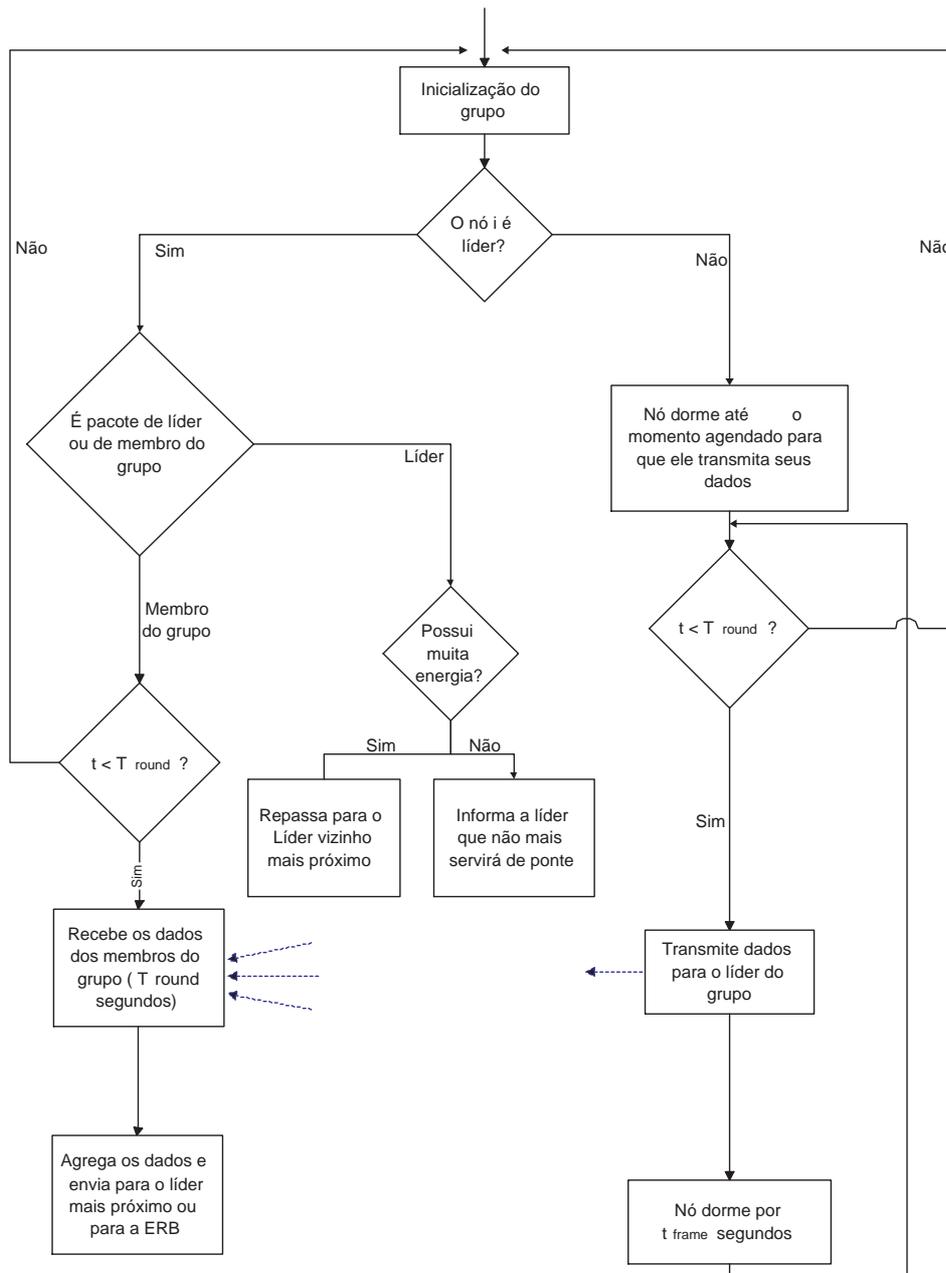


Figura 3.3: Fase de coleta de dados

Para reduzir o gasto de energia os nós que não são líderes ajustam a sua potência de transmissão de forma que seja gasto o mínimo de energia na comunicação com o líder do grupo.

Assim, após recolher os dados de todos os nós de seu grupo, o líder repassa os dados para que eles cheguem à estação base. A potência de transmissão é ajustada entre os líderes de forma a economizar energia e causar a menor interferência possível. Além disso, para reduzir a colisão é utilizada uma frequência de comunicação diferente para cada nó líder. Essa técnica

é utilizada também em [62] como forma de minimizar a colisão.

O ICA economiza energia enviando as mensagens ponto-a-ponto. Entretanto, é necessário que todos os líderes armazenem a frequência de comunicação com o líder mais próximo na direção da ERB. Assim, quando os líderes começam a se comunicar com os líderes mais próximos ao invés de se comunicar com a ERB, a energia despendida por cada nó da rede diminui aumentando assim a quantidade de energia total da rede. Além do modelo energético apresentado por Heinzelman em [2], outra importante suposição é que se aplicarmos algoritmos tradicionais de roteamento multi-hop em redes de sensores, os nós mais próximos da ERB irão esgotar sua energia antes que os outros nós. Isto deverá ocorrer uma vez que esses nós seriam responsáveis por re-enviar os dados de toda a rede para a ERB. Assim, os nós mais próximos da ERB gastarão mais energia que a média dos outros nós da rede. Contudo, nesse cenário, caso os nós fossem desativados, a rede inteira ficaria sem comunicação com a estação rádio base.

O LEACH resolve esse problema fazendo com que cada líder de grupo transmita suas mensagens diretamente à ERB com apenas uma transmissão de alto custo. Embora os líderes mudem periodicamente para aumentar o tempo médio de vida da rede, a transmissão de dados tem uma relação quadrática com a distância. De acordo com o mesmo modelo de dissipação de energia apresentado em [2] temos que:

Energia gasta na transmissão:

$$E_{Tx}(k, d) = E_{Tx-elec}(k) + E_{Tx-amp}(k, d) \quad (1)$$

$$E_{Tx}(k, d) = E_{elec} \times k + E_{amp} \times k \times d^2 \quad (2)$$

Energia gasta na recepção:

$$E_{Rx}(k) = E_{Rx-elec}(k) \quad (3)$$

$$E_{Rx}(k) = E_{elec} \times k \quad (4)$$

onde:

k = número de bits da mensagem

d = distância

E_{Txelec} = Energia gasta na transmissão da mensagem

E_{rxelec} = Energia gasta na recepção da mensagem

$E_{Txelec} = E_{Rxelec} = E_{elec}$ = Energia gasta nos circuitos do rádio.

E_{amp} = Energia do amplificador de transmissão

Nossos experimentos utilizam esse mesmo modelo de dissipação de energia. Através das fórmulas (1) e (2) pode-se observar que quanto mais distante o destino da comunicação maior é o volume de energia gasto na comunicação. Em contraste com os argumentos apresentados em [2] para a utilização de apenas uma comunicação de longa distância, o ICA economiza energia exatamente utilizando-se de várias comunicações de curta distância. Os nós no ICA sabem a posição da ERB uma vez que ela faz uma difusão da sua posição no início do funcionamento da rede. O líder envia mensagens diretamente à ERB somente quando ele não possui outro líder vizinho na direção da ERB, ou quando este se recusa a agir como intermediário na entrega de pacotes. Um nó pode se recusar a ser intermediário quando seu nível de energia está baixo.

As figuras 3.4, 3.5, 3.6 se referem ao diagrama das mensagens trocadas pelo ICA. A figura 3.4 mostra as mensagens trocadas entre a ERB e um nó que é líder de grupo. A figura 3.5 apresenta as mensagens trocadas entre um líder e um nó que não é líder enquanto que a figura 3.6 apresenta as mensagens normalmente trocadas entre dois líderes.



Figura 3.4: Diagrama das mensagens trocadas entre a ERB e um líder

Para evitar o problema da morte prematura dos nós perto da ERB, os líderes no ICA, caso estejam esgotando sua energia, podem se recusar a retransmitir mensagens de outros grupos para a ERB. O líder pára de agir como ponte na entrega de mensagens de outros grupos quando sua energia atinge o limite determinado pela fórmula:

$$E_{min} > E = \sqrt{d * NB} \quad (5)$$

onde E é a mínima quantidade de energia que o nó deve ter para agir como ponte, d é a distância entre o nó e a ERB e NB é a média entre o número de bytes transmitidos em mensagens passadas e o tamanho da mensagem atual. Com essa abordagem para o cálculo

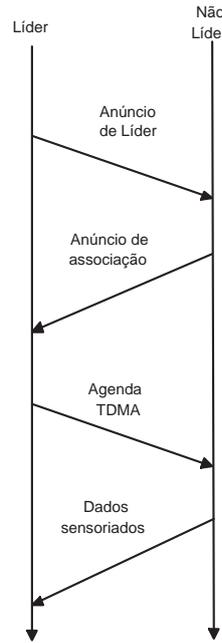


Figura 3.5: Diagrama das mensagens trocadas entre um líder e um nó sensor

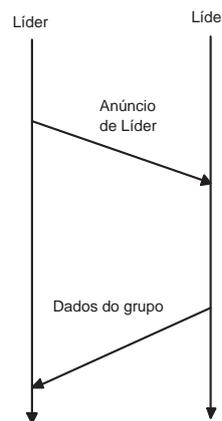


Figura 3.6: Diagrama das mensagens do ICA para a comunicação entre líderes

do número de bytes, o ICA avalia o histórico de dados transmitidos, mas atribui maior peso a tendência atual nos tamanhos dos pacotes. Assim, para um nó servir de ponte a energia residual do nó E_{min} deve ser maior que o valor de E , calculado pelo líder a cada transmissão.

Quando ocorre uma recusa em retransmitir dados, o líder que requisitou o serviço envia a mensagem diretamente a ERB, da mesma forma como ocorre no LEACH. Essa abordagem tenta impedir que os nós próximos à ERB tenham sua energia esgotada prematuramente. A recusa em retransmitir dados tende a ocorrer de forma organizada, ou seja, os nós mais próximos à ERB se recusarão a transmitir os dados antes dos nós mais distantes. Isto ocorre

porque os nós mais próximos à ERB são responsáveis por retransmitir os dados de seu grupo mais os dados de grupos que estão mais distantes que eles da ERB. Dessa forma os nós mais próximos da ERB retransmitem dados de toda a rede. Quando todos os líderes intermediários se recusarem a retransmitir os dados, o ICA irá se comportar exatamente como o LEACH, ou seja, todos os líderes enviam mensagens diretamente para a ERB. Esta é a ultima fase do ICA, quando todos os nós estão prestes a ficar sem energia.

A figura 3.7 ilustra o momento em que o líder do grupo 1, ao chegar ao nível mínimo de energia, se recusa a transmitir os dados do grupo 2. Após a recusa o líder do grupo 2 envia os dados de seu grupo e os dados dos grupos 3 e 4 diretamente para a ERB.

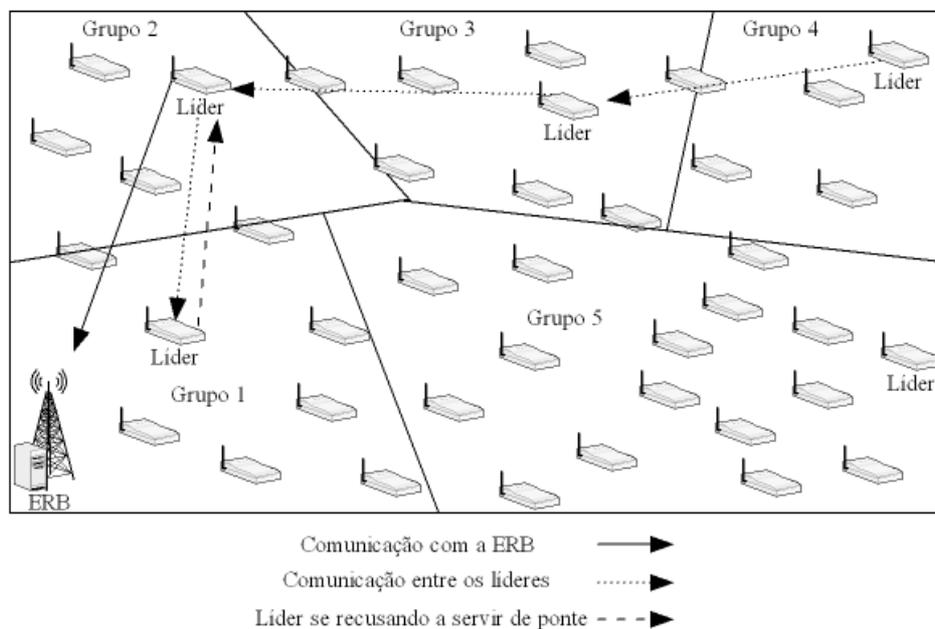


Figura 3.7: Cenário em que o líder do grupo 1 se recusa a servir de ponte para o grupo 2

As mensagens trocadas quando um nó se recusa a funcionar como ponte podem ser melhor observadas no diagrama de mensagens da figura 3.8.

Com esta abordagem os nós próximos à ERB, que durante o processo normal de funcionamento da rede gastam mais energia, garantem a transmissão dos dados de seus próprios grupos para a ERB. Por outro lado os nós mais distantes, que gastaram relativamente menos energia durante o roteamento, agora irão utilizar esta energia economizada para enviar suas mensagens diretamente à ERB. Desta forma o ICA tem como objetivo estender o tempo de vida da rede como um todo. Do ponto de vista do ICA, o cenário ideal é quando toda a

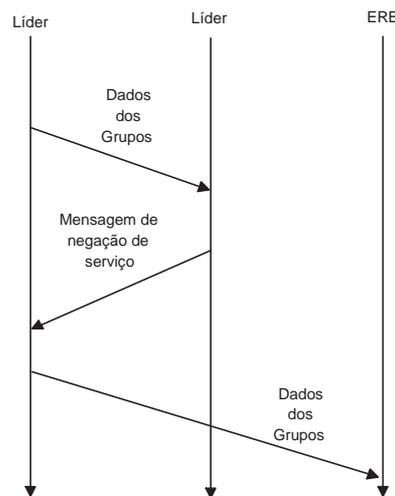


Figura 3.8: Diagrama das mensagens do ICA durante a negação de serviço

rede fica sem energia aproximadamente ao mesmo tempo. O algoritmo tenta alcançar uma degradação suave da energia da rede como um todo, não necessariamente nó a nó. Willing, Karl e Wolisz descrevem em [25] um sério problema para algoritmos geográficos gulosos que ele chama de *void*. O *void* é uma área sem nós entre o nó e a ERB. O ICA não sofre deste problema uma vez que se isto ocorrer o nó irá enviar a mensagem diretamente para a ERB.

A colisão de dados é outro problema que pode afetar o LEACH. Como todos os líderes enviam mensagens de seu grupo para a ERB, caso a rede seja grande e possua muitos líderes, ocorre um aumento do número de colisões na transmissão dos dados e até na formação dos grupos e, por isso, dados dos grupos são perdidos. Para evitar esse problema no ICA os líderes não transmitem ao mesmo tempo.

Após o início de cada *round*, que ocorre de forma síncrona na rede como um todo, os nós líderes no ICA enviam as mensagens para a ERB no tempo t de acordo com a equação:

$$t = \sqrt{\frac{1}{d^2}} + q \quad (6)$$

onde t é a base de tempo em que os nós deveriam enviar mensagens, baseados no início do *round*, d é a distância entre o nó líder e a ERB e q é o tamanho do *round*. Todos os nós devem enviar seus pacotes em um tempo randômico que varia entre t e $t+t/10$. Este intervalo foi adicionado como forma de diminuir a possibilidade de colisão de dados entre líderes que estão à mesma distância da ERB. Pode ser observado pela fórmula (6) que nós mais distantes da ERB irão transmitir suas mensagens primeiro. Isto organiza o processo de envio de dados através da rede, possibilitando inclusive o tratamento das mensagens, se for

necessário. Algoritmos de agregação de dados poderiam ser usados se isso for interessante para a aplicação. A próxima seção mostra um pseudo-código do funcionamento do ICA.

3.4 ICA pseudo-código

Segue abaixo o pseudo-código do ICA.

```

(1) For each node {
(2)   For each round {
(3)     roundStartTime = getTime();
(4)     itIsClusterHead = DecidesIfItIsClusterHead();
(5)     if(itIsClusterHead){
(6)       advertiseClusterHead();
(7)       nodesInTheCluster = waitForNotifications();
(8)       schedule = createsAnSchedule(nodesInTheCluster);
(9)       advertiseSchedule(nodesInTheCluster,schedule);
(10)      // Data transmission phase
(11)      monitoredData = receiveMonitoredData(nodesInTheCluster);
(12)      transmissionTime = sqrt(1/distanceFromBS) + roundLenght;
(13)      transmissionTime = transmissionTime + roundStartTime +
(14)      rand(transmissionTime/10);
(15)      nextNode = findsTheNearestNodeInBSDirection();
(16)      scheduledTransmission(transmissionTime, nextNode,
(17)      monitoredData);
(18)      averageMonitoredDataSize = (averageMonitoredDataSize + monitoredData) / 2;
(19)      //Data receiving phase
(20)      if (receivesDataRetransmissionRequest) {
(21)        minimumEnergyLevel = sqrt(distanceFromBS * averageMonitoredDataSize);
(22)        if (minimumEnergyLevel > actualEnergyLevel()) {
(23)          nextNode = getsTheNearestNodeInBSDirection();
(24)          retransmitData(nextNode, monitoredData);
(25)        } else //can not act as bridge
(26)          refusesRetransmitData();
(27)      }
(28)    } else { //if it is not a cluster head
(29)      cluster = decidesToWhichClusterItIsMember();
(30)      notifyClusterHead(cluster);
(31)    } //if the node is a cluster-head
(32)  } //for each round
(33) } //for each node

```

Na linha (3) deste pseudo-código os nós armazenam o tempo de início do *round* enquanto que em (4) o nó decide se ele é ou não líder de grupo. Se ele for líder, uma mensagem de difusão é enviada para informar que se trata de um nó líder. Em (7), os nós líderes esperam por mensagens de associação dos nós do grupo. Nas linhas (8) e (9) o líder agenda o momento em que o nó não líder pode transmitir os dados e envia este dado para cada um dos nós de seu grupo. Com isso, o nó não líder sabe quando ele pode transmitir os dados. Desta forma, evita-se a colisão de dados dentro do grupo. Na linha (11) o nó líder recebe, dos nós do grupo, os dados sensoriados em seu grupo. Nas linhas (12) a (14) é calculado, segundo a fórmula (6), o momento em que o líder deve enviar seus dados ao líder mais próximo ou à ERB. Na linha (15) o nó busca a frequência de transmissão do líder mais próximo (que é o líder para o qual ele enviará seus dados). Na linha (16) a transmissão dos dados do grupo é agendada. Entre as linhas (20) e (27) o nó, ao receber uma requisição para enviar dados de outros grupos à ERB, decide se ele vai retransmitir estes dados ou se ele vai se recusar a atuar como ponte nessa transmissão. Caso ele recuse a retransmitir estes dados o nó líder que requisitou o serviço é comunicado desta decisão. Com isso, pode-se reenviar os dados diretamente para a ERB. Se o nó não for líder de grupo, ele decide a qual grupo ele vai se associar e notifica ao líder escolhido a sua intenção de associação (linhas (28) a (31)).

Capítulo 4

Análise e simulação do ICA

Para avaliar o algoritmo ICA realizamos simulações e o comparamos com o LEACH e o LEACH-C.

Para a realização dos experimentos, estendemos o módulo do simulador NS (Network Simulator) desenvolvido pelo projeto uAMPS [63]. Assim, a comparação entre o ICA e o LEACH foi feita utilizando os mesmos parâmetros de simulação da implementação original do LEACH. Ou seja, utilizamos o mesmo modelo de tráfego, energia, rádio e propagação do sinal.

4.1 Ambiente utilizado

As simulações foram realizadas em dois computadores Pentium 4, 1.6GHz, 512MB RAM executando o sistema operacional Debian Linux 3.0. O simulador utilizado foi o NS (Network Simulator) versão 2.1b9 [64]. O módulo original do uAMPS foi desenvolvido para a versão 2.1b5a do NS. Fizemos uma atualização no módulo para a versão 2.1b9 do NS.

4.2 Cenários de simulação

Na simulação do ICA seis cenários diferentes foram criados para comparar os algoritmos. Os cenários variam em termos de número de nós e dimensão da rede, são eles:

- Rede 1: 50×50 com 50 nós.
- Rede 2: 50×50 com 100 nós.

Densidade das redes	
Rede	Densidade
1	$\frac{50}{2500} = 0,02$
2	$\frac{100}{2500} = 0,04$
3	$\frac{200}{2500} = 0,08$
4	$\frac{50}{10000} = 0,005$
5	$\frac{100}{10000} = 0,01$
6	$\frac{200}{10000} = 0,02$

Tabela 4.1: Densidade das redes simuladas

- Rede 3: 50×50 com 200 nós.
- Rede 4: 100×100 com 50 nós.
- Rede 5 100×100 com 100 nós.
- Rede 6: 100×100 com 200 nós

A rede 5 foi utilizada, assim como no LEACH, como rede padrão para as simulações.

Pode-se observar que os cenários de simulação utilizados possuem densidades diferentes. Essas densidades são mostradas na tabela 4.1.

Em todos os experimentos os nós foram aleatoriamente depositados no espaço da rede e para cada experimento os cenários variaram 33 vezes. O intervalo de confiança dos resultados é de 95%.

Nos experimentos, assim como em [2], os rádios dos nós podem controlar suas potências de transmissão de modo a consumir o mínimo de energia para atingir seu destino. Quando não estão em uso os rádios podem ser desligados para economizar energia.

Nas simulações do ICA, no início de cada simulação, cada nó possui 1J de energia e no primeiro conjunto de experimentos a ERB se encontra na posição (0,0). No segundo conjunto de experimentos a ERB está na posição (50,175) assim como em [2]. Pode-se observar que no segundo conjunto de experimentos a ERB está fora da rede e a uma distância mínima de 75 metros do sensor mais próximo. Os outros parâmetros também seguem os valores adotados em [1] e podem ser visualizados na tabela 4.2. A condição de parada dos experimentos é atingida, assim como nas simulações realizadas em [2], quando menos de 5% da rede se encontra ativa.

Parâmetros das simulações	
Parâmetro	Valor
Número esperado de líderes	5%
Condição de parada da simulação	Quando menos de 5% da rede se encontra ativa
Tipo de Rede	Homogênea e hierárquica
Tamanho do Pacote	250Kb
Tempo de cada Round	36s
Número de frequências disponíveis	Igual ao número de líderes + 1
Tamanho do <i>frame</i>	1,6s

Tabela 4.2: Parâmetros das simulações

4.3 Resultados

Esta seção analisa os resultados dos experimentos e compara os resultados obtidos pelo ICA com os resultados obtidos pelo LEACH e pelo LEACH-C. Os parâmetros utilizados na simulação foram apresentados na seção anterior.

A figura 4.1 mostra o número médio de pacotes transmitidos nos diferentes cenários. Pode-se observar que para todos os cenários, o número de pacotes enviados pelo ICA foi superior tanto aos do LEACH quanto aos do LEACH-C. A média do ganho no número de pacotes enviados, obtido pelo ICA, foi de 45% com relação ao LEACH-C, o segundo melhor protocolo analisado. Entretanto, quando o número de nós da rede aumenta, redes 3 e 6, a quantidade de pacotes enviados pelo ICA diminui, ficando próximo ao número enviado pelo LEACH-C. Na Rede 3 o ICA envia 13,5% mais dados enquanto que na rede 6 o ganho é de apenas 3,91%. Como com o aumento do número de nós mais líderes são eleitos, o ICA e o LEACH enviam mais mensagens de difusão para a formação dos grupos, ocorrendo assim um maior número de colisões e diminuindo a eficiência da formação dos grupos. Já o LEACH-C, que determina a formação de grupos de forma centralizada, o aumento do número de nós não afeta a eficiência da formação dos grupos.

Verificamos na tabela 4.3 que, mesmo considerando a variância dos resultados, o ICA é melhor do que o LEACH e o LEACH-C em todas as redes.

A figura 4.2 apresenta a média de dados transmitidos com relação ao tempo para uma rede do tipo 5. Pode-se observar que o ICA transmite mais mensagens e durante um maior período de tempo. Entretanto, se for observado o tempo de 140s no gráfico, pode-se verificar que o ICA, nesse momento, transmite menos mensagens que o LEACH e o LEACH-C. Isto ocorre porque a latência do envio das mensagens no ICA é maior do que no LEACH e no

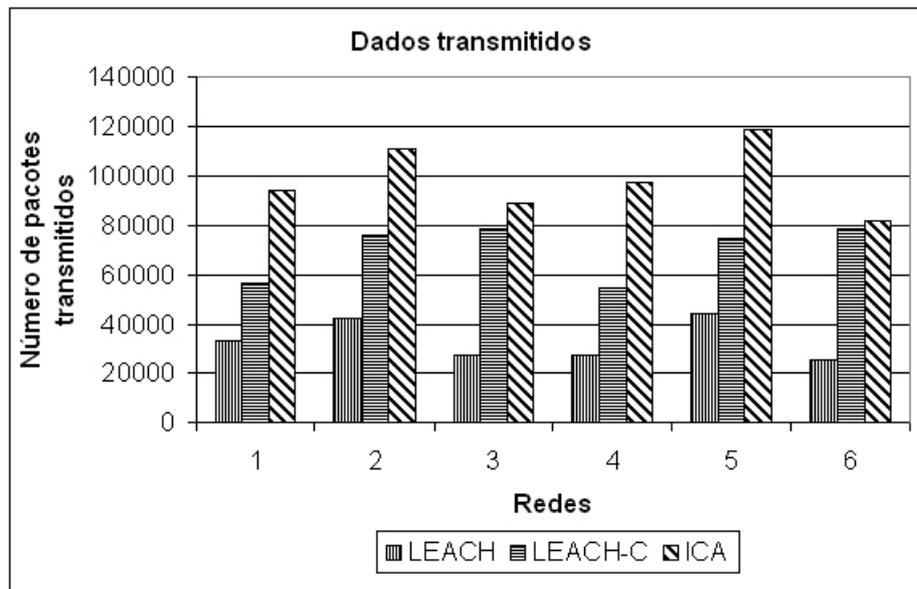


Figura 4.1: Dados transmitidos

Variância dos dados transmitidos						
Algoritmo	Rede 1	Rede 2	rede 3	rede 4	rede 5	rede 6
LEACH	394,4	229,0	521	452,4	302,3	407,24
LEACH-C	54,3	50,2	29,9	65,9	51,2	33,8
ICA	631,2	678,7	724,0	740,8	724,8	633,6

Tabela 4.3: Variância dos dados transmitidos

LEACH-C. Esse comportamento já era esperado uma vez que tanto no LEACH quanto no LEACH-C ocorre o envio direto de mensagens entre os líderes e a ERB. No ICA há o repasse das mensagens entre os líderes até que as mesmas cheguem à estação base.

A figura 4.3 apresenta o tempo médio para o primeiro nó se tornar inoperante. Esse parâmetro é importante pois em uma rede de sensores, a morte ou desligamento de um nó, significa que uma parte da rede, que antes era monitorada, está descoberta. Como podemos observar na figura 4.3 o aumento do tempo gasto para que o primeiro nó fique sem energia, quando comparado com o LEACH-C variou de 204,9% na rede 2 a 367,6% na rede 1. A variância deste resultado pode ser verificada na tabela 4.4. Com os resultados da figura 4.3 e da figura 4.4 pode-se concluir que o ICA atrasa em pelo menos duas vezes o aparecimento de áreas descobertas na rede, se comparado ao LEACH e ao LEACH-C.

O gráfico da figura 4.4 apresenta a média do tempo de vida da rede. Esse tempo é importante para vários tipos de RSSFs, uma vez que quanto maior esse tempo, maior é o

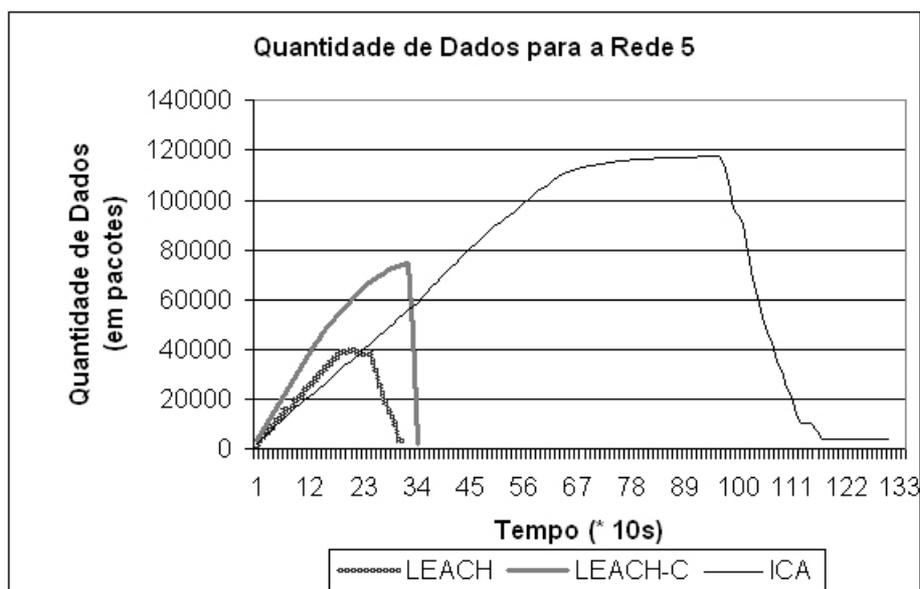


Figura 4.2: Dados transmitidos no tempo

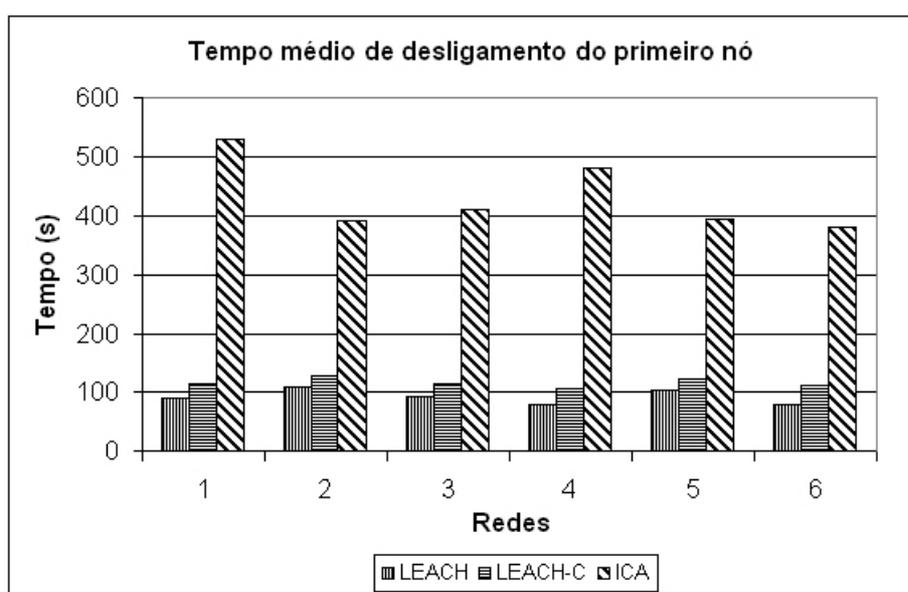


Figura 4.3: Tempo médio de desligamento do primeiro nó

tempo em que ela estará monitorando o ambiente. Pode-se observar na figura 4.4 que o LEACH não consegue prolongar o tempo de vida da rede em muito mais do que 200s. Esse fato possui um efeito diferente na quantidade de dados enviados pelo experimento e uma influência direta nos gráficos apresentados nas figuras 4.1 e 4.2. O LEACH-C apresenta, na média, um desempenho melhor do que o LEACH, mas quando comparado com o ICA o tempo de vida da rede no LEACH-C é de duas a três vezes menor. O acréscimo no tempo de

Variância do desligamento do primeiro nó (em segundos)						
Algoritmo	Rede 1	Rede 2	rede 3	rede 4	rede 5	rede 6
LEACH	5,13	7,07	7,30	7,04	7,18	6,77
LEACH-C	4,47	4,55	4,60	4,48	4,48	4,68
ICA	14,66	9,68	12,47	14,57	11,29	11,03

Tabela 4.4: Variância do tempo de desligamento do primeiro nó

vida da rede causado pelo uso do ICA foi de 198,2% a 329,5% sobre o LEACH-C que obteve o segundo melhor desempenho. A variância destes resultados é mostrada na tabela 4.5.

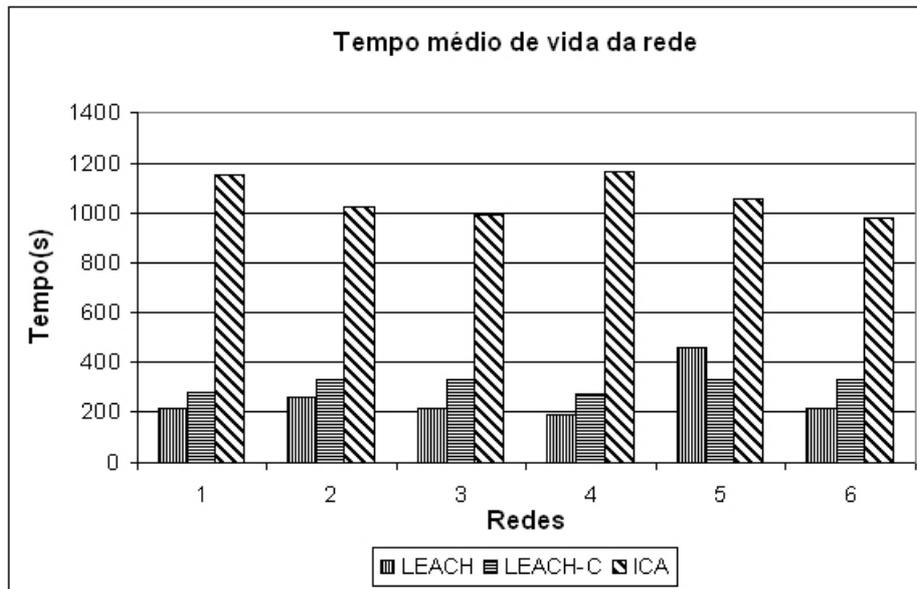


Figura 4.4: Tempo médio de vida da rede

Pode-se observar na figura 4.5 o mapa de energia do LEACH executando um exemplo da rede do tipo 5. Nessa figura vemos o mapa de energia nos tempos 100s e 200s. O eixo Z representa a quantidade de energia dos nós e os eixos X e Y representam as coordenadas dos nós. Pode-se observar que esse mapa de energia é irregular. Há uma grande variação no volume de energia gasto pelos nós. Pode-se observar também que houve uma expressiva queda no nível geral de energia da rede entre os tempos 100s e 200s. O mapa de energia para

Variância do tempo de vida da rede (em segundos)						
Algoritmo	Rede 1	Rede 2	rede 3	rede 4	rede 5	rede 6
LEACH	13,50	15,69	13,21	13,18	18,17	13,90
LEACH-C	12,87	14,63	14,27	13,35	14,28	14,59
ICA	31,07	31,45	26,91	34,70	31,15	31,55

Tabela 4.5: Variância do tempo de vida da rede

o tempo 300s não está disponível, uma vez que, nesse cenário, a rede ficou sem energia antes de alcançar os 300s.

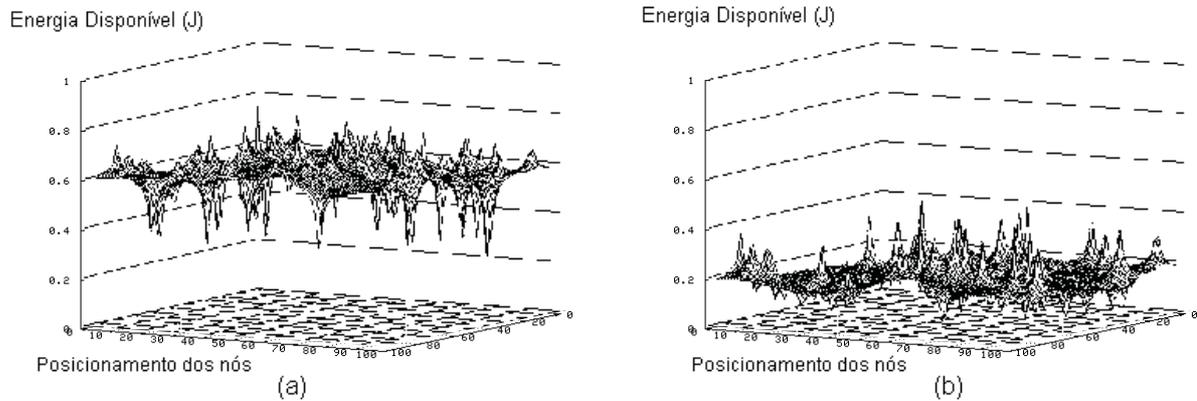


Figura 4.5: Mapa de Energia para o LEACH na rede 5. (a) no tempo 100s e (b) no tempo 200s. O eixo Z representa a energia restante na rede.

Os mapas de energia do mesmo experimento para o LEACH-C e para o ICA são mostrados na figura 4.6 e na figura 4.7, respectivamente. Pode-se observar que o comportamento tanto do LEACH quanto do LEACH-C é semelhante embora o LEACH-C apresente uma quantidade residual de energia maior. A figura 4.6 (c) apresenta o mapa de energia do LEACH-C em 300s. A não ser em alguns poucos pontos de cobertura, toda a rede está descoberta.

Pode-se observar nos gráficos da figura 4.7 o mapa de energia para o ICA. A figura 4.7 (a) apresenta o mapa de energia para 100s. A quantidade de energia no tempo $t=100s$ não é somente maior do que o LEACH e o LEACH-C, mas também apresenta uma melhor distribuição entre os nós da rede. A figura 4.7 (b) apresenta a rede 5 executando o ICA no tempo $t=300s$. Pode-se observar que a quantidade de energia é expressivamente maior do que no LEACH-C. Como esperado, os nós mais próximos da estação base, posição $(0,0)$, possuem menos energia que os nós mais distantes. Isto ocorre pois, em uma abordagem multi-hop, os dados de toda a rede têm que passar por esses nós fazendo com que gastem mais energia. Esta é a maior motivação para, a partir de um dado momento, os nós se recusarem a retransmitir dados. Neste cenário, somente quando a simulação chega no tempo $t=600s$ é que os nós próximos da estação base possuem aproximadamente a mesma quantidade de energia que o LEACH e o LEACH-C no tempo $t=200s$.

Podemos observar que além de o ICA consumir uma menor quantidade de energia, ele também apresenta uma melhor distribuição da energia. A figura 4.7 (d) apresenta a rede no

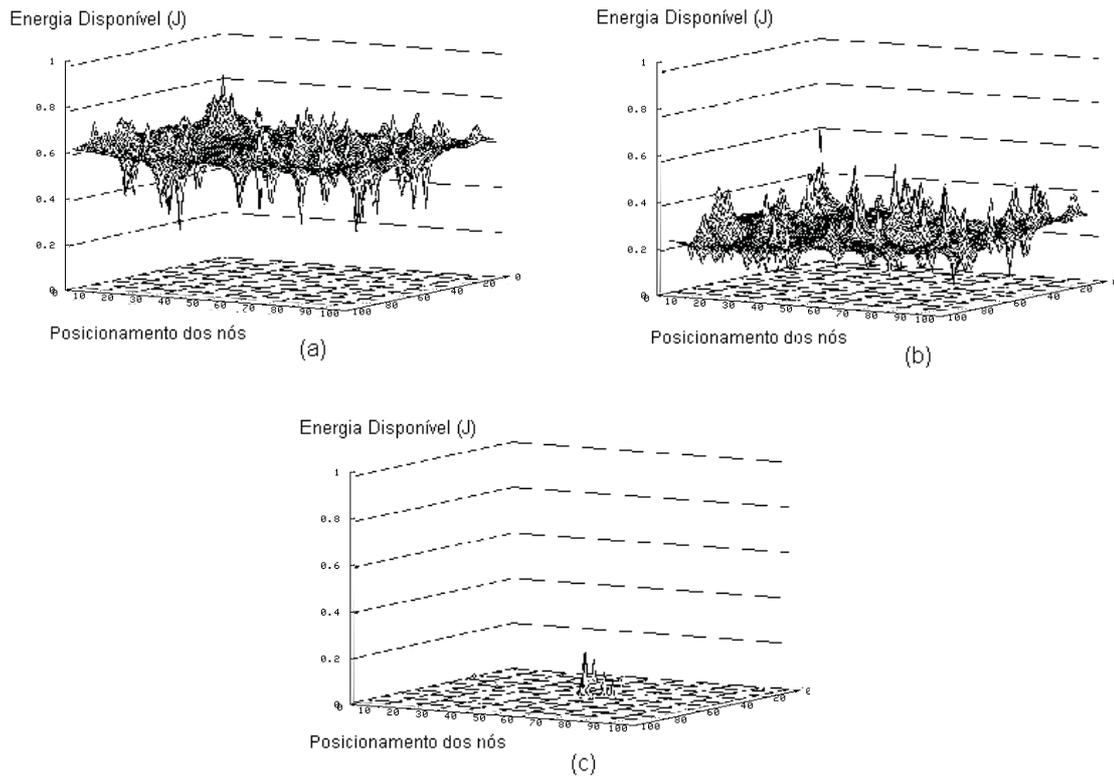


Figura 4.6: Mapa de Energia para o LEACH-C, na rede 5 (a) no tempo 100s, (b) no tempo 200s e (c) no tempo 300s

tempo $t=900s$. Embora comece a existir algumas áreas descobertas, a maior parte da rede ainda está coberta, incluindo os nós próximos à estação base. Isto demonstra não somente a eficiência do mecanismo de negação de roteamento, mas também a eficácia da fórmula (6) responsável por gerenciar este processo. É possível notar a degradação lenta da energia da rede e a quantidade de energia em toda a rede é aproximadamente a mesma. Finalmente, a figura 4.7 (e) mostra a rede no tempo de $t=1000s$. Podemos observar que, como esperado, os nós próximos à estação base ficam sem energia antes que o restante da rede. Embora a fórmula (6) tenha apresentado um bom desempenho, ela é basicamente uma heurística sendo baseada no tráfego anterior e não há garantia que este será o mesmo no futuro. Para aplicações que requerem uma extrema confiança na cobertura, uma nova fórmula pode ter que ser desenvolvida.

A seguir serão apresentados os resultados para as redes 4 e 5 com a ERB na mesma posição indicada em [2] e [1]. Ao analisar os resultados percebe-se que a alteração da posição da ERB não surtiu grande efeito nos resultados dos experimentos. Serão apresentados os resultados

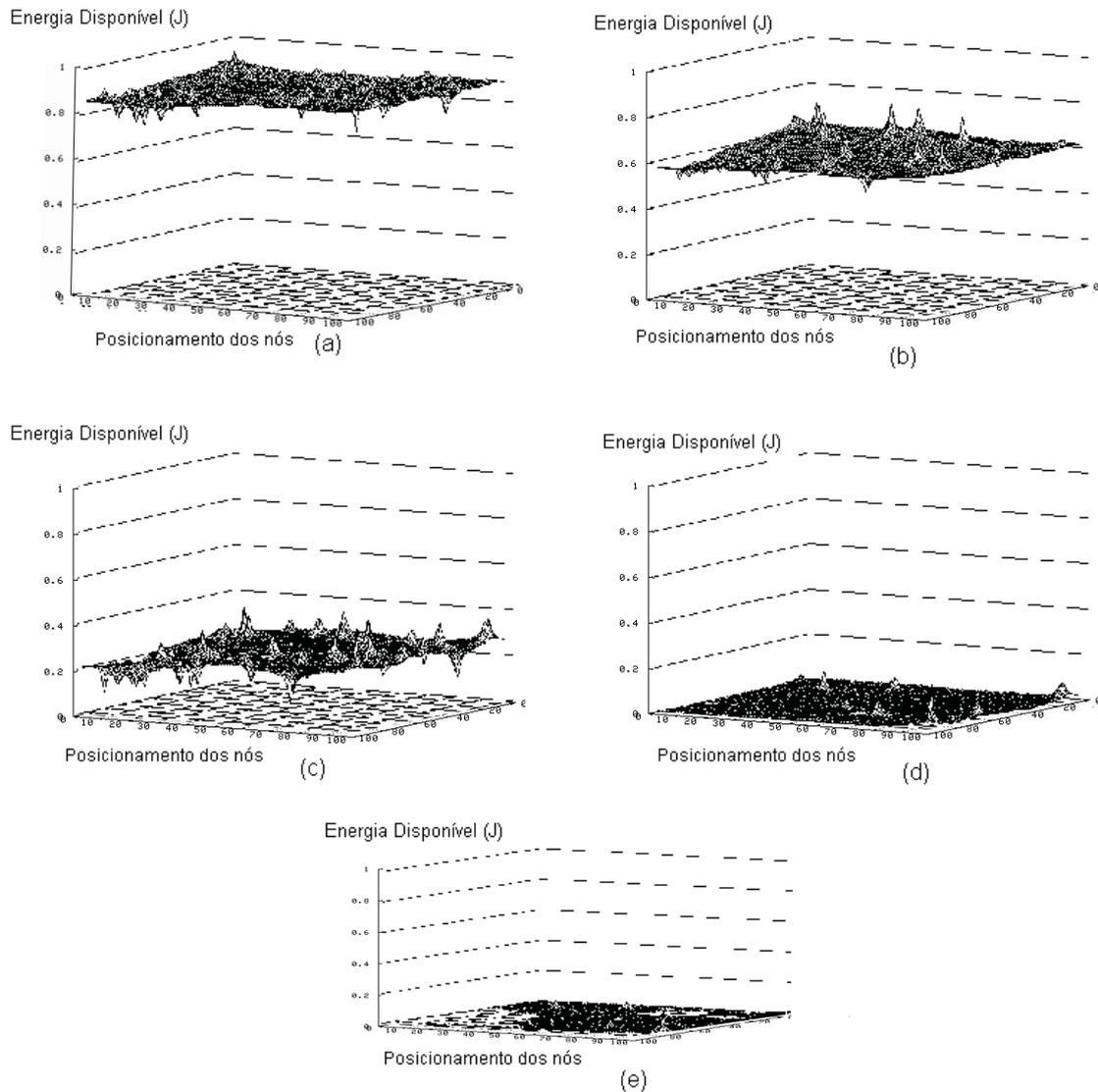


Figura 4.7: Mapa de Energia para o ICA, na rede 5, nos tempos (a) 100s, (b) 300s, (c) 600s, (d) 900s, (e) 1000s

dos experimentos realizados para as redes 4 e 5, pois a rede 5 é exatamente a mesma estudada em [2] e a rede 4 apresenta uma diminuição da densidade nesta rede.

A figura 4.8 mostra a quantidade de pacotes transmitidos para as redes 4 e 5 com a estação base na mesma posição de [2] (50, 175). Na rede 4 o ICA transmite 69% mais dados que o LEACH-C e 184% mais que o LEACH. Para a rede 5, o ICA transmite 35,5% mais pacotes que o LEACH-C e 186,5% mais do que o LEACH. Esses resultados são parecidos com aqueles demonstrados na figura 4.1. A variância destes resultados é mostrada na tabela 4.6.

A figura 4.9 apresenta o tempo de esgotamento do primeiro nó quando a ERB está na mesma posição de [2]. Na média o primeiro nó a ficar sem energia na rede 4 para o ICA foi

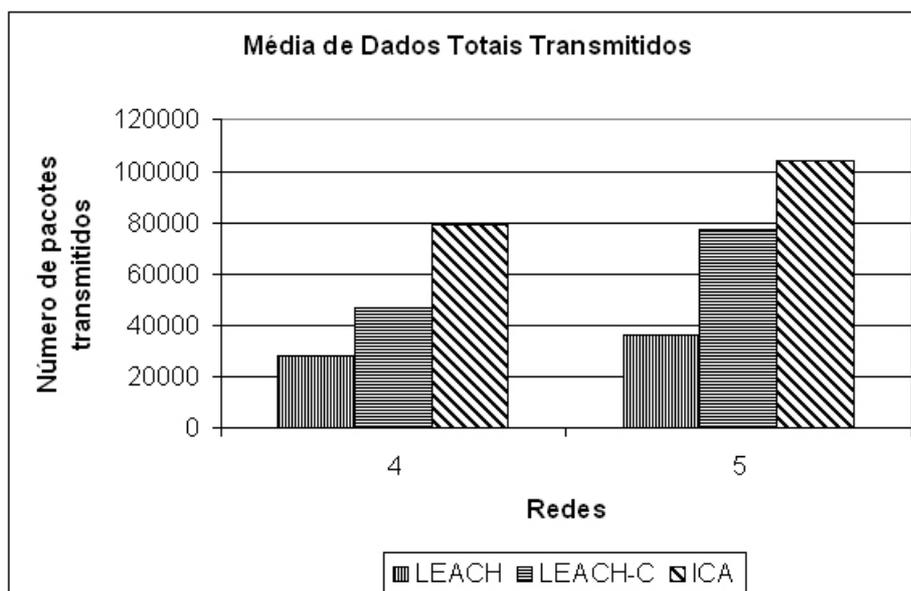


Figura 4.8: Dados transmitidos para as redes 4 e 5 com a ERB na posição (50,175)

Variância dos dados totais transmitidos		
Algoritmo	Rede 4	Rede 5
LEACH	480,3	262,2
LEACH-C	63,37	58,2
ICA	604,9	643,8

Tabela 4.6: Variância dos dados totais transmitidos em uma rede com a ERB na posição (50,175)

no instante $t=245s$ enquanto que no LEACH-C e no LEACH isso ocorreu nos instantes 42,4s e 40s, respectivamente. Na rede 5, para o ICA, o primeiro nó ficou sem energia no tempo de 375s enquanto que no LEACH-C e no LEACH isto ocorreu nos instantes 128,4s e 66,6s, respectivamente. A variância destes resultados é mostrada na tabela 4.7. Esses resultados são próximos aos apresentados na figura 4.3.

A figura 4.10 apresenta o tempo médio de vida da rede para o segundo conjunto de experimentos. O ICA obteve um ganho sobre o LEACH-C de 309% para a rede 4 e de 200% para a rede 5. A variância desses resultados é mostrada na tabela 4.8.

O mapa de energia também não apresenta uma diferença significativa em relação aos apresentados com a estação base na posição (0,0). O comportamento geral é o mesmo, contudo deve-se chamar a atenção para o fato que agora, novamente, os nós que gastam mais energia são os mais próximos da ERB. Isto pode ser visto na figura 4.11.

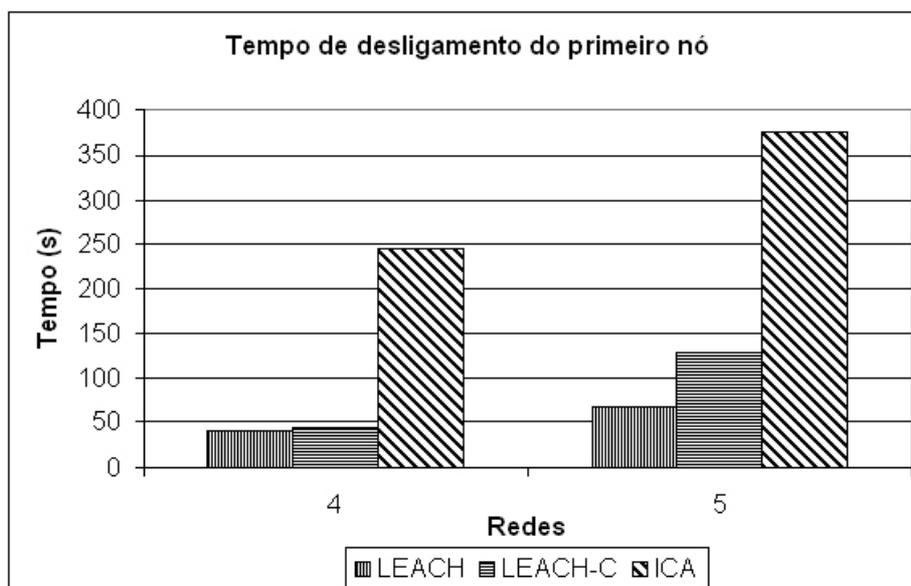


Figura 4.9: Tempo médio de desligamento do primeiro nó com a ERB na posição (50,175)

Variância do desligamento do primeiro nó (em segundos)		
Algoritmo	Rede 4	Rede 5
LEACH	9,6	9,7
LEACH-C	9,1	9,6
ICA	15,2	15,9

Tabela 4.7: Variância do tempo de desligamento do primeiro nó na rede em que a ERB está na posição (50,175)

4.4 Observações sobre o ICA

Vimos neste capítulo que o ICA consegue uma maior eficiência no consumo de energia dos nós da rede prolongando o seu tempo de vida e conseqüentemente, enviar mais pacotes à ERB.

Entretanto, como o ICA foi baseado no LEACH, ele apresenta alguns problemas não resolvidos e que foram "herdados" do LEACH. São eles:

- Não há nenhuma preocupação com a Qualidade de Serviço no envio das mensagens.
- Não é considerada a possibilidade de existirem nós congestionados na rede e que, por isso, não conseguirão encaminhar seus dados para a ERB.
- Não levou-se em consideração, nas simulações, a destruição de nós, fato que em Redes de Sensores é muito pouco provável de ocorrer.

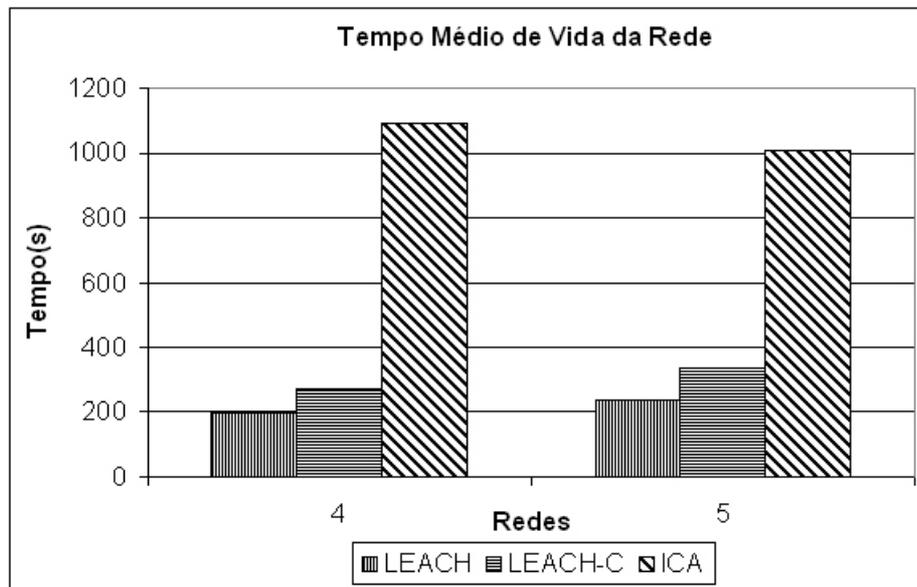


Figura 4.10: Tempo médio de vida da rede com a ERB na posição (50,175)

Variância do tempo de vida da rede(em segundos)		
Algoritmo	Rede 4	Rede 5
LEACH	5,26	5,84
LEACH-C	4,80	6,04
ICA	28,35	23,20

Tabela 4.8: Variância do tempo de vida na rede em que a ERB está na posição (50,175)

- Não há garantia que os líderes de grupo estarão uniformemente distribuídos pela rede. Assim, existe a possibilidade de que os líderes estejam concentrados em uma parte da rede e longe do restante dos nós. Isto faria com que haja um grande gasto de energia na rede ou que exista um líder que possua muitos nós em seu grupo.
- Mesmo que um líder fique sem energia, os nós pertencentes ao grupo que perdeu o líder tentam enviar seus dados para o líder mas, como eles não sabem que o líder esgotou sua energia, esses dados não são enviados à ERB.

Foi tentando resolver alguns dos problemas do ICA que iniciou-se um estudo de como prover uma melhor Qualidade de Serviço na entrega desses dados. Com isso, surgiu a proposta de uma técnica que permite prover QoS em RSSFs. Essa técnica será melhor descrita nos capítulos 5 e 6 desse trabalho.

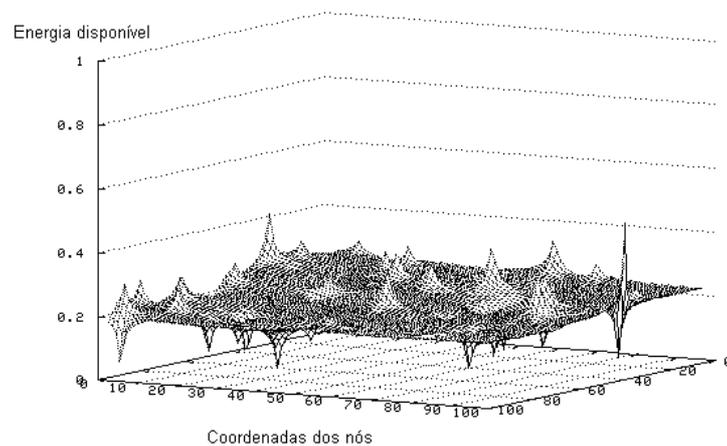


Figura 4.11: Mapa de Energia para o ICA na rede 4 no tempo $t=600s$, com a ERB na posição (50, 175)

Capítulo 5

Overlay for WireLess networks (OWL)

Neste capítulo iremos apresentar a técnica *Overlay for WireLess networks* (OWL). Essa técnica é uma proposta que visa a melhoria do nível de QoS das RSSFs através do uso de Redes *Overlay* (RONS).

Ao desenvolver protocolos para RSSFs é necessário levar em consideração a suscetibilidade dessas redes a falhas que podem ser causadas por diversos motivos dentre eles, a danificação física do nó, ataques à rede, o término da energia, a interferência ou até mesmo a latência causada por congestionamentos na rede. Temos ainda que, dependendo da forma como os sensores forem depositados no ambiente a ser monitorado, muitos nós sequer irão iniciar o funcionamento.

Em [53] é mostrado que até 20% dos nós em uma comunicação sem fio falham durante o envio dos dados. Assim, é interessante tentar resolver possíveis problemas como atraso e falhas de nós. A técnica OWL foi idealizada como forma de contornar as falhas da rede e atender a requisitos mínimos de qualidade. O objetivo é tentar garantir diferentes níveis de QoS para atender às diferentes necessidades das diferentes aplicações. Com isso, através da utilização de RONS, pretende-se prover QoS para os pacotes prioritários.

Com a utilização da técnica OWL pretende-se auxiliar o roteamento de forma genérica, flexível e oferecer níveis diferenciados de QoS às aplicações. Acreditamos que uma maneira de prover este tipo de roteamento é através da utilização de redes *overlay*. Com as redes *overlay* é possível prover um novo nível de abstração na rede tornando-a independente de mudanças futuras na arquitetura.

5.1 Utilizando RONS em RSSFs

Uma rede *overlay* é uma arquitetura que permite que se detecte uma falha ocorrida na rede durante um determinado período. Assim, como os nós agem de maneira cooperativa, é possível contornar falhas pontuais na rede. Através da monitoração cooperativa, uma rede *overlay* consegue identificar a qualidade dos caminhos monitorados e, assim, decidir por qual caminho deve enviar os pacotes de uma aplicação específica.

Por exemplo, em cenários como a monitoração de uma floresta é esperado que, caso haja um incêndio, muitos sensores sejam danificados. Assim, é necessário que os algoritmos desenvolvidos para essa aplicação sejam capazes de contornar as falhas destes nós. Além disso, mesmo que os nós não sejam danificados, caso haja a ocorrência de um evento, é esperado que o número de pacotes enviados pelos nós localizados na área da ocorrência do evento seja maior. Isso pode, dependendo do evento, fazer com que a rede de sensores venha a sofrer com problemas de congestionamento e acesso ao meio comprometendo a Qualidade de Serviço necessária à aplicação. A utilização da rede *overlay*, como uma abstração da interconexão real dos nós, serve como forma de identificar e evitar caminhos possivelmente danosos ao tráfego de informações.

Em [65] as redes *overlay* são utilizadas para aumentar a confiabilidade na transmissão de dados na Internet. O nosso trabalho mostra que a utilização de redes *overlay* também pode ser utilizada para aumentar a qualidade de serviço em RSSF. Para validar os resultados foram feitas simulações utilizando uma rede hierárquica e homogênea.

Em nosso trabalho, as mensagens não prioritárias são enviadas fazendo-se um revezamento entre cada uma das possíveis rotas até o destino. Como as mensagens não prioritárias são utilizadas para a coleta de informações da rede e a cada envio essas mensagens são enviadas por um caminho diferente, é possível que a ERB faça a coleta de métricas de várias possíveis rotas. De posse dessas informações, a ERB informa periodicamente aos líderes dos grupos as melhores rotas para que os dados cheguem até ela. A ERB consegue determinar essas rotas porque a rota é armazenada nos pacotes enviados.

Na técnica OWL, cada nó repassa os pacotes prioritários para uma das três melhores opções que atendem aos requisitos da aplicação. O algoritmo executa um *round robin* entre as três melhores rotas que atendem aos requisitos de QoS do pacote, evitando assim, o esgotamento

dos recursos das melhores rotas. O nó baseia sua decisão de rotas nos requisitos individuais de QoS de cada pacote e no seu conhecimento sobre a rede. Assim, cada pacote será repassado para o seu próprio caminho na rede. Nessa abordagem, a responsabilidade de roteamento é distribuída na tentativa de, com isso, aumentar o tempo de funcionamento da rede. A técnica OWL também diminui a sensibilidade da rede a ataques e congestionamentos, uma vez que não somente uma rota será utilizada, diminuindo assim o impacto da perda de um nó ou de uma rota inteira.

A escolha da melhor rota é dada por:

$$B \subseteq R \quad (7)$$

$$|B| \leq 3 \quad (8)$$

$$\forall x \in R \wedge \forall b \in B | x \in B \implies f(x) > f(b) \quad (9),$$

onde R é o conjunto de todas as rotas de um nó, B é o conjunto das melhores rotas (de acordo com as métricas definidas), b corresponde a uma das rotas disponíveis e f é a função que calcula a qualidade da rota em função dos requisitos de QoS da rota.

A escolha da rota é dada por:

$$\forall b \in B | k = \min(t(b)) \quad (10),$$

onde k é a rota escolhida e t(b) é a função que retorna a última vez que a rota foi usada.

As RSSFs são centradas em dados e muitas vezes a arquitetura da rede é baseada nas características dos dados transferidos pela rede. Assim, ninguém conhece melhor os requisitos de cada pacote do que a própria aplicação que o gerou. É natural que cada aplicação possua requisitos diferentes de QoS e, possivelmente, não seja adequado aplicar o mesmo esquema de QoS para todas as aplicações. Assim, a própria aplicação define os requisitos de QoS que ela necessita.

A figura 5.1 mostra a interconexão entre os nós sensores em uma determinada área. Os círculos demonstram o alcance de transmissão de cada nó. Em uma RSSF, os enlaces podem ser bidirecionais (por exemplo, nós I e J) ou assimétricos (por exemplo, nós G e C onde G alcança C mas C não alcança G). Consideramos que a ERB alcança todos os nós, mas no cenário da figura 5.1, somente A e B alcançam diretamente a ERB. A figura 5.2 mostra uma possível interconexão dos nós formando uma rede *overlay*. Nesse caso, independentemente da conexão real dos nós, eles enviam pacotes somente para os seus vizinhos na rede *overlay*. Cada nó da rede *overlay* atua como um roteador que, através de informações repassadas pela

ERB, tenta encontrar a melhor rota para cada pacote.

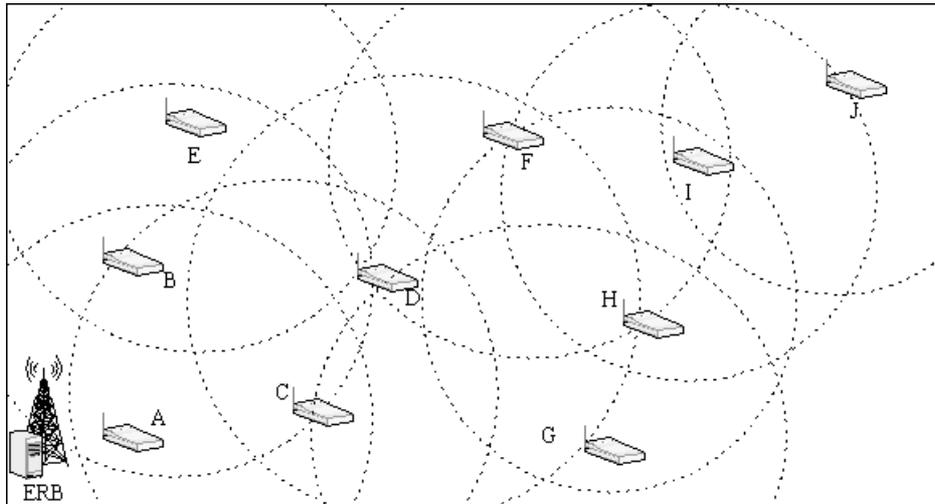


Figura 5.1: Alcance de transmissão de diferentes nós sensores

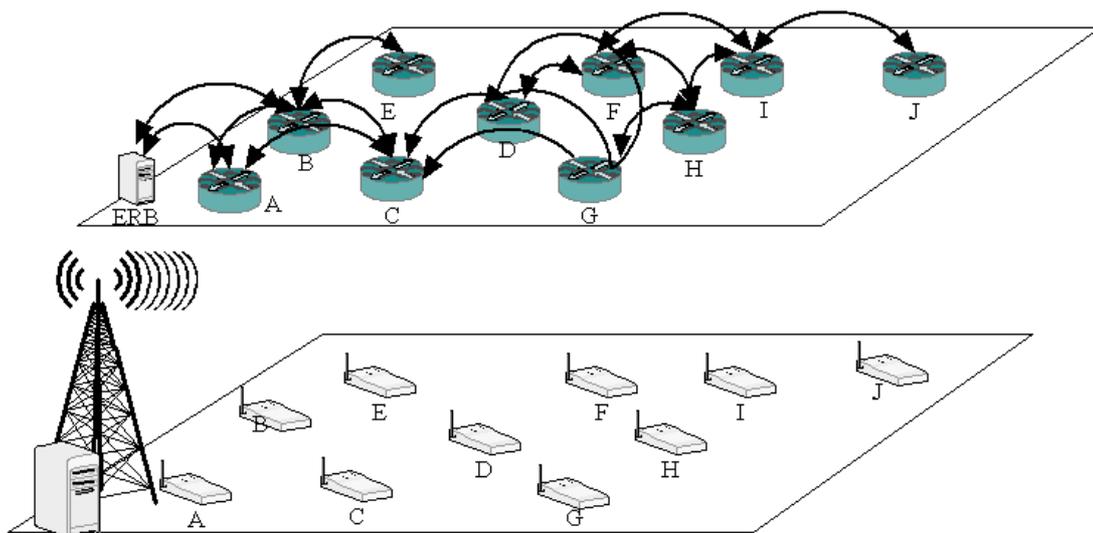


Figura 5.2: Mesmo cenário da figura 5.1 mas mostrando a rede *overlay* formada pela conexão entre os nós

A utilização de redes *overlay* abstrai as características individuais da rede possibilitando assim focar na qualidade requerida pela aplicação. Nesta proposta as melhores rotas são identificadas pelos nós *overlay* e baseadas em parâmetros de QoS de cada pacote pois parte-se do pressuposto que quem conhece melhor as necessidades de QoS do pacote é a aplicação que gerou o mesmo.

5.2 Arquitetura da rede

Este trabalho é focado em redes de sensores que trabalham cooperativamente na entrega de mensagens, ou seja, os nós usam outros nós como roteadores, pois nem todos os nós estão próximos o bastante da ERB para enviar seus dados diretamente. Além disso, neste trabalho, a ERB é capaz de enviar mensagens diretamente para todos os nós da rede. Nesse modelo, sensores vizinhos se comunicam em uma frequência, mas recebem as mensagens da ERB em outra frequência. Assim, as mensagens da ERB não interferem na comunicação dos sensores. Esta arquitetura segue os princípios encontrados no projeto de RSSFs onde consideramos que a ERB não sofre das mesmas restrições de processamento e energia dos nós sensores normais.

Os nós das RSSFs podem coletar dados de um ambiente continuamente, responder a uma requisição ou funcionar com um misto dessas duas abordagens. A proposta deste trabalho é aplicável a essas abordagens uma vez que a aplicação irá informar os requisitos de QoS que melhor atende às suas necessidades.

5.3 Overlay em RSSFs

Neste trabalho, a exemplo de Slijepcevic et al [66] foram consideradas duas classes de mensagens: as prioritárias e as não prioritárias. Utilizam-se as mensagens não prioritárias como forma de coletar informações da rede. Entende-se por mensagens não prioritárias aquelas que, caso sejam perdidas, não representam grandes danos ao funcionamento da rede ou da aplicação.

As mensagens não prioritárias são enviadas em um esquema de *round robin* para cada possível rota até o destino final. Assim, garantimos que todas as rotas serão testadas. Dessa forma, teremos uma maior chance de encontrar uma rota que se adeque melhor aos parâmetros de QoS exigidos por uma pacote prioritário.

Para os pacotes prioritários, a aplicação insere os requisitos de QoS no pacote. Para validar a proposta foram considerados os seguintes parâmetros de QoS: atraso fim-a-fim, pacotes perdidos e a energia residual. Entretanto, outros parâmetros de QoS poderiam ser utilizados. Para cada um desses parâmetros de QoS a aplicação insere um nível de importância. Um nó, ao receber o pacote prioritário, analisa os requisitos de QoS do pacote e o envia para uma das três melhores rotas que atendam aos requisitos de QoS do pacote. Ao utilizar o pacote para

armazenar os parâmetros de QoS teremos certeza que diferentes aplicações, com diferentes níveis de QoS, poderão utilizar a mesma infra-estrutura da rede ao mesmo tempo.

Todas as mensagens são identificadas e através desse identificador, a estação rádio base sabe se alguma mensagem anterior foi perdida e pode ainda, estimar a qualidade fornecida pelo caminho percorrido pela mensagem. De posse desses dados a ERB informa, individualmente a cada nó líder, quais os requisitos máximos de QoS que cada rota com a qual ele se comunica pode atender. O nó armazena, então, as três melhores rotas para um determinado parâmetro de QoS. Assim, ao analisar os requisitos de QoS de um pacote prioritário o nó envia esse pacote para uma das três melhores rotas.

Na técnica OWL, a melhor rota é aquela que melhor atende aos requisitos de QoS do pacote. Se houver empate entre duas rotas, o desempate é feito de acordo com o segundo parâmetro mais importante de QoS. É importante ressaltar que os pacotes não prioritários não possuem requisitos específicos de qualidade.

5.4 Pseudo-código

Nesta seção apresentaremos o pseudo-código do algoritmo que deve ser executado nos nós sensores e na ERB.

5.4.1 Nós sensores

```

(1)//Código executado nos nós sensores
(2) A aplicação informa aos nós sensores os requisitos de QoS que ela necessita.
(3) Para cada nó que receber os requisitos da aplicação {
(4)   Coleta os dados;
(5)   If(É Pacote prioritário){
(6)     Cria um pacote com os requisitos informados pela aplicação;
(7)     Verifica os três melhores caminhos para este pacote;
(8)   }Else{
(9)     Cria um pacote não prioritário;
(10)    Verifica as rotas atuais para a ERB;
(11)  }
(12)
(13)  Em um esquema de round robin envia os pacotes para a opção escolhida;
(14)  If (período de coleta de dados > 1){
(15)    Atribui o próximo round como o round que o dado será enviado;
(16)  }
(17) }
(18)
(19)If (o nó receber uma mensagem de outro líder){
(20)  If(É Pacote prioritário){
(21)    Lê as opções de QoS do pacote;
(22)    Verifica quais são as três melhores rotas;
(23)
(24)  }Else{
(25)    Cria um pacote não prioritário;
(26)    Verifica as rotas atuais para a ERB;
(27)  }
(28)
(29)  Em um esquema de round robin envia os pacotes para a opção escolhida;
(30)}
(31)If (nó receber a mensagem da ERB){
(32)  Atualiza a sua tabela com as melhores rotas;
(33) }

```

Na linha (2) a aplicação avisa aos nós sensores quais os requisitos de QoS que ela necessita. Caso os dados coletados sejam dados de pacotes prioritários (linhas 4 a 7) é criado um pacote com os requisitos informados pela aplicação. O nó sensor verifica, em sua tabela de rotas, os três melhores caminhos para que seus dados sejam enviados. Se o pacote não for um pacote prioritário o nó sensor cria um pacote não prioritário e verifica quais são as possíveis rotas que ele pode enviar este dado (linhas 9 e 10). Os pacotes são, então, enviados em um esquema de *round robin* para a opção escolhida (13). Na linha (14) é verificado se houve mudança de round. Se sim, o pacote será enviado após a formação dos novos grupos (15). Caso um nó líder receba a mensagem de outro nó líder é verificado se o pacote é prioritário. Se sim, o pacote é enviado,

em um esquema de *round robin* para uma das três rotas que melhor atendem os requisitos de pacote de QoS do pacote (linhas 20 a 24 e 29). Se o pacote não for um pacote prioritário é verificado quais as rotas atuais para a ERB e em seguida, esta mensagem é enviada em um esquema de *round robin* para uma das possíveis rotas (linha 24 a 27 e 29). Os nós líderes só recebem mensagens da ERB para informar que as rotas devem ser atualizadas. Se isso ocorrer, o nó atualiza a sua tabela de rotas (31 a 33).

5.4.2 ERB

```
(1) // Código executado na estação base
(2) Para cada pacote recebido {
(3)   Registra que o pacote foi recebido;
(4)   If (há algum pacote com ID com diferença maior do que 64 unidades deste ID){
(5)     Marca o pacote pendente como pacote perdido;
(6)     Atualiza os parâmetros de QoS;
(7)     If (Há mensagem agendada para ser enviada ao nó){
(8)       Atualiza o pacote com os novos parâmetros de QoS;
(9)     }Else {
(10)      Inicia o agendamento do envio de um pacote para informar o novo parâmetro de QoS da rota;
(11)    }
(12)  }
(13) // Verifica a rota do pacote a ser enviado;
(14)   if (Houve mudança nos parâmetros de QoS da rota){
(15)     If (há mensagem agendada para ser enviada ao nó){
(16)       Atualiza o pacote com os novos parâmetros de QoS;
(17)     }Else{
(18)       Inicia o agendamento do envio de um pacote para informar o novo parâmetro de QoS da rota;
(19)     }
(20)  }
(21)
(22)While (Existem mensagem a serem enviadas){
(23)   If (o temporizador do pacote expirou){
(24)     Envia a mensagem diretamente para o destino;
(25)   }
(26) }
```

A linha (2) inicia um *loop* que é executado para cada pacote recebido pela ERB. Na linha (3) a ERB registra que o pacote foi recebido. O *if* da linha (4) verifica se há alguma mensagem com ID cujo número de sequência é 64 unidades maior do que o ID de alguma mensagem que ainda não chegou. Se isso ocorrer supõe-se que a mensagem foi perdida (5). A linha (6) atualiza os parâmetros de QoS da rota em que o pacote foi perdido. A linha (7) verifica se já existe alguma mensagem de atualização de rotas agendada para ser enviada para

os nós afetados pela modificação de QoS da linha (6). Caso já exista o pacote é atualizado com os novos parâmetros de QoS da rota (8). Se não existir (10), é agendado o envio de pacotes de dados para todos os nós afetados pela mudança dos parâmetros de QoS da rede. O pacote não é enviado imediatamente. O servidor inicia um temporizador com o intuito de, neste intervalo, receber mais informações de QoS de outras rotas. Na linha (14) é verificado se houve mudança nos parâmetros de QoS das rotas em relação aos valores previamente armazenados. Caso haja mudanças e já exista uma mensagem agendada para ser enviada ao nó, o pacote com a mensagem a ser enviada é atualizado(16). Caso haja mudanças na rota mas ainda não existe nenhum pacote de atualização de rota agendado para ser enviado, a ERB agenda o envio de um pacote(18). Nas linha (22) a (26) os pacotes de atualização das rotas são enviados.

5.5 Custos

A utilização da técnica OWL acarreta em custos adicionais à RSSF. Caracterizaremos esses custos em termos de espaço, processamento e transmissão. Ignoraremos os custos da ERB pois assumimos que a ERB não tem limitações de energia, processamento ou potência de transmissão.

5.5.1 Custo de espaço

A figura 5.3 mostra o esquema genérico das mensagens considerado em nosso trabalho.



Figura 5.3: Esquema genérico da mensagem

A utilização do OWL aumenta o tamanho do pacote prioritário em quatro bytes. Destes, dois bytes são utilizados para incluir um *timestamp* no pacote. No byte seguinte, dois bits são usados para armazenar a importância de cada parâmetro de QoS e um bit para informar se o pacote é prioritário ou não. O último byte indica o menor nível de qualidade requerido, por exemplo, se o mais importante requisito de QoS for atraso fim-a-fim, há a necessidade

de informar que o atraso mínimo tem que ser menor do que um valor mínimo (0,5s, por exemplo). Nos pacotes não prioritários é necessário que se aumente somente o espaço para incluir o *timestamp* no pacote. Assim, o *overhead* no pacote prioritário é de quatro bytes e no não prioritário é de dois bytes.

A figura 5.4 mostra o formato padrão das mensagens prioritários e a figura 5.5 mostra o formato das mensagens não prioritárias, necessárias para o funcionamento da técnica OWL.

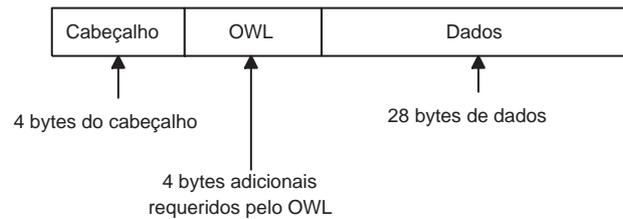


Figura 5.4: Esquema da mensagem prioritária

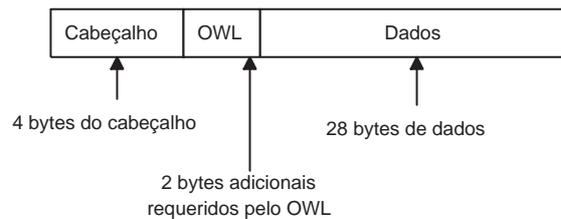


Figura 5.5: Esquema da mensagem não prioritária

Cada nó necessita ainda, de um byte adicional de memória para armazenar a referência para cada uma das rotas até a ERB.

5.5.2 Custo de processamento

Cada nó intermediário tem que processar cada pacote para encontrar os parâmetros de QoS desse pacote. Em [65] é demonstrado experimentalmente que o uso de redes *overlay* na Internet, acrescenta um custo extra para processar cada mensagem na camada de aplicação. Entretanto, será mostrado nas simulações que este custo é ínfimo se comparado aos ganhos na qualidade de serviço da entrega das mensagens. Pelos resultados das simulações verificamos que isto também se aplica às RSSFs.

5.5.3 Custo de transmissão

Periodicamente a estação base envia uma mensagem para os nós informando a qualidade de suas rotas. Para economizar energia, somente modificações nos comportamentos das rotas são enviados aos nós sensores. Quando a ERB possui alguma nova informação ela inicializa um temporizador e não envia esta informação até que este tempo expire. A idéia é evitar *hysteresis*, ou seja, evitar que as rotas fiquem em constante alteração. Assim, espera-se por mais informações sobre outras rotas e somente após isso são enviadas todas estas informações no mesmo pacote.

Capítulo 6

Análise e simulação da técnica OWL

Neste capítulo iremos avaliar a eficiência e o desempenho da técnica OWL. Comparamos a solução proposta com uma rede sem a utilização de *overlay* e com outra rede utilizando um algoritmo que sempre encontra a solução ótima. Embora seja difícil implementar esse algoritmo ótimo em uma rede real é importante utilizá-lo para verificar o quão próxima a solução proposta está da solução ideal.

6.1 Ambiente utilizado

As simulações foram realizadas em dois computadores Pentium 4, 1.6GHz, 512MB RAM executando no sistema operacional Debian Linux 3.0. O simulador utilizado foi o NS (Network Simulator) versão 2.1b9 [64] com o módulo uAMPS desenvolvido pelo MIT [63].

6.2 Cenário de simulação

O cenário padrão utilizado como base para realizar nossos testes foi uma rede 100×100 com 100 nós, depositados de forma aleatória na área a ser monitorada. Os resultados aqui apresentados são um dez percentil das simulações. Assim, cada experimento foi executado 50 vezes e foram retirados os cinco melhores e os cinco piores resultados. Com isso, os resultados apresentados neste capítulo são a média aritmética das quarenta simulações restantes. O intervalo de confiança utilizado nos resultados foi de 95%;

Em nossos experimentos, a ERB se encontra na posição (0,0) da rede e, assim como em [1], os rádios dos nós podem controlar suas potências de transmissão de modo a consumir o

mínimo de energia para atingir seu destino. Quando não estão em uso os rádios podem ser desligados para economizar energia. Os nós em nossas simulações estão organizados em uma rede hierárquica.

Para tornar os resultados mais reais, seguimos os resultados encontrados em [53]. Assim, em nossas simulações, 20% dos nós falham e, independentemente, outros 20% dos nós causam atrasos nas mensagens. No cenário padrão da simulação, a prioridade na transmissão dos dados foi, em ordem de prioridade, entrega do pacote, atraso e energia residual, respectivamente. Chamaremos esse pacote de tipo 1. São definidos ainda mais dois tipos de pacotes: 2 e 3. No pacote 2 as prioridades são, respectivamente, atraso fim-a-fim, entrega de pacotes e energia residual. No pacote 3 as prioridades são energia residual, entrega de pacotes e por último, o atraso fim a fim.

Em nossas simulações, cada nó possui energia infinita. Desse modo pode-se medir o *overhead* de energia causado pela técnica OWL na rede. Como a técnica OWL foi implementada no ICA os mesmos parâmetros de simulação da tabela 4.2 se aplicam aqui. O único parâmetro diferente foi a condição de parada das simulações. Na simulação do OWL simulamos até o tempo de 800 segundos.

6.3 Resultados

Esta seção analisa os resultados dos experimentos comparando a técnica OWL com o algoritmo ótimo e com o algoritmo sem a utilização de redes *overlay*. O algoritmo ótimo é um oráculo. Ele já sabe previamente quais nós irão falhar e, por isso, sempre encontra a melhor rota, só perdendo pacote caso não exista uma rota que atenda aos requisitos mínimos de QoS dos pacotes e ele seja obrigado a enviar o pacote para uma rota que não atenda aos requisitos de QoS dos pacotes. O algoritmo sem *overlay* envia pacotes em *Round Robin* para todas as rotas disponíveis. Entretanto, ele não leva em consideração qualquer possível prioridade dos pacotes.

A figura 6.1 mostra o número médio de pacotes prioritários perdidos nos cenários simulados. O número de pacotes perdidos é acumulativo. Podemos verificar assim que a técnica OWL perde menos pacotes no decorrer do tempo do que a rede sem *overlay*.

A figura 6.1 mostra o cenário onde todos os pacotes gerados são do tipo 1 e a figura 6.2

mostra o cenário onde os pacotes 1, 2 e 3 são gerados e transmitidos na mesma proporção. Os resultados são similares para ambos os experimentos. Assim, sem perda de generalidade, os próximos experimentos serão restritos ao cenário padrão.

No cenário da figura 6.1 a técnica OWL perdeu, no fim da simulação, uma média de 39,2 pacotes prioritários. Ou seja, 68,5% menos pacotes do que o algoritmo sem *overlay* e 42,5% mais pacotes do que o algoritmo ótimo. A variância deste resultado foi de 5,66 pacotes, 6,96 pacotes e 3,35 pacotes para o OWL, algoritmo sem *overlay* e algoritmo ótimo, respectivamente. É necessário ressaltar que mesmo o algoritmo ótimo pode perder pacotes quando não existem nós intermediários que levem até a ERB ou estes nós existem mas não são confiáveis. No cenário da figura 6.2 observamos que a técnica OWL perdeu 65,92% menos pacotes que o algoritmo sem *overlay* e 43% mais pacotes que o algoritmo ótimo. A variância deste resultado foi de 5,75 pacotes, 6,94 pacotes e 3,41 pacotes para a técnica OWL, algoritmo sem *overlay* e algoritmo ótimo, respectivamente.

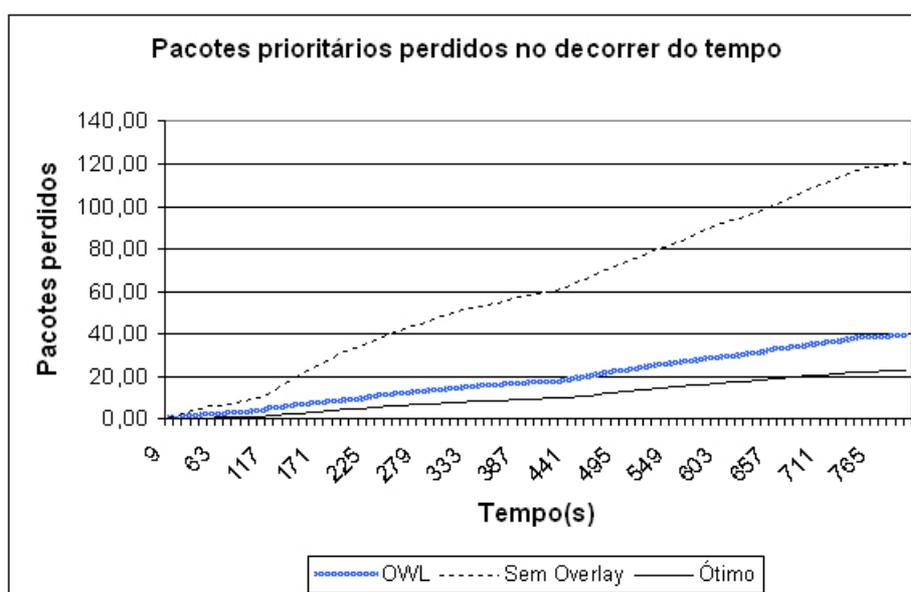


Figura 6.1: Pacotes prioritários totais perdidos

Podemos observar ainda, nas simulações que o número de pacotes não prioritários perdidos é praticamente o mesmo para os três métodos. O algoritmo ótimo perde em média 1097,62 pacotes enquanto que a técnica OWL perde em média 1099,5 pacotes e o algoritmo sem *overlay* perde em média 1100,16 pacotes. A variância deste resultado foi de 10,84 pacotes, 11,45 pacotes e 12,0 pacotes para a técnica OWL, algoritmo sem *overlay* e algoritmo ótimo,

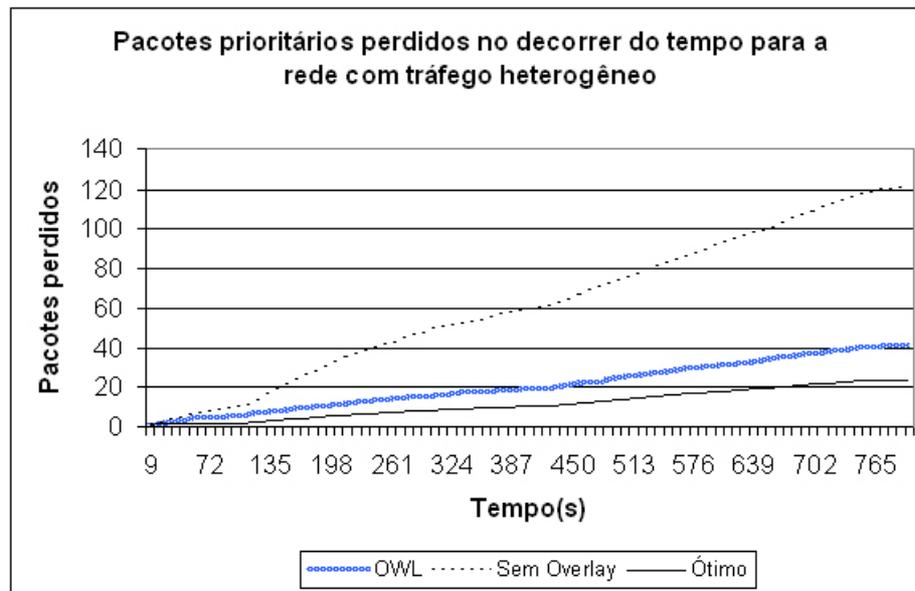


Figura 6.2: Pacotes prioritários totais perdidos com o tráfego heterogêneo

Pacotes prioritários perdidos			
Tipo de pacote	Sem <i>Overlay</i>	OWL	Ótimo
Pacote 1	40,36	13,01	7,49
Pacote 2	40,36	14,06	7,89
Pacote 3	40,36	14,19	8,22

Tabela 6.1: Pacotes prioritários perdidos

respectivamente.

A diferença entre os diferentes parâmetros de QoS para cada um dos tipos de pacotes são explicitadas nas tabelas 6.1 e 6.3. A tabela 6.1 mostra a média de pacotes perdidos para os experimentos onde temos diferentes tipos de pacotes na rede. Pode-se observar que tanto o algoritmo ótimo quanto a técnica OWL perdem menos pacotes do tipo 1 que outros tipos de pacotes. Isto já era esperado pois a maior preocupação do pacote 1 é justamente evitar a perda de pacotes. Entretanto, pode-se observar na tabela 6.1 que o algoritmo ótimo perdeu aproximadamente 29% menos pacotes que o OWL. No algoritmo sem *overlay* a diferença de pacotes perdidos entre os diferentes tipos de pacotes não foi expressiva, uma vez que esse algoritmo não considera a prioridade dos pacotes. A variância referente aos pacotes prioritários perdidos estão mostrados na tabela 6.2.

A tabela 6.3 apresenta a média do atraso dos pacotes no mesmo experimento da tabela 6.1. Na tabela 6.3, tanto para o algoritmo ótimo quanto para a técnica OWL, o atraso dos pacotes

Variância dos pacotes prioritários perdidos			
Tipo de pacote	Sem Overlay	OWL	Ótimo
Pacote 1	3,68	1,93	1,63
Pacote 2	3,69	2,08	1,35
Pacote 3	3,68	2,02	1,38

Tabela 6.2: Variância dos pacotes prioritários perdidos

Atraso dos pacotes prioritários (em segundos)			
Tipo de pacote	Sem <i>Overlay</i>	OWL	Ótimo
Pacote 1	0,48	0,30	0,21
Pacote 2	0,49	0,22	0,15
Pacote 3	0,49	0,38	0,27

Tabela 6.3: Atraso dos pacotes prioritários

Variância no atraso dos pacotes prioritários (em segundos)			
Tipo de pacote	Sem Overlay	OWL	Ótimo
Pacote 1	0,07	0,04	0,04
Pacote 2	0,07	0,03	0,02
Pacote 3	0,07	0,04	0,04

Tabela 6.4: Variância no atraso dos pacotes prioritários

do tipo 2 são menores que o atraso dos outros tipos de pacotes. Esse resultado é consistente com os requisitos do pacote 2 pois o seu objetivo é, em primeiro lugar, minimizar o atraso fim-a-fim. Podemos ver na tabela 6.3 que, para a métrica de atraso do pacote 2, a técnica OWL foi aproximadamente 104% melhor que o algoritmo sem *overlay* e o algoritmo ótimo foi 30,45% melhor do que a técnica OWL. O pior desempenho do pacote 3 pode ser explicado pelos seus requerimentos de QoS, onde, o atraso é o parâmetro menos importante. A variância referente aos atrasos dos pacotes prioritários estão mostrados na tabela 6.4.

A figura 6.3 mostra o número de pacotes perdidos no decorrer do tempo para a simulação padrão. Os valores apresentados nessa figura se referem a valores instantâneos, não sendo portanto, acumulativos. Podemos observar que o desempenho da técnica OWL fica sempre próximo do desempenho do algoritmo ótimo, chegando, em alguns instantes, a igualar o resultado.

Como era de se esperar, nem sempre o menor caminho é o melhor caminho. O número médio de passos do algoritmo ótimo foi 4,64 enquanto que o número médio de passos para o algoritmo sem *overlay* foi 5,62 e para a técnica OWL foi de 5,89 passos. A variância destes resultados foi de 0,4, 0,5 e 0,6 para o algoritmo ótimo, algoritmo sem *overlay* e a técnica

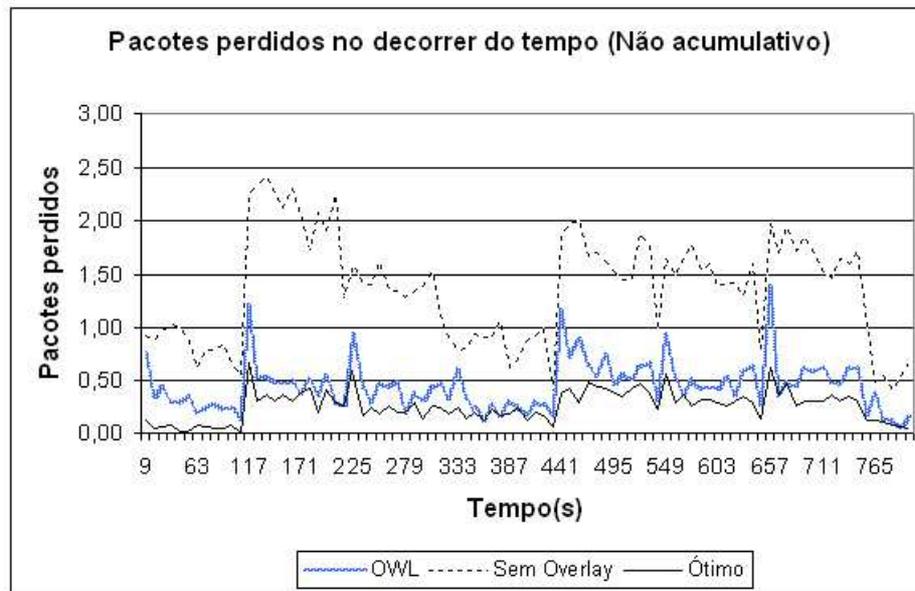


Figura 6.3: Número não acumulativo de pacotes perdidos

OWL, respectivamente. O algoritmo ótimo é um observador global e, com isso, reconhece a menor rota e, quando possível, escolhe o menor número de passos. Assim, é esperado que o algoritmo ótimo possua menores rotas que os outros algoritmos. O algoritmo sem *overlay* não escolhe o caminho mais confiável ao enviar um pacote prioritário. Assim, o número de pacotes prioritários perdidos é maior. A técnica OWL aumenta o tamanho médio da rota porque ela escolhe, com base nos dados que dispõe, o melhor caminho para a ERB de forma que a perda de pacotes prioritários seja minimizada. Assim, algumas vezes ela escolhe percorrer caminhos mais longos porém que atendem melhor aos requisitos de QoS dos pacotes. Entretanto, apesar do número de passos médio ser maior na técnica OWL do que o algoritmo sem *overlay*, observa-se ao verificar a variância do resultado que não podemos afirmar com uma confiança de 95% que a técnica OWL possui um número de passos médio maior do que o algoritmo sem *overlay*.

Em nossos experimentos, o consumo médio de energia da rede foi de 187,51J, 184,91J e 182,71J para o algoritmo ótimo, a técnica OWL e para o algoritmo sem *overlay*, respectivamente. A variância destes resultados foi de 6,2J, 5,8J e 5,8J para o algoritmo ótimo, a técnica OWL e para o algoritmo sem *overlay*, respectivamente. Nos nossos experimentos não há a retransmissão de mensagens quando elas são perdidas. Assim, como o algoritmo ótimo perde, em média, um número menor de mensagens, mais energia é gasta com a retransmissão de mensagens. Pelo mesmo motivo, a técnica OWL gasta um pouco mais energia do que o

algoritmo sem *overlay*. Entretanto, ao considerarmos as margens de erro, verificamos que os três algoritmos gastam praticamente a mesma quantidade de energia.

Pelo gráfico da figura 6.4 observamos ainda que o consumo de energia da técnica OWL é menor que o do algoritmo sem *overlay* caso haja um tráfego igual de mensagens com pacotes dos tipos 1, 2 e 3. Pelos resultados do gráfico, vemos que o consumo de energia da técnica OWL é 5% menor do que o algoritmo sem *overlay* e 3,8% pior do que o algoritmo ótimo. Esse resultado já era esperado pois, agora, aproximadamente 33,33% dos pacotes enviados possuem como principal objetivo economizar energia. Ao considerarmos a variância dos resultados da figura 6.4 podemos verificar que o gasto de energia continua praticamente o mesmo. A variância desses resultados foi de 4,1%, 3,8% e 3,7% para o algoritmo ótimo, a técnica OWL e para o algoritmo sem *overlay*, respectivamente.

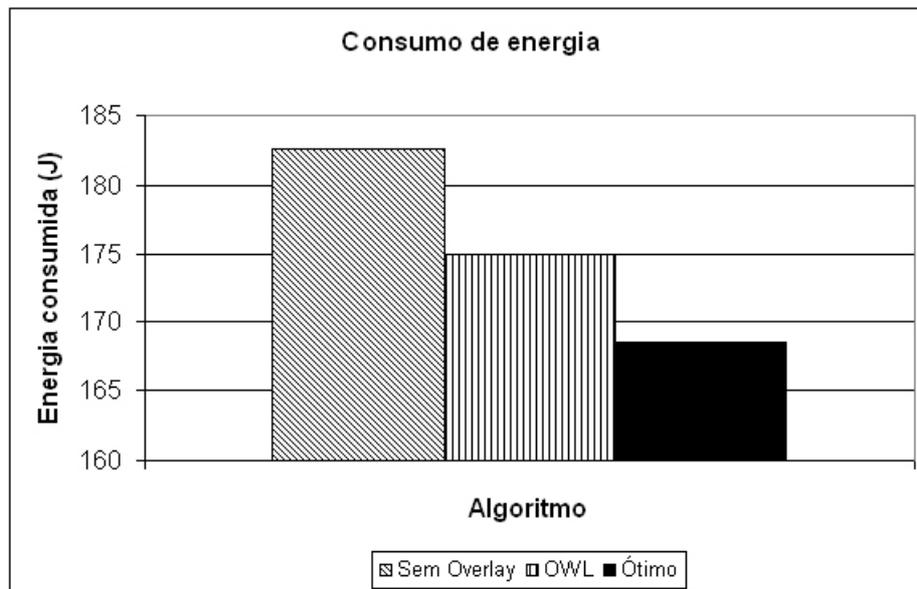


Figura 6.4: Energia consumida quando os três tipos de pacotes coexistem na rede

Na figura 6.5 podemos observar que o atraso médio dos pacotes prioritários para o cenário padrão na técnica OWL é cerca de 38,5% pior do que o algoritmo ótimo. Entretanto, a técnica OWL consegue um atraso 40% menor do que o algoritmo sem *overlay*. A variância deste resultado é de 8,1%, 5,2% e 3,3% para o algoritmo sem *overlay*, a técnica OWL e o algoritmo ótimo, respectivamente.

Pelo gráfico da figura 6.6 vemos que, caso o primeiro parâmetro de qualidade a ser verificado em todos os pacotes prioritários seja o atraso médio dos pacotes, a técnica OWL é

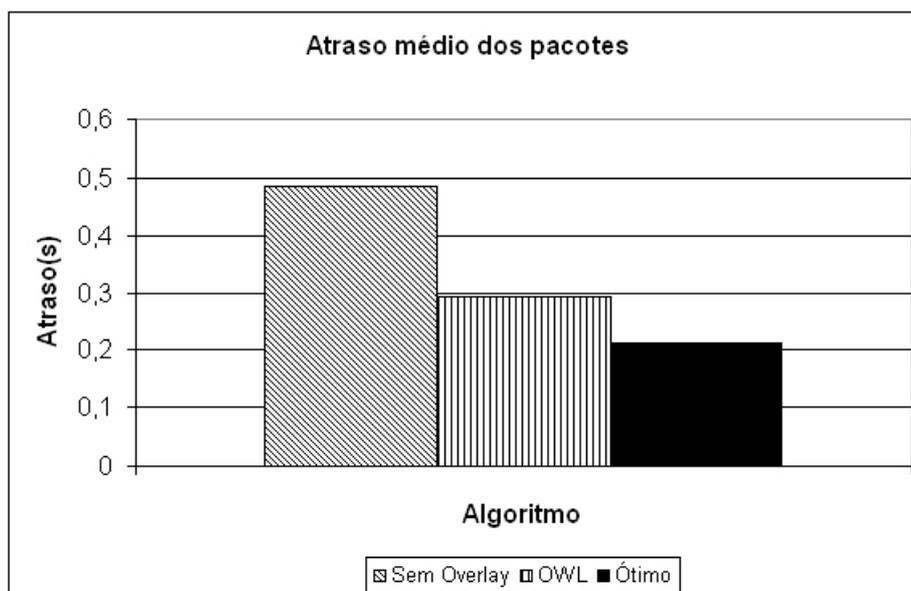


Figura 6.5: Atraso médio dos pacotes prioritários para o cenário padrão

cerca de 55,34% mais eficiente que o algoritmo sem *overlay* e é cerca de 30,45% pior do que o algoritmo ótimo. A variância deste resultado é de 8,2%, 5,4% e 3,0% para o algoritmo sem *overlay*, a técnica OWL e o algoritmo ótimo, respectivamente.

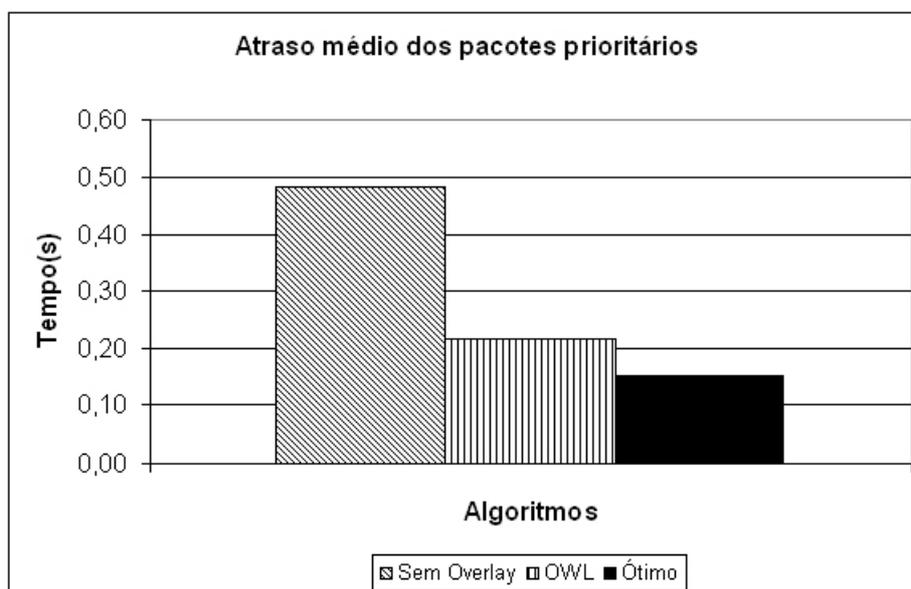


Figura 6.6: Atraso médio dos pacotes prioritários caso o parâmetro de maior prioridade seja o atraso dos pacotes

A partir do gráfico da figura 6.7 podemos observar o atraso da transmissão no tempo. A abordagem ótima demonstra qual seria o menor atraso possível em um determinado tempo.

Podemos ver que os resultados apresentados pela técnica OWL estão relativamente próximos dos resultados do algoritmo ótimo. Chamamos a atenção para o fato de que, no começo da simulação, a técnica OWL apresenta um atraso aproximadamente igual ao atraso apresentado pelo algoritmo sem *overlay*. Entretanto, a medida que a técnica OWL vai obtendo informações da rede, ela consegue reduzir o atraso dos pacotes prioritários. O atraso dos pacotes não prioritários é aproximadamente o mesmo para os três algoritmos.

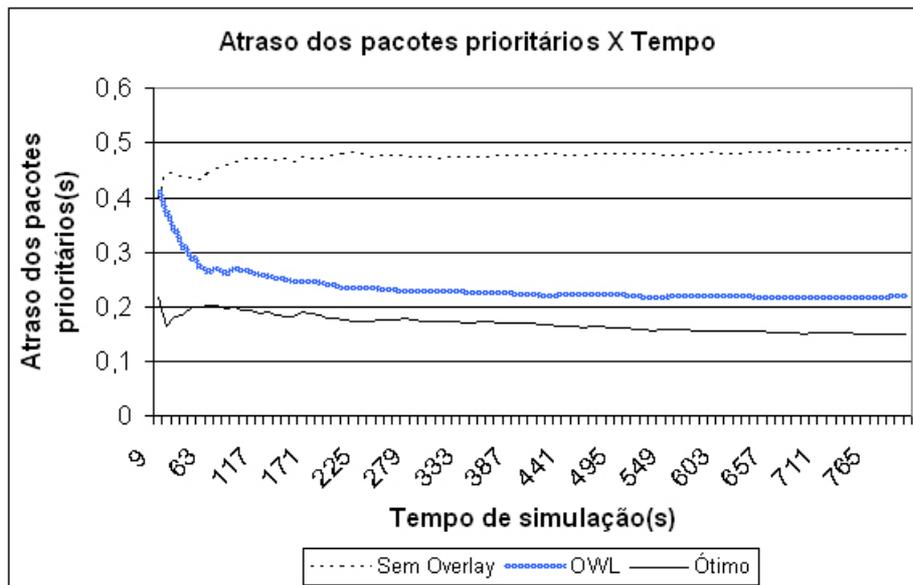


Figura 6.7: Atraso médio dos pacotes prioritários versus Tempo

O gráfico da figura 6.8 mostra o atraso dos pacotes prioritários caso os três tipos de pacotes coexistam na rede. Verificamos que o atraso da técnica OWL e do algoritmo ótimo aumentaram se comparado à simulação em que o principal objetivo dos pacotes era a diminuição do atraso. Esse resultado também já era esperado pois, agora, somente 33% dos pacotes possuem como principal objetivo diminuir o atraso na transmissão. Mesmo assim, a técnica OWL consegue ser 49% mais eficiente nessa métrica que o algoritmo sem *overlay*. A variância deste resultado já foi apresentada na tabela 6.4.

Pelos resultados apresentados é possível observar que a técnica OWL consegue ser melhor que o algoritmo que não utiliza *overlay* em todos os aspectos observados.

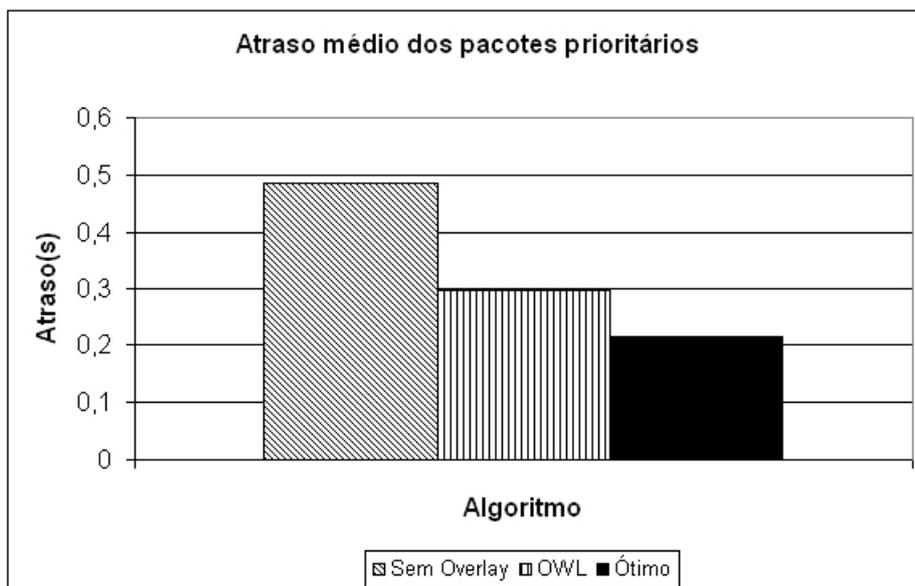


Figura 6.8: Atraso médio quando os três tipos de pacotes coexistem na rede

Capítulo 7

Conclusões e trabalhos futuros

Com o avanço da computação, vários paradigmas estão sendo quebrados possibilitando, assim, o surgimento de novas tecnologias.

Este trabalho apresentou duas contribuições principais, que são:

- A proposta do ICA, que é um novo algoritmo de roteamento para RSSFs.
- O estudo de uma técnica para se prover QoS em RSSFs.

A primeira contribuição foi a proposta do Inter Cluster routing Algoritm (ICA). O ICA tenta resolver problemas de diminuição do tempo de vida da rede, baixo envio de pacotes e colisão de pacotes. Esses problemas, presentes no LEACH, são críticos em redes de sensores sem fio. Através dos experimentos verificou-se que o ICA apresenta um aumento médio de 45% no número de dados transmitidos quando comparado com o LEACH-C, o segundo melhor protocolo analisado. O tempo médio para que o primeiro nó fique sem energia no ICA varia de 204,9% a 479,3% a mais que no LEACH-C.

Os experimentos também mostram que não somente as redes executando o ICA ficam ativas de 198,2% a 329,5% mais tempo que o LEACH-C, como também apresentam uma distribuição mais uniforme do volume de energia acumulado na rede. O ICA também apresenta uma degradação mais uniforme do nível de energia, que os outros protocolos avaliados. Caso o problema do atraso na entrega de mensagens seja crítico é preferível utilizar o LEACH que o ICA pois o LEACH envia suas mensagens diretamente para a ERB, sem intermediários. O ICA pode ser uma boa opção para cenários onde as redes de sensores precisam de um tempo de vida maior e uma degradação uniforme da quantidade de energia da rede.

Apesar do ICA ter se mostrado muito eficiente no consumo de energia e na quantidade de dados transmitidos, ele não é o melhor algoritmo em todos os casos. O atraso na entrega dos dados é bem maior no ICA do que em algoritmos similares como o LEACH e o LEACH-C. Isso ocorre porque no ICA as mensagens são enviadas líder a líder enquanto que, no LEACH e no LEACH-C, as mensagens são enviadas diretamente para a ERB. Assim, caso a redução do atraso na entrega das mensagens seja crucial para a aplicação, é melhor utilizar o LEACH do que o ICA.

A segunda contribuição deste trabalho foi a proposta da técnica *Overlay for WireLess networks*(OWL). A técnica OWL é um algoritmo que permite que diferentes aplicações, possivelmente com requisitos distintos de QoS, compartilhem a mesma estrutura da rede obtendo um desempenho melhor do que um algoritmo que não utilize a técnica de QoS proposta neste trabalho.

Os experimentos realizados demonstram que a utilização da técnica OWL reduz a perda de pacotes prioritários em até 68,5%. Os experimentos também demonstram que o custo dessa técnica em termos de energia é pequeno. Caso nenhum pacote na rede possua como primeira métrica de qualidade a economia de energia, o aumento do gasto da técnica OWL em relação a um algoritmo que não utiliza redes *overlay* foi de 1,02%. Entretanto, caso na rede existam três tipos de pacotes na mesma proporção e um destes tipos de pacotes tiver como principal prioridade a diminuição do consumo de energia, a técnica OWL consegue diminuir o consumo de energia em 5%.

Vimos também que a técnica OWL aumenta o número médio de saltos em 4,8%. Esse é o custo que se paga para se ter um aumento da qualidade de serviço fornecida aos pacotes prioritários. Acredita-se que o desempenho dessa técnica possa ser melhorado ainda mais.

Com relação à técnica OWL pode-se observar que apesar dessa técnica conseguir uma melhora na qualidade de serviço na entrega dos pacotes prioritários, não é garantido que os requisitos do tráfego desses pacotes serão atendidos. Portanto, para uma aplicação que exige que os requisitos de qualidade dos pacotes sejam atendidos, outro protocolo deve ser desenvolvido.

Existem várias extensões destes trabalhos. Contudo os próximos passos devem ser o tratamento estatístico tanto do ICA quanto da técnica OWL. também pretendemos determinar a porcentagem ótima de líderes no ICA de forma a otimizar o consumo de energia e diminuir

as colisões.

Além disso, pretendemos implementar ambos os algoritmos em sensores reais e confirmar os resultados simulados.

Em relação à técnica OWL, trabalhamos apenas com redes hierárquicas. Pretendemos avaliar se a nossa abordagem se aplica tanto a redes planas quanto a redes hierárquicas.

Outro trabalho que pretendemos fazer é a comparação da técnica OWL com algum outro algoritmo que provê QoS em RSSFs.

Referências Bibliográficas

- [1] W. Rabiner Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Conference on System Sciences (HICSS 2000)*, pages 3005–3014, January 2000.
- [2] Wendi Beth Heinzelman. *Application Specific Protocol Architectures for Wireless Networks*. PhD thesis, MASSACHUSETTS INSTITUTE OF TECHNOLOGY, june 2000.
- [3] Danyu Zhu, Matt W. Mutka, and Zhiwei Cen. Qos aware wireless bandwidth aggregation (qawba) by integrating cellular and ad-hoc networks. In *Quality of Service in Heterogeneous Wired/Wireless Networks (QSHINE)*, pages 156–163, 2004.
- [4] Arati Manjeshwar and Dharma P. Agrawal. Teen: A routing protocol for enhanced efficiency in wireless sensor networks. *International Parallel & Distributed Processing Symposium (IPDPS 2001)*, pages 2009–2015, April 2001.
- [5] Akyildiz I.F., Weilian Su, Sankarasubramaniam Y., and Cayirci E. A survey on sensor networks. *Communications Magazine*, pages 102–114, August 2002.
- [6] Andrew S. Tanenbaum. *Redes de Computadores*. Campus, 4^o edition, Agosto 2003.
- [7] Charles E. Perkins. *Ad Hoc Networking*. Addison Wesley Professional, 2001.
- [8] Linnyer Beatrys Ruiz, Luiz Henrique A. Correia, Luiz Filipe M. Vieira, Daniel F. Macedo, Eduardo F. Nakamura, Carlos M. S. Figueiredo, Marcos Augusto M. Vieira, Eduardo Habib Bechelane Maia, Daniel Câmara, Antonio A. F. Loureiro, José Marcos S. Nogueira, and Diógenes C. da Silva Jr. e Antônio O. Fernandes. Architectures for wireless sensor networks (in portuguese). In *Proceedings of the 22nd Brazilian Symposium on Computer*

- Networks (SBRC'04)*, pages 167–218, Gramado, RS, Brazil, May 2004. Tutorial. ISBN: 85-88442-82-5.
- [9] D. G. Anderson, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient overlay networks. *Symposium on Operating Systems Principles (SOSP)*, pages 131–145, October 2001.
- [10] Lakshminarayanan Subramanian, Ion Stoica, Hari Balakrishnan, and Randy H. Katz. Overqos: An overlay based architecture for enhancing internet qos. *Networked Systems Design and Implementation (NSDI 2004)*, pages 71–84, March 2004.
- [11] Gnutella. <http://www.gnutella.com>.
- [12] S. Shi and J. Turner. Routing in overlay multicast networks. *Infocom 2002*, pages 1200–1208, June 2002.
- [13] G. Peng. Cdn: Content distribution network. <http://citeseer.ist.psu.edu/peng03cdn.html>.
- [14] Eduardo Habib, Daniel Câmara, and A. F. Loureiro. Ica: Um novo algoritmo de roteamento para redes de sensores. *XXII Simpósio Brasileiro de Redes de Computadores*, May 2004.
- [15] C. Huitema. Routing in the internet. Prentice Hall, 1996.
- [16] T. Clausen, P. Jacquet (editors), C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L.Viennot. Optimized link state routing protocol (olsr). RFC 3626, October 2003. Network Working Group.
- [17] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsv) for mobile computers. *In Proceedings of the SIGCOMM'94, Conference on Communications Architectures*, pages 234–244, August 1994.
- [18] D. Johnson, D. Maltz, and J. Broch. *DSR The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks*.
- [19] C. Perkins and E. Royer. Ad-hoc on-demand distance vector (aodv) routing. *Second Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, pages 90–100, February 1999.

- [20] Bhaskar Krishnamachari, Deborah Estrin, and Stephen Wicker. Modelling data-centric routing in wireless sensor networks. *USC Computer Engineering Technical Report (CENG)*, pages 02–14, February 2002.
- [21] Chris Karlof and David Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. *International Workshop on Sensor Network Protocols and Applications*, pages 293–315, May 2003.
- [22] J.Hill, R. Szewczyk, A. Woo, S.Hollar, D. Culler, and K. Pister. System architecture directions for network sensors. *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000)*, pages 93–94, November 2000.
- [23] Lindsey S. and Raghavendra C.S. Pegasus: Power-efficient gathering in sensor information systems. *Aerospace Conference*, pages 1125–1130, May 2002.
- [24] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. *Mobile Computing and Networking (MobiCOM '00)*, pages 56–67, August 2000.
- [25] H.Karl, A.Willing, and A.Wolisz. Improving the energy efficiency of directed diffusion using passive clustering. *European Workshop on Wireless Sensor Networks (EWSN)*, pages 172–187, January 2004.
- [26] Williams B. and Camp T. Comparison of broadcasting techniques for mobile ad hoc networks. *International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '02)*, pages 194–205, June 2002.
- [27] Figueiredo C. M. S., Nakamura E. F., and Loureiro A. A. F. Protocolo adaptativo híbrido para disseminação de dados em redes de sensores sem fio auto-organizáveis. *In Proceedings of the 22nd Brazilian Symposium on Computer Networks (SBRC'04)*, pages 43–56, May 2004.
- [28] Nogueira J. M., Loureiro A. A. F, Macedo D. F., Santos A. L., and Correia L. H. A. A pro-active routing protocol for continuous data dissemination in wireless sensor

- networks. *In Proceeding of the 10th IEEE Symposium on Computers and Communications (ISCC'2005)*, pages 361–366, June 2005.
- [29] Nakamura E. F., Figueiredo C. M. S., and Loreiro A. A. F. Information fusion for data dissemination in self-organizing wireless sensor networks. *In Proceedings of the 4th International Conference on Networking*, pages 585–593, April 2005.
- [30] A. A. F.; Mateus G. R.; Nogueira J. M. Goussevskaia O.; Machado, M. V.; Mini R. A. F.; Loureiro. Data dissemination based on the energy map. *IEEE Communications Magazine*, pages 134–143, July 2005.
- [31] Hamid R. Arabnia, Laurence Tianruo Yang, and Chi-Hsiang Yeh. Qos support in wireless sensor networks: A survey. *International Conference on Wireless Networks (ICWN 2004)*, pages 227–233, June 2004.
- [32] S. Brim, B. Carpenter, and F. Le Faucheur. Per hop behavior identification codes. IETF, RFC 1633, May 2000.
- [33] R. Iyer and L. Kleinrock. Qos control for sensor networks. *International Conference on Communications (ICC 2003)*, pages 517–521, May 2003.
- [34] Samba Sesay, Zongkai Yang, and Jianhua He. A survey on mobile ad hoc wireless network. *Information Technology Journal*, pages 168–175, May 2004.
- [35] Braden R., Clark D., and Shenker S. Integrated services in the internet architecture: an overview. IETF, RFC 1633, June 1994.
- [36] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. RFC 2475: An architecture for differentiated services, December 1998. Status: PROPOSED STANDARD.
- [37] S. Ratnasamy et al. A scalable content-addressable network. *SIGCOMM*, pages 161–172, August 2001.
- [38] I. Stoica et al. Chord: A scalable peer-to-peer lookup service for internet applications. *Proc. ACM SIGCOMM*, pages 149–160, August 2001.

- [39] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. *International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, November 2001.
- [40] D. Malki, M. Naor, and D. Ratajczak. Viceroy: A scalable and dynamic emulation of the butterfly. *Principles of Distributed Computing*, pages 183–192, July 2002.
- [41] Alexander Klemm, Christoph Lindemann, and Oliver P. Waldhorst. A special purpose peer-to-peer file sharing system for mobile ad hoc networks. *Semiannual Vehicular Technology Conference (VTC2003-Fall)*, pages 41–49, October 2003.
- [42] S. Chen, K. Nahrstedt, and Y. Shavitt. A qos aware multicast routing protocol. *IEEE INFOCOM 2000*, pages 1594–1603, March 2000.
- [43] V.P. Kompella, J.C. Pasquale, and G.C. Polyzos. Multicast routing for multimedia communication. *Transactions on Networking*, pages 286–292, June 1993.
- [44] Q. Zhu, M. Parsa, and J.J. Garcia-Luma-Aceves. A source based algorithm for delay constrained minimum-cost multi-casting. *INFOCOM*, April 1995.
- [45] R. Beyah, R. Sivakumar, and J. Copeland. Application layer switching: A deployable technique for providing quality of service. *Global Telecommunications Conference (GLOBECOM 2003)*, pages 3231–3236, December 2003.
- [46] Naouel Ben Salem, Levente Buttyán, and Jean-Pierre Hubaux. Self-organizing packet radio ad hoc networks with overlay (soprano). *IEEE Communications Magazine*, pages 149–157, June 2002.
- [47] T. Chen, J. Tsai, and M. Gerla. Qos routing performance in multihop multimedia wireless networks. *Sixth International Conference of Universal Personal Communications*, pages 557–561, October 1997.
- [48] K. Sohrabi et al. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, pages 16–27, October 2000.
- [49] Daniel F. Macedo, Leonardo B. Oliveira, and Antonio A.F. Loureiro. Integrating overlay networks in wireless sensor networks. *Wireless Communication and Mobile Computing Workshop (WCSF'03)*, pages 190–198, October 2003.

- [50] Sonia Waharte, Jin Xiao, and Raouf Boutaba. Overlay wireless sensor networks for application-adaptive scheduling in wlan. *7th IEEE International Conference, HSNMC 2004*, pages 676–684, June 2004.
- [51] Grace P., Coulson G., Blair G., Mathy L., Yeung W.K., Cai W, Duce D., and Cooper C. Gridkit: Pluggable overlay networks for grid computing. *Distributed Objects and Applications (DOA'04)*, pages 1463–1481, Oct 2004.
- [52] T. He, J. Stankovic, C. Lu, and T. Abdelzaher. Speed: A stateless protocol for real-time communication in sensor networks. *International Conference on Distributed Computing Systems (ICDCS 2003)*, pages 46–55, May 2003.
- [53] T. Chen, J. Tsai, and M. Gerla. Qos routing performance in multihop multimedia wireless networks. *Sixth International Conference of Universal Personal Communications*, pages 557–561, October 1997.
- [54] Leonard B.A. and S. Takuo. A genetic algorithm (ga) based routing method for mobile ad-hoc networks. *Journal of Interconnection Networks*, pages 257–270, September 2003.
- [55] B. Hughes and V. Cahill. Achieving real-time guarantees in mobile ad hoc wireless networks. *24th Real-Time Systems Symposium*, December 2003. Work in Progress.
- [56] B. Deb, S. Bhatnagar, and B. Nath. Reinform: Reliable information forwarding using multiple paths in sensor networks. *International Conference on Local Computer Networks*, pages 406–415, October 2003.
- [57] F. Stann and J. Heidemann. Rmst: Reliable data transport in sensor networks*. *International Workshop on Sensor Network Protocols and Applications*, pages 102–112, May 2003.
- [58] Jamal N. Al-Karaki and Ahmed E. Kamal. Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communication*, pages 6–28, December 2004.
- [59] Macedo D. F., Correia L. H. A., Santos A. L., and Loureiro A. A. F. Evaluating fault tolerance aspects in routing protocols for wireless sensor networks. *Fourth Annual Mediterranean Ad Hoc Networking Workshop*, June 2005.

-
- [60] O.Younis, S.Fahmy, and P.Santi. Robust communications for sensor networks in hostile environments. *International Workshop on Quality of Service (IWQoS 2004)*, pages 10–19, June 2004.
- [61] K. Pahlavan and A. Levesque. *Wireless Information Networks*. Wiley-Interscience, 1995.
- [62] L. Hu. Distributed code assignments for cdma packet radio networks. *IEEE ACM Transactions on Networking*, pages 668–677, December 1993.
- [63] W.Heinzelman, A.Chandrakasan, and H.Balakrishnan. uamps ns code extensions. <http://www-mtl.mit.edu/research/icsystems/uamps/leach>.
- [64] Steven McCanne and Sally Floyd. ns Network Simulator. <http://www.isi.edu/nsnam/ns/>.
- [65] Y. Amir and C. Danilov. Reliable communication in overlay networks. *International Conference on Dependable Systems and Networks (DSN 2003)*, pages 511–520, June 2003.
- [66] S. Slijepcevic, V. Tsiatsis, S. Zimbeck, M.B. Srivastava, and M. Potkonjak. On communication security in wireless ad-hoc sensor networks. *11th WETICE*, pages 139–144, June 2002.