

ROTEAMENTO COM BALANCEAMENTO DE CONSUMO DE ENERGIA PARA
REDES DE SENSORES SEM FIO

Ingrid Teixeira

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS
PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE
FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS
EM ENGENHARIA ELÉTRICA.

Aprovada por:

Prof. José Ferreira de Rezende, Dr.

Prof. Aloysio de Castro Pinto Pedroza, Dr.

Prof. Otto Carlos Muniz Bandeira Duarte, Dr. Ing.

Prof^a. Luci Pirmez, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

ABRIL DE 2005

TEIXEIRA, INGRID

Roteamento com balanceamento de consumo de energia para redes de sensores sem fio
[Rio de Janeiro] 2005

XV, 94 p. 29,7 cm (COPPE/UFRJ, M.Sc.,
Engenharia Elétrica, 2005)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Redes de sensores
2. Protocolo de roteamento
3. Consumo de energia em redes de sensores

I. COPPE/UFRJ II. Título (série)

Aos meus pais Ademar e Eloisa Elena.

Agradecimentos

A Deus, nosso Pai.

À minha família. Aos meu pai Ademar pelo amor e apoio durante toda a minha vida. À minha mãe Eloisa Elena por ter sido a minha primeira professora cujos ensinamentos de amor, honestidade e trabalho jamais serão esquecidos. Eles me mostraram que as mais belas flores surgem após o mais rigoroso frio. À minha irmã Illene pelo amor e conselhos amigos. Ao meu noivo Paulo David pelas idéias discutidas sobre o projeto e por cuidar de mim com tanto carinho e amor. Obrigada por compartilharem comigo todos os momentos da vida.

Aos meus companheiros do Grupo de Teleinformática e Automação pela convivência harmoniosa. A Andréa, por ser uma amiga atenciosa e companheira nos diversos encontros *intelectuais* em pizzarias e etc, essenciais para nos manter em estado são durante os anos de mestrado. Aos amigos Ivana, Luiz Gustavo e Luiz Antônio pelas risadas e companhia. Aos amigos Antônio e Jorgito, pelas idéias trocadas durante as diversas fases dessa tese. Aos amigos Myrna e Glauco que ofereceram muito mais que sua casa, nosso QG de estudos nas manhãs de domingo, ofereceram apoio, amizade e alegria, sem os quais esse trabalho não estaria concluído. À minha amiga Lilian presente em todos os momentos da minha vida. Aos meus companheiros da Accenture do Brasil Arton, Carlos Lima e Luciana. Enfim, a todos os meus amigos que muitas vezes estão invisíveis para o mundo, mas presentes em meus pensamentos.

Aos professores que tive durante toda a minha vida acadêmica, pois eles foram os semeadores de idéias que contribuiram para o meu desenvolvimento intelectual. Aos professores da COPPE/UFRJ, pelo profissionalismo e competência. Aos professores convidados Otto e Lucy por terem avaliado o trabalho desenvolvido. Aos meus professores orientadores Rezende e Aloysio por iluminarem a minha pesquisa.

Ao CNPq, CAPES, COFECUB, FAPERJ pelos recursos utilizados durante a pesquisa.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

ROTEAMENTO COM BALANCEAMENTO DE CONSUMO DE ENERGIA PARA
REDES DE SENSORES SEM FIO

Ingrid Teixeira

Abril/2005

Orientadores: Aloysio de Castro Pinto Pedroza

José Ferreira de Rezende

Programa: Engenharia Elétrica

Os avanços nas áreas de tecnologias sem fio e microeletrônica permitem o monitoramento remoto de uma região utilizando uma rede formada por microsensores. Essas aplicações sofrem restrições de recursos severas e a energia disponível é escassa. Para determinadas aplicações em rede de sensores sem fio todos os nós desempenham funções equivalentes, devendo, portanto, ter suas vidas úteis prolongadas. Como o intuito de aumentar o tempo de vida da rede é apresentada nessa dissertação uma proposta de protocolo de roteamento para as redes de sensores sem fio. É utilizada a energia residual dos nós como métrica de escolha de rotas. A proposta é implementada como uma modificação do protocolo de difusão direcionada, que é centrado em dados e utiliza o endereçamento baseado em atributos, facilitando assim o processamento distribuído na rede. Ao protocolo original, foi adicionado um módulo de software, denominado dentro da arquitetura um filtro. O filtro Energia proposto é instanciado em todos os nós da rede e age sobre as mensagens de dados exploratórias. As mensagens exploratórias são enviadas periodicamente pelos nós sensores que são fontes da informação e servem para, dentre outras funções, estabelecer as rotas. A avaliação da proposta por meio de simulação demonstrou que a utilização da energia residual dos nós como métrica de roteamento apresenta resultados positivos para aumentar o tempo de vida da rede.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

ROUTING WITH ENERGY CONSUMPTION BALANCING FOR WIRELESS
SENSOR NETWORKS

Ingrid Teixeira

April/2005

Advisors: Aloysio de Castro Pinto Pedroza

José Ferreira de Rezende

Department: Electrical Engineering

Recently, advances in wireless technologies and microelectronics have enabled exciting new applications for wireless devices. These applications span a wide range including remote monitoring and home networking of appliances. Nevertheless, these applications have severe resources constraints that do not appear in wired computing environments. Sensor nodes are usually battery-operated and energy is scarce. In order to provide a longer network lifetime this thesis proposes a routing protocol to wireless sensor networks. The protocol uses nodes' residual energy as a metric to discover new routes. The work developed modifies the original behaviour of directed diffusion protocol. Directed diffusion protocol was selected because of its data-centric support and attribute-based addressing scheme, that makes distributed network processing easier. A software module, named filter in directed diffusion architecture, was developed. The Energy filter proposed is instantiated in every sensor node and process the exploratory data. Exploratory data are periodically sent by the sensor nodes that have data about the phenomenon of interest. These are the data used to establish routes. The proposal evaluated by simulation has shown that the use of residual nodes energy provides a good metric to extend sensor network lifetime.

Sumário

Resumo	v
Abstract	vi
Lista de figuras	xi
Lista de tabelas	xiv
Lista de acrônimos	xv
1 Introdução	1
1.1 A importância do consumo de energia nas redes de sensores	2
1.2 Trabalhos Relacionados	3
1.3 Organização da Dissertação	5
2 Conceitos Básicos	6
2.1 Introdução	6
2.2 Redes móveis sem fio	8
2.3 Redes de Sensores Sem Fio	11
2.3.1 Projetos em Redes de Sensores Sem Fio	13

2.3.2	Características das Redes de Sensores Sem Fio	14
2.4	Pilha de protocolos para as Rede de Sensores Sem Fio	21
2.4.1	Camada Física	22
2.4.2	Camada de Enlace	23
2.4.3	Camada de Rede	26
	Protocolo de difusão direcionada	30
2.4.4	Camada Transporte	34
2.5	Comentários	34
3	Protocolo de Roteamento com Métrica de Energia	36
3.1	Introdução	36
3.2	Extensão ao protocolo difusão direcionada	37
3.2.1	Modelagem	37
3.2.2	Implementação	39
3.2.3	Configuração do Filtro	41
3.3	Comentários	45
4	Simulação e Análise	47
4.1	Ambiente de simulação	47
4.2	Cenários e Métricas utilizadas	49
4.3	Cenário 1 : Grade com 9 nós	49
4.3.1	Análise do protocolo de roteamento para a difusão direcionada . .	50
	Métrica 1: Análise das árvores de roteamento	51
	Métrica 2: Número de transmissões	51

Métrica 3: Energia residual de cada nó	51
Métrica 4: Energia residual total da rede por nó	52
Métrica 5: Consumo total da rede	53
Métrica 6: Custo do recebimento das mensagens	53
Métrica 7: Tempo de vida da rede	54
4.3.2 Protocolo de Difusão Direcionada com o filtro Energia	54
Métrica 1: Análise das árvores de roteamento	55
Métrica 2: Número de transmissões	56
Métrica 3: Energia residual de cada nó	56
Métrica 4: Energia residual total da rede por nó	61
Métrica 5: Consumo total da rede	61
Métrica 6: Custo do recebimento das mensagens	62
Métrica 7: Tempo de vida da rede	64
4.4 Cenário 2 : Grades com maior número de nós	64
4.4.1 Comparativo entre o protocolo original e o Filtro Energia	66
Métrica 1: Número de transmissões	66
Métrica 2: Energia residual de cada nó	66
Métrica 3: Energia residual total da rede por nó	67
Métrica 4: Custo do recebimento das mensagens	68
Métrica 5: Tempo de vida da rede	68
4.5 Comentários	69

<i>SUMÁRIO</i>	x
5.1 Conclusões	77
5.2 Sugestões para Trabalhos Futuros	79
Referências Bibliográficas	80
A O Filtro Energia	85

Lista de Figuras

2.1	Exemplo de uma rede de sensores sem fio com gateway.	7
2.2	Tipos de redes de comunicação.	10
2.3	Estabelecimento de uma rede de sensores sem fio.	11
2.4	Projetos em RSSF	15
2.5	Componentes de um nó sensor.	18
2.6	Um esquema simplificado para o difusão direcionada	30
2.7	Esquema de nomes	31
2.8	Aspectos da difusão direcionada	33
3.1	Interação do filtro Energy com o núcleo do protocolo	40
3.2	Estabelecimento das rotas com temporizador constante	42
3.3	Propagação das mensagens de interesse	44
3.4	Propagação das mensagens exploratórias	45
3.5	Propagação dos dados	45
4.1	Topologia em Grid 3x3.	50
4.2	Difusão pura: Energia residual dos nós.	52
4.3	Difusão Pura: Energia residual final por nó.	53

4.4	Difusão Pura: Energia residual da rede.	54
4.5	Filtro Energia: Tempo de descoberta das rotas.	56
4.6	Filtro Energia: Número de transmissões de dados.	57
4.7	Filtro Energia: Energia residual dos nós em 51s.	58
4.8	Filtro Energia: Energia residual dos nós em 891s.	59
4.9	Filtro Energia: Energia residual dos nós em 591s.	60
4.10	Filtro Energia: Energia residual por nó (temporizador 200ms).	61
4.11	Filtro Energia: Energia residual da rede.	62
4.12	Filtro Energia x Difusão pura: Energia residual da rede.	63
4.13	Filtro Energia: Quantidade de mensagens de dados recebidas.	63
4.14	Filtro Energia: Custo do recebimento das mensagens.	64
4.15	Filtro Energia: Tempo de vida da rede.	65
4.16	Topologia em Grid 10x10.	66
4.17	Comparativo Difusão pura x Filtro Energia: Número de transmissões de dados.	67
4.18	Energia residual total da rede por nó (tempo de simulação = 51s).	70
4.19	Energia residual total da rede por nó (tempo de simulação = 51s).	71
4.20	Energia residual total da rede por nó - Grades 10x10 e 12x12.	72
4.21	Energia residual total da rede por nó - Grades 14x14 e 16x16.	73
4.22	Comparativo Difusão pura x Filtro Energia: Número de mensagens recebidas.	74
4.23	Comparativo Difusão pura x Filtro Energia: Custo do recebimento das mensagens.	75

4.24 Tempo de vida da rede. 76

Lista de Tabelas

2.1	Consumo de corrente do rádio CC1000 - CrossBow.	20
2.2	Consumo de energia de alguns dispositivos sensores.	21
2.3	Características do processador - MICA2.	21
3.1	Tabela descritiva do comportamento do filtro Energia	41
4.1	Parâmetros das simulações.	50
4.2	Difusão pura: Análise das árvores descobertas.	51
4.3	Filtro Energia: Análise das rotas descobertas.	55

Lista de acrônimos

API :	<i>Application Program Interface;</i>
CDMA :	<i>Code Division Multiple Access;</i>
CMOS :	<i>Complementary Metal Oxide Semiconductor;</i>
EAR :	<i>Energy Aware Routing;</i>
ED :	<i>Exploratory Data;</i>
GPS :	<i>Global Positioning System;</i>
IEEE :	<i>Institute of Electrical and Electronics Engineers;</i>
ISM :	<i>Industrial, Scientific and Medical;</i>
LEACH :	<i>Low Energy Adaptative Clustering Hierarchy;</i>
LWIM :	<i>Low power Integrated Microsensors;</i>
MAC :	<i>Medium Access Control;</i>
MANET :	<i>Mobile Ad Hoc Network;</i>
MEMS :	<i>Micro Electro Mechanical Systems;</i>
NS-2 :	<i>Network Simulator-2;</i>
PR :	<i>Positive Reinforcement;</i>
RSSF :	<i>Rede de Sensores Sem Fio;</i>
SCADDS :	<i>Scalable Coordination Architectures for Deeply Distributed Systems;</i>
SMAC :	<i>Sensor Medium Access Control;</i>
SPIN :	<i>Sensor Protocols for Information via Negotiation;</i>
TDMA :	<i>Time Division Multiple Access;</i>
VINT :	<i>Virtual Internetwork Testbed;</i>
WINS :	<i>Wireless Integrated Network Sensors;</i>

Capítulo 1

Introdução

OS avanços na indústria eletrônica permitem a integração de elementos mecânicos, sensores, atuadores e eletrônicos em um substrato único de silício. Através da tecnologia de microfabricação, os MEMS (*Micro-Electro-Mechanical Systems*) são construídos. Nesse processo, são selecionadas partes da superfície do silício e adicionadas novas camadas estruturais. Dessa forma, um único dispositivo possui capacidade de sensoriamento, armazenamento e processamento de dados e comunicação via rádio.

As infra-estruturas computacionais emergentes capazes de sensoriar e reagir a ambientes físicos sugerem um futuro onde o papel principal da tecnologia sem fio será de aumentar a interação das pessoas com o mundo físico. O ponto principal nessa evolução não é somente a capacidade de comunicação que os dispositivos embarcados terão, mas o fato de estarem realmente interconectados. Por questões de conveniência, esses dispositivos serão pequenos e o custo esperado para o usuário final será de alguns dólares, conforme o projeto desenvolvido na Universidade de Berkley [1].

As redes de sensores sem fio (RSSFs) estão situadas na nova geração de computação pessoal, a era da computação ubíqua [2]. Nessa nova era, os dispositivos estarão conectados entre si, permitindo diversas aplicações nas áreas militares e civis [3]. Muitas empresas de tecnologia, apostam que as rede de sensores sem fio possuem o potencial de aumentar o desempenho e transformar o modo como as empresas fazem negócio [4], [5].

Os pequenos dispositivos sensores são poderosos, e, na medida do possível, são auto-

configuráveis e autônomos. Para serem portáteis eles devem interagir por enlaces sem fio, formando um sistema em rede. Existem algumas diferenças entre as redes de sensores e as redes multimídias sem fio. Dentre outras características, pode-se citar que as distâncias de transmissões são menores (de 10 a 20 m), assim como as taxas de transmissão de dados (<kbps). Em uma RSSFs, existe a possibilidade de colaboração entre os nós para a geração da informação solicitada pelo usuário. Entretanto, a característica mais marcante é a capacidade restrita da fonte de energia dos dispositivos.

1.1 A importância do consumo de energia nas redes de sensores

Um dos mais importantes desafios para o projeto das redes de sensores sem fio é o consumo de energia, ao invés da complexidade na integração. A fonte de energia é freqüentemente limitada às baterias internas, pois a operação dessas redes será não assistida para permitir mobilidade e portabilidade dos nós. Infelizmente, a tecnologia das baterias não está progredindo no mesmo ritmo da tecnologia de processamento do silício, mas é esperado que haja um acréscimo de 20% da capacidade para os próximos anos [6]. Os esforços recentes da pesquisa estão deslocando a ênfase no projeto dos sistemas que aumentem tão somente a capacidade. Ao invés, o consumo de energia e potência estão se tornando restrições importantes e estão sendo estudadas em vários níveis, da arquitetura ao *hardware*, dos protocolos aos algoritmos [7].

Os projetos dos circuitos eletrônicos levam em consideração que a rede possui baixa taxa de transmissão. Com isso, os circuitos operam com baixo ciclo de trabalho, ou seja, alternando entre os estados ativo e inativo para conservar a energia. Os transceptores comerciais de rádio são inadequados para as aplicações em RSSFs devido à elevada sobrecarga de energia empregada para ligá-los e desligá-los. São necessários, então, o projeto de circuitos específicos e com baixo consumo de energia. A necessidade de colaboração requer o desenvolvimento de esquemas de agregação de dados de modo a minimizar a dissipação de energia [8]. Uma solução para repor a energia consumida é a utilização de técnicas de conversão de energia ambiente (solar ou vibração) em energia elétrica.

O consumo de energia pode ser dividido em três domínios: sensoriamento, comunicação e processamento dos dados. A unidade de sensoriamento e seus componentes dependem da complexidade da detecção do evento. Aumentos nos níveis de ruído ambiente podem elevar a complexidade de detecção. A energia gasta para o processamento dos dados é muito menor que a energia necessária para a comunicação. O exemplo mostrado por Pottie e Kaiser [9] ilustra essa diferença. Considerando uma interferência tipo Rayleigh e uma perda de potência elevada a quarta potência com relação à distância, o custo em energia para se transmitir 1kB por uma distância de 100m é aproximadamente o mesmo que executar 3 milhões de instruções em um processador com capacidade de processamento de 100 MIPS/W. Dessa forma, o processamento local e distribuído, ao invés de centralizado, é crucial para minimizar o consumo de potência em uma rede de sensores com múltiplos saltos.

O objetivo deste trabalho é de estudar as técnicas de roteamento existentes para redes *ad-hoc* e de sensores e implementar uma que considere a energia dos nós como métrica para o roteamento dos dados, ou seja o roteamento é dito *energy-aware*. O protocolo de roteamento proposto utiliza a quantidade de energia residual dos nós para distribuir o fluxo dos dados pela rede. Em algumas aplicações em RSSF, como monitoramento de variáveis ambientais ou humanas, o atraso do pacote não é a métrica principal para o roteamento, o melhor caminho é aquele que maximiza a energia residual dos nós que compõem as rotas. O ganho imediato é a distribuição do consumo de energia pelos nós. Com isso, estende-se o tempo de vida da rede, evitando intervenção humana e reduzindo os custos operacionais de manutenção da rede.

1.2 Trabalhos Relacionados

Existem diversos trabalhos sobre roteamento na literatura de redes *ad-hoc*. Entretanto, os trabalhos e seus resultados não podem ser diretamente implementados nas redes de sensores. Em primeiro lugar, o número de nós que compõem uma rede de sensores pode ser algumas ordens de grandeza maior que o número de nós em uma rede *ad-hoc*. Além disso, os nós sensores são mais limitados em termos de potência, computação e memória

e possuem maior suscetibilidade a falhas. Na maioria das vezes, os nós se comunicam por *broadcast* (difusão), enquanto que na rede *ad-hoc* é utilizado o paradigma ponto-a-ponto. Em uma rede de sensores, os nós não possuem um identificador global, devido ao elevado número de nós e a sobrecarga de mensagens necessárias para a atribuição de um identificador distinto para cada nó. Dessa forma, desenvolveram-se alguns grupos de pesquisa exclusivamente voltados para a análise e propostas de soluções para as RSSFs.

Chang e Tassiulas [10] propuseram um algoritmo para maximizar o tempo de vida da rede quando a taxa de envio de mensagens é conhecida. A idéia proposta é de evitar a utilização de nós com baixa potência e escolher o menor caminho no início. O algoritmo redireciona o fluxo de pacotes entre os nós de modo a balancear o consumo entre os nós em proporção à energia disponível. As desvantagens desse trabalho são a dependência do conhecimento prévio da taxa de geração da informação (fluxo de dados), o conjunto de nós que compõem a origem da informação e o conjunto de nós destino.

Em Gupta e Kumar é discutido a potência limite que o nó deve utilizar para transmitir os dados de modo a garantir que a rede estará conectada. Wu et al. [11] propõem uma abordagem centrada na energia baseado em um conjunto de nós dominantes. Um conjunto é dominante se todos os nós do sistema ou estão no conjunto ou seus nós vizinhos estão. Em geral, os nós pertencentes ao conjunto dominante consomem mais energia, pois o tráfego dos outros nós é encaminhado por eles. A idéia é usar regras baseadas no nível de energia para prolongar o tempo de vida de um nó no processo de escolha dos conjuntos dominantes.

Um outro conjunto de trabalhos estudam a otimização do consumo de energia durante o tempo em que os nós ficam em espera (em *idle time*). Esses protocolos colocam determinados nós no modo *sleep* (desligado) para conservar a energia, enquanto mantém a conectividade da rede garantindo a comunicação. O Span [12] é um algoritmo distribuído, onde os nós realizam decisões locais para determinar se entram no modo *sleep* ou formam a coluna vertebral da rede como coordenadores. Os coordenadores permanecem ativos durante o tempo necessário para encaminhar os pacotes salto a salto. A decisão para determinar se o nó tornará coordenador baseia-se em dois fatores: o número de vizinhos que se beneficiarão se o nó ficar ativo, e a quantidade de energia restante no nó.

O trabalho apresentado por Shah e Rabey em [13] inspirou a proposta dessa dissertação. Um custo de energia é calculado para os possíveis caminhos para o roteamento e uma probabilidade inversamente proporcional a esse custo é atribuída a cada nó. Esse mecanismo pode ser classificado como múltiplo caminho, uma vez que é construída e mantida uma tabela de roteamento com várias entradas. Entretanto, protocolos de múltiplos caminhos podem não se adequar a redes compostas por muitos nós e a proposta de tabela de roteamento pode não ser capaz de se adaptar com baixo custo à dinâmica da rede e do tráfego presentes nas redes de sensores.

O objetivo desse trabalho, é o de apresentar uma proposta de um mecanismo de distribuição do tráfego pela rede sem manter múltiplas entradas em tabelas de roteamento. Em cada nó, é armazenada na tabela de roteamento somente um vizinho para o qual será encaminhado o fluxo de dados. Esse vizinho é escolhido utilizando determinados critérios a cada rodada do protocolo.

As propostas de conservar a energia dos nós através da operação no modo de conservação de energia *sleep modes* são complementares aos resultados apresentados neste estudo. Combinados, modos eficazes de tratar o tempo ocioso dos nós com a comunicação podem levar a soluções de gerenciamento de energia mais poderosas.

1.3 Organização da Dissertação

Esta dissertação está estruturada em cinco capítulos. O capítulo 2 descreve os conceitos básicos sobre redes de sensores sem fio, os projetos de pesquisas atuais, as características dos nós sensores e a sua pilha de protocolos, especialmente a camada de rede. No capítulo 3 é apresentado o mecanismo proposto para a seleção de rotas privilegiando os caminhos com maior energia residual. O capítulo 4 apresenta as simulações realizadas para verificar o mecanismo e os resultados alcançados. Finalmente, o capítulo 5 traz as conclusões e sugestões para trabalhos futuros.

Capítulo 2

Conceitos Básicos

ALGUNS conceitos sobre as redes de sensores sem fio são apresentados nesse capítulo. Inicialmente são feitas algumas definições da terminologia utilizada no texto. Em seguida, nas seções 2.2 e 2.3, são mostradas as principais diferenças entre as redes móveis e as RSSFs. A seção 2.3.1 destaca os alguns projetos atuais nas universidades. A seção 2.3.2 mostra as unidades básicas que formam os nós sensores, mostrando a contribuição de cada unidade no consumo de energia de um nó. As camadas que formam a pilha de protocolos nas RSSFs são mostradas na seção 2.4. Na camada de rede, seção 2.4.3, onde são mostrados alguns trabalhos relacionados à área de pesquisa em roteamento de dados em redes de sensores. Nessa seção também é descrito o mecanismo de funcionamento do protocolo original de difusão direcionada.

2.1 Introdução

OS sensores estão ficando cada vez menores, mais baratos e com isso aumentando abundantemente em número. As projeções feitas pela indústria eletrônica é de que esses dispositivos custem no máximo 1 dólar americano. Esses dispositivos pequenos e baratos teriam a capacidade de se comunicarem por interfaces sem fio, sem assistência humana direta e atuarem de modo colaborativo. Os avanços conseguidos se devem ao desenvolvimento do MEMS (*Micro-Electro-Mechanical Systems*), que é a integração de

elementos mecânicos, sensores, atuadores e eletrônica em um substrato único de silício através da tecnologia de microfabricação.

Redes de sensores partilham muitos dos desafios de redes sem fio ad-hoc, incluindo capacidade de energia para cada nó e largura de banda limitadas e canais ruidosos. Entretanto, a comunicação em redes de sensores sem fio difere da comunicação em outros tipos de redes por não ser uma comunicação fim-a-fim. Mais especificamente, a função da rede é de reportar a informação a respeito do fenômeno para o observador, que não necessariamente está ciente da infra-estrutura da rede de sensor e dos sensores individualmente, ilustrado na figura 2.1. Além disso, a energia é mais limitada nas redes de sensores do que em outras redes sem fio devido à natureza dos dispositivos sensores e da dificuldade em recarregar as baterias.

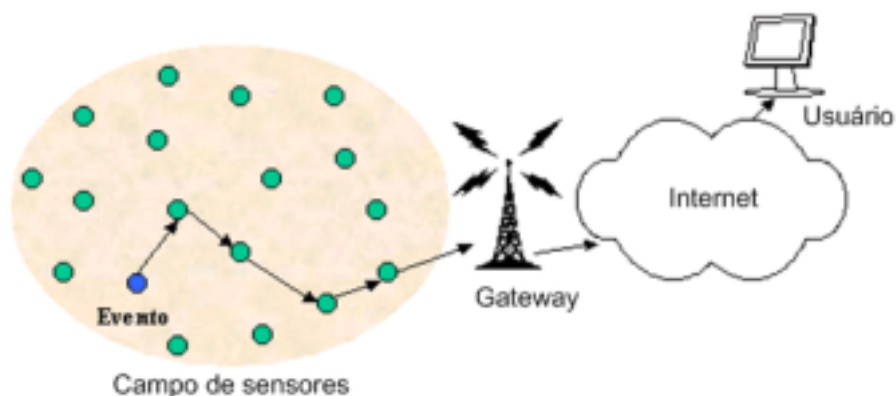


Figura 2.1: Exemplo de uma rede de sensores sem fio com gateway.

Nesta dissertação utilizar-se-á a terminologia descrita a seguir.

- **Sensor:** é o dispositivo que implementa o sensoriamento físico do fenômeno ambiental e reporta as medições (através da comunicação sem fio). Tipicamente é constituído de cinco componentes - hardware para sensoriamento, memória, bateria, processador e transceptor;
- **Observador:** é o usuário final interessado em obter a informação disseminada pela rede de sensor sobre o fenômeno. O observador informa a rede qual é seu interesse, através do envio de perguntas, e aguarda as respostas. Podem existir múltiplos observadores na rede de sensores;

- Sorvedouro: é o dispositivo que coleta as informações geradas pela rede de sensores e disponibiliza para o usuário. O sorvedouro pode ser, por exemplo, uma estação base;
- Fenômeno: é a entidade de interesse do observador que está sendo monitorado/analísado. Múltiplos fenômenos podem estar sendo monitorados pela mesma rede de sensores. Para tanto, devem haver diversos tipos de sensores disponíveis na rede.

Em uma aplicação de sensoriamento, o observador está interessado em monitorar o comportamento de um fenômeno. Em uma rede de sensores típica, os sensores individuais coletam valores locais (medições) e disseminam a informação ou para outros sensores ou diretamente para o observador. As medidas coletadas pelos sensores são amostras discretas do fenômeno físico sujeitas à acurácia da medida assim como da localização com relação ao fenômeno.

Neste capítulo, serão apresentados alguns conceitos básicos de redes móveis sem fio, na seção 2.2, e os conceitos básicos para as Redes de Sensores Sem Fio (RSSF), na seção 2.3. Serão descritos alguns dos projetos atuais de pesquisa em desenvolvimento pelas universidades e centros de pesquisas em 2.3.1, permitindo uma maior visibilidade da área.

Na seção 2.3.2 são apresentados os modelos de consumo de energia da bateria e alguns exemplos de consumo de energia dos demais dispositivos que compõem o nó sensor: processador, transmissor e o elemento de sensoriamento. Em 2.4 são apresentados os protocolos propostos para as RSSF, destacando-se a camada de rede. Na seção 2.5 são feitos alguns comentários a respeito dos desafios presentes na área de pesquisa em protocolos de roteamento para as redes de sensores sem fio.

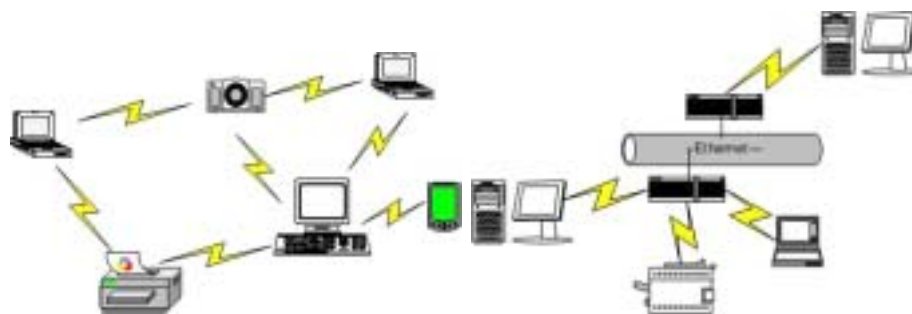
2.2 Redes móveis sem fio

Uma rede é dita móvel se os elementos de rede que a compõem se locomovem durante a operação da rede. Geralmente os nós se comunicam por interfaces aéreas para garantir maior flexibilidade. As redes móveis possuem uma estética melhor, pois eliminam os fios,

geralmente tem a redução de custos de infra-estrutura e tempo de instalação nos projetos. Por outro lado, tem-se uma baixa qualidade de serviço (QoS) e o custo dos equipamentos ainda está elevado, embora a tendência seja de queda de preços com a minituarização dos componentes. O ambiente de redes móveis apresenta restrição do uso de frequências devido à regulamentação governamental. O suprimento de energia dos nós é restrito e os nós possuem baixa potência para a transmissão, existindo a preocupação com a segurança da informação. Os pré-requisitos para o projeto de redes móveis sem fio são a operação global e sem licença, baixa potência de transmissão, robustez, interoperabilidade com redes existentes, segurança e transparência para aplicações.

Existem dois tipos de redes móveis: estruturadas ou não-estruturadas, neste caso chamadas de *ad-hoc*. Nas redes de comunicações infra-estruturada, uma parte da rede é fixa, garantindo a infra-estrutura necessária para acesso das estações móveis e para interconexão com redes externas. Este é o caso da telefonia celular e redes locais sem fio interligadas com redes padrão ethernet. O equipamento que interliga a parte fixa com a móvel da rede é chamado de Ponto de Acesso (LAN sem fio) ou Estação Base (Sistema Celular). Todas as comunicações da rede são feitas entre um nó móvel e o ponto de acesso, simplificando o projeto. Entretanto, existe perda de flexibilidade quando os pontos de acesso são danificados ou tornam-se indisponíveis. Além disso, uma vez que vários nós móveis se comunicam com o mesmo ponto de acesso, mecanismos de controle devem ser implementados. Da mesma forma, quando um nó móvel se afasta do seu ponto de acesso e a comunicação se torna difícil, ele precisa achar outro ponto de acesso próximo da sua nova localização, transferir a comunicação para este novo ponto e continuar a comunicação sem interrupção ou perda de dados. Este processo é chamado *handoff* [14].

As MANET - *Mobile Ad-hoc NETWORK* [15] são redes de computadores que não precisam de uma infra-estrutura prévia existente para que a comunicação ocorra. Em uma MANET, os nós, que são por exemplo laptops, PDAs ou impressoras, estabelecem comunicação entre si sem a necessidade de um ponto centralizador, figura 2.2(a). Numa rede infra-estruturada, a comunicação entre os elementos computacionais é feita através de estações base de rádio, que constituem uma infra-estrutura de comunicação, como ilustrado na figura 2.2(b).



(a) Rede de comunicação sem infraestrutura.

(b) Rede de comunicação infra-estruturada.

Figura 2.2: Tipos de redes de comunicação.

A rede móvel ad-hoc é uma rede peer-to-peer que é usualmente compreendida de dezenas a centenas de nós comunicantes que são capazes de cobrir centenas de metros. Cada nó é visto como um dispositivo de informação pessoal, como um PDA (Personal Digital Assistant) com um rádio transceptor sofisticado. A MANET tem por objetivo formar e manter uma rede com múltiplos saltos capaz de transportar o tráfego multi-mídia entre os nós.

Em uma MANET, a rede é projetada para prover uma razão vazão-atraso em face da mobilidade dos nós. Embora os nós sejam dispositivos energizados por baterias, o consumo de energia no sistema é de importância secundária, uma vez que cada dispositivo está conectado a pessoas, e uma bateria descarregada pode ser substituída quando necessário for.

Nas maioria das redes *ad-hoc* estudadas até o presente, o número de dispositivos móveis é da ordem de dezena ou centena e a limitação de energia não é tão restritiva quanto é nas RSSF. Com a utilização de dispositivos microsensores [16] limitados em capacidade de comunicação, processamento, armazenamento e energia, temos a formação das redes de sensores sem fio, que são estudas neste trabalho.

2.3 Redes de Sensores Sem Fio

Rede de sensores é um sistema distribuído, composto de um grande número de nós e onde a comunicação é sem fio. O sensoriamento é feito de forma distribuída pela região de interesse e os nós colaboram entre si para realizar o monitoramento da região. Os nós sensores estão densamente dispostos dentro do fenômeno ou muito próximo a ele. O posicionamento dos nós sensores não precisa ser previamente calculado, permitindo a disposição aleatória em terrenos inacessíveis ou operações em áreas de resgate, conforme mostrado na figura 2.3. Por outro lado, as redes de sensores impõem a restrição de que os algoritmos sejam auto-organizáveis. Uma outra característica da diferença entre as redes de sensores e a MANET é o esforço cooperativo entre os nós sensores. Ao invés de mandarem dados brutos para os nós responsáveis pela coleção dos dados, os nós sensores utilizam seus processadores para realizar computações simples e transmitir somente os dados solicitados e parcialmente processados.

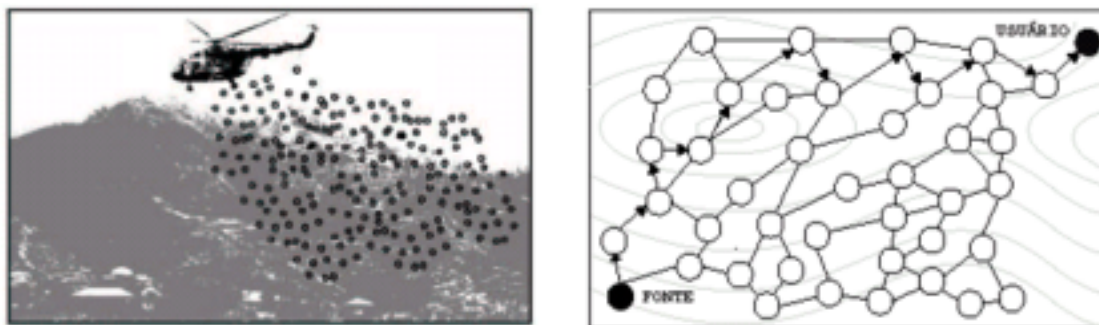


Figura 2.3: Estabelecimento de uma rede de sensores sem fio.

As RSSFs permitem diversas aplicações militares ou civis. Na área militar, as RSSFs permitem o rastreamento do campo inimigo remotamente e a defesa de uma região. Elas também podem ser utilizadas para detectar e caracterizar ataques químicos, biológicos, radiológicos, nucleares e explosivos. O lançamento dos dispositivos na área de monitoramento pode ser feito pelos veículos aéreos controlados por rádio (UAV - Unmanned Aerial Vehicles). As aplicações civis podem ser a construção de uma rede de monitoramento das condições ambientais, mudanças climáticas em oceanos e florestas. Na área industrial, pode-se controlar o maquinário das fábricas e o estoque de materiais. Na área comercial, as redes de sensores permitem a construção de espaços inteligentes [3] onde

os usuários podem não saber da existência dos dispositivos sensores.

Para permitir todas essas aplicações, a construção das RSSFs requer o desenvolvimento de técnicas de rede sem fio *ad-hoc*. Embora muitos protocolos e algoritmos tenham sido propostos para redes *ad-hoc* sem fio tradicionais, eles não são adequados às redes de sensores. Algumas das diferenças entre redes de sensores e as redes *ad-hoc* são enumeradas a seguir:

- o número de nós sensores em uma rede de sensores pode ser algumas ordens em magnitude superiores ao número de nós em uma rede *ad-hoc*;
- os nós sensores estão dispostos de forma mais densa na área de operação;
- os nós sensores possuem maior probabilidade de falhas;
- a topologia da rede de sensor varia frequentemente;
- os nós sensores utilizam o paradigma da comunicação por *broadcast* (difusão) enquanto as redes *ad-hoc* são baseadas na comunicação ponto-a-ponto;
- os nós sensores estão limitados em potência, capacidade computacional e em memória;
- não é adequado o uso de identificador global pelos sensores, devido à grande quantidade de overhead necessário na comunicação e de alto número de sensores.

O identificador único poderia ser atribuído aos nós antes do lançamento da rede, porém a topologia da rede de sensores é muito dinâmica. Para que a identificação única seja mantida, há a necessidade dos nós se comunicarem entre si, elevando o gasto de energia.

As características peculiares dos nós e da rede impõem decisões para a arquitetura. O tamanho pequeno dos dispositivos limita a capacidade de bateria requisitando que toda a operação seja feita de forma eficiente. Em [17] é mostrado um exemplo em que a comunicação consome 3J de energia para transmitir 1kb de dados a uma distância de 100m. Um processador de uso geral capaz de processar 100 milhões de instruções por segundo executaria 300 milhões de instruções com a mesma quantidade de energia.

A energia restrita impõe um limite no alcance de transmissão do rádio e sugere um esquema de transmissão em múltiplos saltos. Os nós devem ficar próximos, o que sugere processamento local da informação para reduzir a quantidade de informações redundantes a serem transmitidas.

2.3.1 Projetos em Redes de Sensores Sem Fio

Atualmente, existem diversos projetos de pesquisa em redes de sensores sem fio, conforme Akyildiz [18]. Dentre eles, destaca-se o projeto Factoid [19], desenvolvido pela *Compaq Palo Alto Western Research Laboratory* (WRL), que propõe um dispositivo portátil pequeno o suficiente para ser embutido em uma chave. Esse dispositivo coleta anúncios de outros dispositivos que se comunicam em *broadcast* no ambiente, e as informações podem ser levadas até a estação base do usuário. Nessa primeira geração, os protótipos ainda são maiores que os smart dust. O esquema de comunicação é feito por transmissão em RF em múltiplos saltos através de enlaces curtos.

O projeto WINS (*Wireless Integrated Network Sensors*) [16] na UCLA desenvolve dispositivos baseados em MEMS. Esse projeto iniciou em 1993 com o LPWIN (*Low Power Wireless Integrated Microsensors*). Além do sensor, estão presentes em um nó WINS o conversor de dados e a unidade de processamento de sinais e funções de controle. É utilizada a tecnologia CMOS para implementar a lógica digital que controla o sensor e realiza o processamento de sinais. Rádios de baixa potência possibilitam a construção da rede de baixa taxa e a pequenas distâncias.

O projeto Smart Dust [20] na Universidade da Califórnia, em Berkley, possibilitou a construção de uma rede formada por nós capazes de comunicação e sensoriamento, em uma escala milimétrica. A transmissão é feita passivamente usando refletores ópticos. A transmissão óptica ativa também é possível, porém consome maior potência e é utilizada quando não se tem uma visada direta entre o nó dust e a estação base.

O Ultra Low Power Wireless Sensor [21] é um projeto do MIT que desenvolveu dispositivos para sensoriamento com baixa potência que se comunicam entre si. O protótipo permite a transmissão de dados em uma variedade de taxas, desde 1 bit/seg até 1 me-

gabit/seg, com os níveis de transmissão variando de $10\mu\text{W}$ a 10mW . O sub-sistema de comunicação está sendo desenvolvido pela Analog Devices. A comunicação entre os nós é realizada em múltiplos saltos até o destino.

Outro projeto do MIT denominado μAMPS (*micro-Adaptive Multidomain Power-Aware Sensors*) [22] também desenvolve protótipos para a construção das redes de sensores. A pesquisa tem seu foco em técnicas de armazenamento de potência, comunicação e processamento de sinais. São utilizadas soluções programáveis de hardware para a implementação das aplicações de sensores.

No projeto PicoRadio do *Berkley Wireless Research Center* [23] [24] são realizadas pesquisas em várias áreas em redes de sensores sem fio: projeto de circuito RF, antenas e circuitos digitais com baixa voltagem, roteamento e posicionamento dos nós. Neste projeto é desenvolvido o PicoBeacon que é um dispositivo sensor operado somente por baterias solares.

O GTA/UFRJ (Grupo de Teleinformática e Automação) também possui um núcleo de estudos na área de RSSFs. Alguns trabalhos estão sendo desenvolvidos por alunos de mestrado e doutorado da COPPE Elétrica. As áreas atuais de pesquisa são protocolos de roteamento utilizando métricas de energia, protocolos de roteamento que favorecem a agregação de dados [8] e [25] e desenvolvimento de *middleware* inteligente para RSSFs [26], em parceria com o NCE/UFRJ.

2.3.2 Características das Redes de Sensores Sem Fio

O projeto das redes de sensores é influenciado por diversos fatores, que incluem : tolerância a falhas, escalabilidade, custo de produção, ambiente de operação, topologia da rede de sensores, meio de transmissão, consumo de potência, e restrições de hardware. Esses fatores estão diretamente relacionados com qualquer algoritmo ou solução proposta para as redes.

Tolerância a falhas: alguns nós sensores podem falhar ou serem bloqueados devido a falta de potência, danos físicos ou interferência do meio. Em ambientes hostis como campos de resgate, por exemplo, a probabilidade dos nós falharem é maior do que em

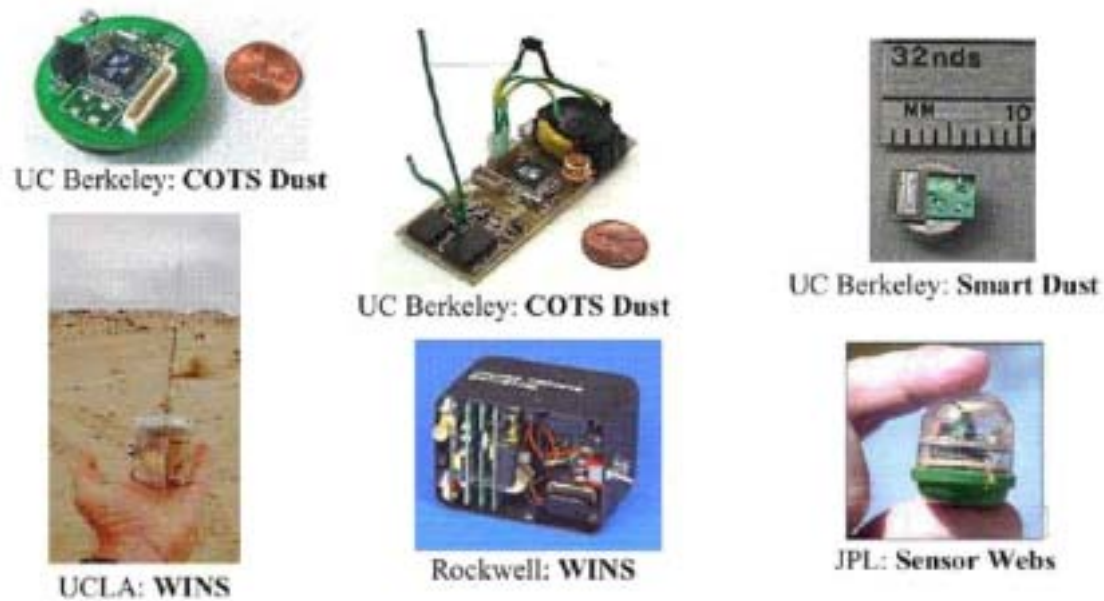


Figura 2.4: Projetos em RSSF

um ambiente doméstico. A falha de alguns desses nós não pode afetar a tarefa da rede de sensores. A tolerância a falhas é a capacidade em sustentar as funcionalidades de rede sem qualquer interrupção devido a falhas dos elementos da rede. O nível de tolerância a falhas está relacionado com a aplicação a ser suportada pela rede.

Escalabilidade: o número de nós sensores dispostos em operação na área de observação do fenômeno pode ser da ordem de centenas ou milhares. Dependendo da aplicação, o número pode chegar a milhões. Além disso, a densidade das redes de sensores pode variar de alguns nós a algumas centenas de nós em uma região, que pode ser menor que 10m em diâmetro.

Custo de produção: como a maioria das aplicações são projetadas para o monitoramento de áreas extensas será utilizada uma grande quantidade de nós sensores. Para que as aplicações sejam viáveis economicamente deve-se ter um baixo custo de produção. A indústria estima que o custo de um nó sensor seja em torno de US\$1.

Ambiente de operação: geralmente ambiente hostil e de difícil acesso. Essa característica impede a reposição das baterias dos nós. Além disso requer que a rede seja auto-organizável.

Topologia da rede de sensores: a rede é dinâmica porque os nós sensores possuem uma probabilidade grande de falharem e ficarem inoperantes.

Meio de transmissão: o mais utilizado é a transmissão por rádio frequência, mas também existe transmissão por meio ótico, infra-vermelho, conforme discutido em maior detalhe na seção 2.4.1.

Consumo de potência: os nós sensores carregam fontes de potência limitadas, geralmente insubstituíveis. Mesmo que o nó não esteja se comunicando, ou seja transmitindo ou recebendo os pacotes de dados, existe a dissipação estática dos circuitos digitais. Essa dissipação é originária do vazamento de corrente indesejável da fonte para o terra. Quando o nó entra em operação existe a energia de chaveamento para uma dada computação que é independente do tempo, enquanto que a energia desperdiçada é linear com o tempo. Apesar da energia de chaveamento ter historicamente excedido a energia de vazamento para as aplicações em CMOS, recentemente com o avanço dos processos semicondutores esse fato está se revertendo. A energia desperdiçada na operação de um circuito digital pode chegar a 50% do total da energia.

O consumo de energia do circuito digital do nó sensor pode ser dividido nas dissipações dinâmica e estática. A energia dinâmica da computação digital é a energia necessária para energizar os componentes do circuito integrado da voltagem zero (nível zero digital) para a voltagem de operação (nível um digital). A energia dinâmica é descrita por $CV_D D^2$, onde C representa a capacitância de chaveamento e $V_D D$, a voltagem da fonte.

A técnica mais antiga e direta de reduzir o consumo da energia que é desperdiçada em um sistema é desligar as partes que não estão em uso. O gerenciamento de potência baseado no desligamento foi proposto para discos rígidos, displays, e módulos de comunicação, dentre outros. Para processadores, foi incluído no kernel de Sistemas Operacionais de Tempo-Real (RTOS). Mais informações podem ser encontradas nos trabalhos publicados por Lorch e Smith [27], e Benini et al [28].

Existem algumas técnicas para economizar energia que desligam a parte de comunicação do sistema. Por exemplo, o rádio pode ser desligado usando-se um limiar de inatividade, e ligado novamente após um determinado período de tempo. Outros trabalhos

evitam a comunicação durante os períodos nos quais o canal de comunicação apresenta qualidade ruim, inadequada para uma transmissão de dados aceitável.

A redução do tamanho do sensor tem como consequência a redução no tamanho e na capacidade de seus componentes. Como existe uma grande diferença entre as capacidades de memória, processamento e tecnologias de fabricação de baterias há, consequentemente, diferentes consumos de energia. Deve-se, portanto, escolher o hardware que atenda aos requisitos da aplicação.

Os componentes presentes nos nós sensores dependem da aplicação, mas genericamente são compostos de quatro componentes básicos: unidade de sensoriamento, unidade de processamento, unidade de comunicação e unidade de potência, vide figura 2.5. A unidade de sensoriamento é geralmente composta de duas subunidades: os sensores e o conversor A/D. Os sinais analógicos produzidos pelos sensores baseados no fenômeno observado são convertidos para sinais digitais pelo conversor A/D, e então encaminhados a unidade de processamento. Quanto ao consumo de energia, a unidade de sensoriamento requer energia para a corrente de polarização, amplificação e filtragem analógica. A potência dissipada é relativamente constante quando ligado, e melhorias na sua eficiência de energia dependem do aumento da integração e projeto de circuitos analógicos eficazes.

A unidade de processamento dos nós é tipicamente utilizada para processamento de sinal digital dos dados coletados, implementação das camadas dos protocolos de comunicação e para o armazenamento de dados. Existe uma unidade transceptora que conecta o nó a rede. A unidade de potência é a responsável por prover toda a energia utilizada pelas demais unidades. A unidade de potência pode ser uma bateria química comum ou ser composta por células conversoras de energia ambiental como as células solares em energia elétrica. Podem existir nos nós sensores outras subunidades, que são dependentes da aplicação, como um mobilizador, necessário para mover o nó sensor, e o sistema de localização, que pode ser utilizado por técnicas de roteamento de rede baseado na posição.

A **unidade de potência** dos nós sensores é a bateria. Park et al. [29] propõem três tipos diferentes de modelos de bateria, mostrando como os diferentes aspectos do comportamento das baterias reais afetam a eficiência de energia das diferentes aplicações. A métrica utilizada é expressa em unidades de Ah (Ampere*Hora), que é um método co-

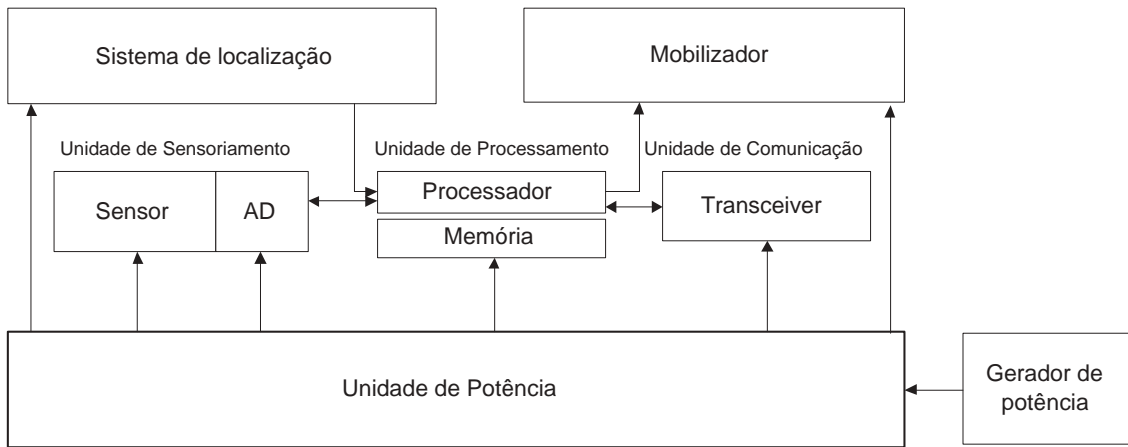


Figura 2.5: Componentes de um nó sensor.

um utilizado pelos fabricantes de baterias para especificar a capacidade teórica total da bateria. O cálculo do tempo de vida teórica da bateria pode ser calculado através da equação $T = \frac{C}{I}$, onde T é o tempo de vida da bateria, C é a capacidade da bateria em Ah e I é a corrente de descarga em A .

No modelo linear, a bateria é considerada como um armazenador linear de corrente. A capacidade máxima da bateria é atingida independentemente da taxa de descarga. A capacidade residual C_r depois do tempo de operação t_d pode ser expressa pela equação

$$C_r = C' - \int_{t=t_0}^{t_0+t_d} I(t) dt, \quad (2.1)$$

onde C' é a capacidade anterior da bateria em Ah e $I(t)$ é a corrente instantânea consumida em A pelo circuito no tempo t em horas.

Neste modelo $I(t)$ permanecerá a mesma durante o tempo t_d desde que o modo de operação permaneça o mesmo. Neste caso a equação pode tornar-se

$$C_r = C' - \int_{t=t_0}^{t_0+t_d} I(t) dt = C' - It \Big|_{t_0+t_d}^{t=t_0} = C' - It_d \quad (2.2)$$

Enquanto o modelo linear assume que a capacidade máxima da bateria não é afetada pela taxa de descarga, o Modelo Dependente da Taxa de Descarga considera o efeito da

taxa de descarga da bateria na capacidade máxima da mesma. Com o intuito de considerar tal efeito é introduzido um fator k que é o fator de eficiência da capacidade da bateria, que é determinado pela taxa de descarga. A definição de k é $k = \frac{C_{eff}}{C_{max}}$, onde C_{eff} é a capacidade efetiva da bateria e C_{max} é a capacidade máxima da bateria, ambas expressas em Ah . No Modelo Dependente da Taxa de Descarga, a Equação 2.1 é transformada em:

$$C_r = kC' - It_d \quad (2.3)$$

O fator de eficiência k varia com a corrente I e é próximo de 1(um) quando a taxa de descarga é baixa, mas se aproxima de 0(zero) quando a taxa de descarga se torna alta.

As baterias reais exibem o fenômeno nomeado *efeito de relaxação*, que pode ser brevemente explicado como a seguir. Quando a bateria é rapidamente descarregada, ocorre a difusão dos ingredientes ativos nos meios eletrolítico e no eletrodo. Se a taxa de descarga alta for mantida, a bateria atingirá seu fim de vida mesmo ainda havendo materiais ativos disponíveis. Entretanto, se a corrente de descarga da bateria é cortada ou reduzida durante o período de descarga, a taxa de transporte e difusão dos materiais ativos acompanha a taxa de degradação dos materiais que compõem a bateria.

A **unidade de comunicação**, responsável pela formação do enlace, pode ser formada por rádio, infra-vermelho ou meio óptico. Certamente, a forma mais comum de comunicação sem fio em uso é por rádio frequência (RF). Muitas das pesquisas de hardware para nós sensores são baseadas no desenho de circuitos RF. A utilização das faixas ISM (Industrial, Scientific and Medical) oferecem licença gratuita em muitos países. O nó μ AMPS, descrito em [22], utiliza um transceptor 2,4GHz compatível com Bluetooth com um sintetizador de frequência integrado. Os nós WINS (Wireless Integrated Network Sensors) [9] utilizam a comunicação por RF nas faixas de 902MHz-928MHz.

Os valores indicados na tabela 2.1 são referentes ao consumo de corrente do rádio CC1000 da Chipcon utilizado em redes de sensores sem fio comercializados pela Cross-Bow. A frequência de operação do rádio é de 868MHz. O valor da corrente consumida na transmissão com a potência máxima é de $5dBm$.

Operação	Corrente (mA)
Transmissão	25,4
Recepção	12,5
<i>Sleep</i>	0,001

Tabela 2.1: Consumo de corrente do rádio CC1000 - CrossBow.

O consumo de energia do rádio consiste na potência estática dissipada pela eletrônica analógica (análoga à energia desperdiçada no caso digital) e pela energia de radiação RF. A energia irradiada que varia com a distância transmitida de d^2 a d^4 , dependendo das condições ambientais, domina a energia do rádio. Para os microsensores, entretanto, a eletrônica do rádio se torna um ponto crítico. A energia necessária para uma transmissão completa de rádio pode ser descrita como $P_{tx}(T_{tx} + T_{start}) + P_{out}T_{out}$, onde P_{tx} representa a potência da eletrônica do transmissor, T_{tx} a duração da transmissão, T_{start} o tempo necessário para o início da operação e $P_{out}T_{out}$ a energia irradiada. Hoje em dia, os transceivers requerem um T_{start} da ordem de centenas de microsegundos, que são os transientes do circuito eletrônico.

Nós sensores com circuitos relativamente simples, como sensores analógicos, podem se beneficiar dessas técnicas. Entretanto, desligar circuitos mais complicados pode custar sobrecarga de tempo e energia. Por exemplo, desligar um processador requer preservação de seu estado. Se o processador for necessário imediatamente depois de ter sido desligado, a energia e o tempo gastos para salvar e restaurar seu estado são desperdiçados. Dessa forma, existe um ponto limite no qual os benefícios de economia de energia se iguala ao gasto associado com o desligamento e restauração do circuito à operação.

A **unidade de sensoriamento** presente nos nós sensores é em sua maioria das vezes composta pelos dispositivos sensores e pelo conversor de sinal analógico para digital (*Analog to Digital Converter - ADC*). De acordo com o fenômeno observado, são gerados os sinais analógicos gerados pelos sensores. Esses sinais são convertidos em sinais digitais pelo ADC que serão captados pelo processador. Existem diversos tipos de sensores entre eles magnetômetros, sensores de luz, temperatura, pressão e umidade.

O consumo de energia nos sensores passivos, como o termômetro, pode ser considerado se comparado às demais unidades do nó sensor. Entretanto, sensores ativos, o consumo de energia é maior e deve ser considerado. Na tabela 2.2 são mostrados alguns

Sensor	Corrente (μA)	Precisão
Magnetômetro	650	2mGauss
Luz	200	6mW/m ²
Temperatura	600	0.25°C
Pressão	650	2,4mPSI

Tabela 2.2: Consumo de energia de alguns dispositivos sensores.

Memória flash (programação)	128Kb
EEPROM (configuração)	4Kb
Consumo de corrente (Ativo)	8mA
Consumo de corrente (<i>Sleep</i>)	<15uA

Tabela 2.3: Características do processador - MICA2.

tipos de sensores e o consumo de corrente associados [30].

É na **unidade de processamento** que são executados os algoritmos, os protocolos de comunicação e todo o procedimento que permite o controle dos sensores e faz com que os mesmos trabalhem de forma colaborativa. Em virtude da limitação de energia dos nós sensores, os processadores devem buscar economizar o máximo de energia possível. Na tabela 2.3 são mostradas algumas informações a respeito do processador Atmel ATmega 128L utilizado no nó Mica Motes - MICA2 [31].

2.4 Pilha de protocolos para as Rede de Sensores Sem Fio

Para que o projeto de uma rede de sensor seja eficiente em termos do consumo de energia, as soluções apresentadas devem estar integradas a vários aspectos da rede. É importante conhecer o funcionamento do hardware que irá compor o nó sensor. Assim como não pode ser negligenciado o consumo de energia do dispositivo responsável pela transmissão dos dados. Serão apresentadas nas seções a seguir alguns aspectos das camadas física, enlace, rede, transporte e aplicação, destacando as principais pesquisas relacionadas.

2.4.1 Camada Física

Enquanto o desenvolvimento dos MEMS possibilita a redução do tamanho dos componentes de rádio-frequência (RF) e permite a integração de circuitos cada vez menores, existem certas limitações que trazem desafios para o projeto dos dispositivos sensores [30]. No projeto Smart Dust Mote [1] os nós sensores são projetados para possuírem um tamanho muito reduzido, tornando a comunicação via RF extremamente inadequada. É proposta, então, a utilização de transmissão óptica onde exista visada direta entre o receptor e o emissor. *Transceivers* ópticos requerem uma banda básica analógica, circuitos digitais e não requerem moduladores, filtros passa-faixas ativos ou demoduladores. Como o comprimento de onda é pequeno (da ordem de 1 micron) é possível a emissão de feixes estreitos, ou seja, com maior ganho da antena. Existem dois mecanismos de transmissão. O passivo que utiliza o CCR -*Corner-cube retroreflector* e o ativo usando um diodo laser com espelhos dirigíveis para enviar um feixe de luz colimada na direção do receptor. No passivo, não é necessária uma fonte de luz a bordo do circuito. Uma configuração com três espelhos (CCR) é usada para comunicar o sinal digital alto ou baixo.

De acordo com o modelo de propagação de rádio, a potência recebida diminui com o inverso da distância elevada a segunda ou quarta potência, devido ao desvanescimento de múltiplo-caminho (*multi-path fading*). Para aplicações baseadas em solo, como a comunicação móvel, o fator médio é quatro. Os transceptores de rádio comerciais são pouco apropriados para as aplicações em redes de sensores devido a seu gasto excessivo de energia para ligar e desligar. No projeto μ AMPS [22] são estudadas propostas alternativas para o projeto dos transceptores necessários para se realizar uma transmissão eficiente de pacotes a distâncias curtas.

No trabalho de Akyildiz [18] é sugerida a utilização da faixa de frequência ISM para as RSSFs. A comunicação utilizada no projeto WINS (*Wireless Integrated Network Sensors*) é por RF nas faixas 902-928MHz e em torno de 2,4GHz. Essas faixas são as mais apropriadas para o consumo de energia associado à transmissão e dimensão das antenas.

As redes IEEE802.15.4/ZigBee [32] permitem conectar de forma mais eficiente os dispositivos. Essas redes podem ter até 250 dispositivos conectados, operando na faixa

de frequência ISM de 2,4GHz. São empregados o protocolo CSMA-CA, que verifica a ausência de outros tráfegos antes da transmissão de dados, e a técnica de transmissão DSSS (*Direct-sequence spread spectrum*), que modula a fase da onda com uma sequência de códigos.

A rede *Bluetooth* permite uma comunicação de curto alcance com baixo consumo de energia e baixo custo. A unidade básica de uma rede *Bluetooth* é a picorede. A picorede é formada por um nó mestre conectado a até sete nós escravos, formando uma topologia em estrela. O mecanismo de acesso ao meio utilizado é TDMA e o rádio utiliza também a técnica de vários saltos de frequências. Alguns estudos são feitos para determinar a viabilidade da utilização do *Bluetooth* na implementação das RSSFs [33].

2.4.2 Camada de Enlace

A camada de enlace de dados possui um número específico de funções a realizar. Essas funções incluem prover um serviço de interface bem definido para a rede, determinando como os bits da camada física são agrupados nos quadros (frames), tratando os erros de transmissão, e regularizando o fluxo de quadros de modo que receptores mais lentos não sejam inundados por transmissores rápidos. Na subseção a seguir serão discutidos alguns mecanismos de acesso ao meio para redes de sensores.

O protocolo MAC (Medium Access Control) em uma rede de sensores sem fio com múltiplos caminhos e auto-ajustável deve atingir dois objetivos. O primeiro é a criação da infra-estrutura de rede. Como milhares de nós sensores estão densamente posicionados em um campo de sensoriamento, o protocolo MAC deve estabelecer enlaces de comunicação para a transferência dos dados. O segundo objetivo é compartilhar de forma eficiente e justa os recursos de comunicação entre os nós sensores.

Esquemas MAC tradicionais podem ser categorizados baseado no mecanismo de compartilhamento de recurso utilizado. O mecanismo de alocação fixo, ou pré-determinado é apropriado para tráfego contínuo e promove um atraso limitado. O TDMA (*Time Division Multiple Access*) requer um controle preciso da frequência de oscilação do rádio, e o CDMA (*Code Division Multiple Access*) requer a operação rápida de circuitos de

processamento digital de sinais por um período relativamente longo, o que pode levar a um elevado consumo de energia. Existem também o mecanismo baseado em contenção, onde existe a contenção do canal quando os pacotes são transmitidos. Esse esquema é adequado para tráfego em rajada, mas ineficiente para tráfego sensível ao atraso.

O protocolo SMACS (*Self-Organizing Medium Access Control for Sensor Networks*) [34] é utilizado para a inicialização e organização da camada de enlace da rede. O SMACS é distribuído e permite que os nós descubram seus vizinhos e agendem a transmissão e recepção sem a necessidade de qualquer nó mestre global ou local. Não há a formação de *clusters* nem nós principais, ou seja é construída uma topologia flat. Nesse protocolo, as fases de descoberta do vizinho e de alocação do canal são combinadas de modo que no instante em que os nós ouvem todos os seus vizinhos eles terão formado uma rede conectada. Um enlace de comunicação consiste no par de faixas (*slots*) de tempo que operam a uma frequência aleatoriamente escolhida, mas fixa (FHS - *Frequency Hopping Sequence*). Essa é uma opção possível em RSSF, uma vez que a largura de banda disponível é muito superior a taxa de envio dos dados. Os nós se comunicam intermitentemente e podem se desligar automaticamente para economizar energia quando não há dados para transmitir. A camada MAC utiliza TDMA, sendo que sua estrutura pode mudar de tempos em tempos. O esquema TDMA consiste de duas regiões: a primeira é chamada de período de bootup, quando os nós aleatoriamente procuram uma frequência fixa para novos nós serem incluídos na rede ou reconstruir enlaces perdidos; a segunda região é reservada para a comunicação de dados com os nós vizinhos.

O SMACS evita a necessidade de uma sincronização global da rede, embora os vizinhos em uma sub-rede devam estar sincronizados. Após a distribuição dos nós sensores pela região, cada nó é despertado e no período de bootup é feita a descoberta dos nós vizinhos. Assim que um enlace é descoberto, o primeiro período de tempo que está livre nos dois nós é atribuído a um canal, que é adicionado permanentemente aos seus esquemas. À medida que o tempo avança, os nós vão aumentando sua vizinhança, conectando novos nós, quando, eventualmente, todos os nós estarão conectados uns aos outros. A habilidade de se ter um esquema de comunicação assíncrono permite aos nós formar enlaces sob demanda. Os padrões de recebimento e transmissão são repetidos periodicamente, com um tempo fixo para todos os nós, característica do TDMA. É conseguida a conservação de

potência utilizando uma agenda para despertar os nós durante a fase de conexão e desligar o rádio durante os tempos de inatividade.

O protocolo EAR (*Eavesdrop-And-Register*) [17] oferece um serviço contínuo para os nós móveis sob condições estacionárias e fixas. O nó móvel assume controle total do processo de conexão e também decide quando terminar as conexões, minimizando a sobretaxa de mensagens. O EAR representa o aspecto de gerenciamento da mobilidade do protocolo e é transparente ao SMACS.

O EAR utiliza o primeiro período de tempo seguido do período de bootup para a comunicação. Ele usa as mensagens de convite enviadas em broadcast pelos nós estáticos para formar o registro de todos os nós alcançáveis. O protocolo usa as seguintes mensagens: BI (*Broadcast Invite*), para os nós estacionários convidarem outros a se conectarem à rede; MI (*Mobile Invite*), resposta de um nó móvel a um BI; MR (*Mobile Response*), resposta de um nó estacionário a um MR; e MD (*Mobile Disconnect*), informação de desconexão de um nó móvel a um estacionário.

O nó móvel adiciona a seus registros qualquer nó estacionário enviando a mensagem BI. Quando recebe um MI, um nó estacionário determina quando ele tem um slot TDMA disponível para a comunicação. Se houver um slot disponível, uma aceitação MR é enviada. Se o sinal entre os nós for se degradando ou aumentando, um MD pode ser enviado ou uma nova conexão solicitada. Para prevenir overheads de comunicação, são usados temporizadores para evitar uma espera indeterminada caso um MI seja recebido.

O algoritmo SAR (*Sequential Assignment Routing*) [17] cria múltiplas árvores cujas raízes estão a um salto do nó sink, ou seja, são seus vizinhos. Cada árvore cresce em direção externa ao sink deixando de fora nodos com baixa QoS (baixa vazão e alto atraso) e baixo nível de energia. No final desse procedimento, pode ocorrer a inclusão de alguns nós em árvores diferentes. Neste caso, os nós podem escolher a árvore a ser usada na transmissão de suas informações de volta ao sink. A cada uma dessas árvores são atribuídos parâmetros para a definição de qual caminho adotar. Por exemplo, energia estimada pelo número de pacotes que poderiam ser enviados caso houvesse um uso exclusivo da árvore, e métrica de QoS adicional.

Periodicamente, a redefinição dos caminhos são iniciadas pelo sink de forma a ocorrer a adaptação do roteamento devido a possíveis mudanças na topologia da rede. Adicionalmente, existem dois algoritmos chamados SWR (*Single Winner Election*) e MWE (*Multi Winner Election*), que são usados para realizar as atividades de sinalização e transferência de dados no processamento e informações cooperativas locais.

O S-MAC (Sensor-MAC) [35] é proposto como sendo um protocolo de controle de acesso ao meio especificamente projetado para RSSF. Através da utilização de esquema de contenção e agendamento combinados, o protocolo evita as principais causas de desperdício de energia provocados pela camada MAC: colisão e escuta de mensagens (overhearing).

2.4.3 Camada de Rede

A camada de rede possui a tarefa de realizar o roteamento dos dados. A maioria dos protocolos tradicionais de redes tem interesse é reduzir o retardo fim-a-fim ou aumentar a vazão dos dados. Nas RSSFs os protocolos da camada de rede devem estabelecer rotas que aumentem o tempo de vida da rede em detrimento das outras métricas de desempenho. Para o estabelecimento das rotas, os protocolos de roteamento podem utilizar a escolha de nós com maior energia disponível, nós com menor consumo de transmissão, caminhos mais curtos (redução do número de saltos), ou uma combinação das alternativas anteriores. Além de ser eficiente em termos de energia é desejável que o protocolo de roteamento seja centrado nos dados, possua endereçamento baseado em atributos [36] e seja tolerante a falhas.

Um dos primeiros propósitos do endereçamento em redes tradicionais é prover informação topológica que pode ser utilizada para a descoberta de rotas. Endereços são, às vezes, também utilizados como nomes para especificar os pontos finais de comunicação: "Preciso contactar *aquele* nó". Uma característica importante nas redes tradicionais é que cada nó possui um endereço único global. Um benefício adicional é que o endereçamento atribui um identificador único para os nós. Diferentemente de endereços, identificadores únicos não têm significado inerente. Eles não têm propriedades especiais além da capa-

cidade de identificação distinta. Muitos protocolos são desenvolvidos sob a suposição de que identificadores únicos estão disponíveis, os endereços são, por vezes, utilizados para este papel por estarem presentes.

A maioria dos esquemas de endereçamento pode ser classificado como utilizando endereços globais ou locais. Em sistemas como Ethernet, o endereço é estaticamente atribuído pelo fabricante dos dispositivos de rede. Essa abordagem garante um endereço distinto para um conjunto de dispositivos Ethernet interconectados. O custo de um endereço em uma rede com energia restrita pode ser considerado elevado se o endereço em si representa a maioria do total de bits transmitidos. Em RSSF os endereços únicos globais necessitariam ser grandes comparados ao tamanho médio dos dados trafegado pela rede. Dessa forma, o endereçamento local parece ser o mais adequado.

Para manter endereços locais, uma rede de sensores poderia utilizar um protocolo que dinamicamente atribua endereços aos nós baseado nos endereços de outros nós da vizinhança. Entretanto, quando a topologia de rede se torna mais dinâmica, mais trabalho é necessário para manter os endereços locais únicos. Esse esquema será eficiente somente enquanto o *overhead* para alocação do endereço for menor comparada a quantidade de dados úteis sendo transmitidos. Em redes estáticas, o trabalho necessário no início para resolver os conflitos de endereço é amortizado pelo trabalho a ser feito pela rede como um todo. Em redes de sensores, a dinâmica esperada torna este esquema potencialmente ineficiente dada a baixa taxa de transmissão de dados. A arquitetura de comunicação utilizada neste projeto utiliza o endereçamento baseado em atributos [37].

Existem duas topologias aplicáveis nas RSSFs: a plana e a hierárquica. Na topologia hierárquica os nós formam grupos (*clusters*) e alguns nós são eleitos líderes (*cluster head*). Dentro de cada grupo a comunicação só pode acontecer entre o nó líder e os demais nós. Pode haver ainda um agrupamento de líderes, formando mais um nível na hierarquia e a eleição de outro líder de nível superior. Em cada nível, o líder recebe as informações coletadas pelos outros nós. Pode haver o emprego de técnicas de agregação de dados no nó líder. Ao reunir todas os dados, o nó líder pode transmitir essas informações para o líder acima. Desta forma a informação atinge o usuário final, geralmente uma estação base. Os desafios presentes neste tipo de topologia são a eleição do líder em um ambi-

ente distribuído e a configuração da hierarquia de modo a evitar o consumo de energia excessivo em alguns nós da rede.

Na topologia plana, cada nó está conectado a um conjunto de outros nós, que se comunicam entre si. Não existem nós centrais por onde a comunicação deve sempre passar. Todos os nós são responsáveis por coletar e rotear as mensagens. Nessa topologia pode haver um excessivo número de transmissões devido ao grande número de interconexões na rede. Para evitar isso, os protocolos de roteamento devem descobrir caminhos que minimizem o consumo de energia.

Dentre os trabalhos para as topologias hierárquicas existentes, o de maior destaque é o LEACH (*Low Energy Adaptive Clustering Hierarchy*) [38]. A cada ciclo de funcionamento, o protocolo LEACH inicia a formação dos *clusters* através da auto-eleição aleatória dos nós como líderes. Após a eleição do líder, os demais nós decidem a que *cluster* devem pertencer, utilizando como critério de decisão a energia para se comunicar com o líder. O LEACH utiliza computação local nos nós líderes de modo a comprimir os dados e reduzir a quantidade de pacotes transmitidos até o sorvedouro.

No PEGASIS (*Power Efficient Gathering in Sensor Information Systems*) [39], cada nós se comunica apenas com seu vizinho próximo. É formada uma cadeia de nós para a transmissão da mensagem até o sorvedouro e todos os nós processam a agregação de dados, exceto o último. Apenas um nó por rodada transmite à estação base. A abordagem da cadeia pode aumentar o retardo na transmissão dos dados, o que era menor no LEACH.

Na topologia plana, uma das técnicas mais antigas de transmissão de dados é o *flooding*, inundação, que pode ser utilizada em RSSF. Cada nó retransmite as mensagens que não são destinadas a ele ou que atingiram o número de saltos estabelecidos. Na inundação, não é necessária a descoberta prévia de rotas, sendo uma técnica reativa. No entanto a utilização dos recursos de energia disponíveis na rede não é eficiente. Existem também os problemas de implosão e sobreposição de mensagens. Existe uma variação desse protocolo denominado *gossiping* [18]. No *gossiping*, o nó que recebe a mensagem seleciona um de seus vizinhos aleatoriamente para re-enviar os dados. Dessa forma é feita a transmissão de dados. Como só existe uma cópia da mensagem em cada nó, essa técnica evita a implosão. Entretanto, o tempo para a propagação da mensagem pode ser longo demais.

Um método de escolher as rotas é utilizar o algoritmo de roteamento MTE (*Minimum Transmission Energy*). Os nós intermediários são escolhidos de forma que a soma do quadrado das distâncias (e consequentemente a energia total transmitida, assumindo uma perda de potência d^2) seja minimizada. Nessa abordagem é necessário que haja o conhecimento da distância entre os nós.

O *Geographic and Energy Aware Routing* - GEAR é um protocolo de roteamento que utiliza uma heurística para a seleção de vizinhos baseada na energia e na posição geográfica dos nós. O algoritmo divide a área geográfica em regiões retangulares. Cada nó tem um custo associado a cada região da rede. O processo de encaminhamento de um pacote para um nó alvo compreende em dois passos. Primeiramente é feito o encaminhamento do pacote até a região. Se o nó atual não está na mesma região que é o alvo, ele seleciona o próximo nó dentre seus vizinhos que possui o menor custo até a região alvo. Quando o nó possuidor do pacote está na mesma região do alvo, é feito um *flooding* (inundação) restritiva, dentro da região.

No projeto PicoRadio [23] é desenvolvido o protocolo EAR (*Energy Aware Routing*) [13]. O EAR é um protocolo reativo que busca caminhos de mínimo dispêndio de energia pela rede. O protocolo mantém uma lista de bons caminhos descobertos, e a cada rodada escolhe probabilisticamente um deles. Desta forma evita que o melhor caminho tenha a sua energia esgotada, aumentando o tempo de sobrevivência da rede. Esse protocolo inspirou a proposta dessa tese.

O protocolo de roteamento desenvolvido é centrado nos dados, onde as primitivas de comunicação são expressas não em termos de identificadores dos nós, mas em termos de dados nomeados. Nesse trabalho os nós são estáticos e não possuem qualquer sistema de localização que permite aos nós saberem sua posição atual. Motivado pelas restrições de energia, o trabalho utiliza como métrica a energia residual dos nós. O protocolo é reativo, ou seja, constrói as rotas a medida que tem dados de interesse a serem coletados. A idéia é evitar um *overhead* desnecessário para o estabelecimento dos caminhos. As rotas ótimas são escolhidas baseada na energia de cada nó ao longo do caminho. É feita uma distribuição do tráfego pela rede, evitando a formação dos *hot-spots*, conforme será mostrado no capítulo 4.

Um outro protocolo de comunicação centrado nos dados é o *Directed Diffusion* (Difusão Direcionada) proposto pelo grupo da pesquisadora Prof. Deborah Estrin [40]. A proposta dessa tese foi implementada sobre a arquitetura do protocolo para a difusão direcionada. Dessa forma, tal protocolo será apresentado com maiores detalhes na subseção seguinte.

Protocolo de difusão direcionada

O protocolo para a difusão direcionada consiste em alguns elementos. Os *dados* são nomeados utilizando pares atributo-valor. Uma tarefa de sensoriamento é disseminada pela rede de sensores como um *interesse* para um dado nomeado. Essa disseminação estabelece *gradientes* pela rede destinados a coletar os *eventos*, que são os dados que atendem aos interesses. Os eventos começam a fluir na direção dos nós originadores dos interesses por múltiplos caminhos. A rede *reforça* um, ou um pequeno número, desses caminhos. A figura 2.6 é um esquema simplificado das etapas de operação do protocolo de difusão direcionada.

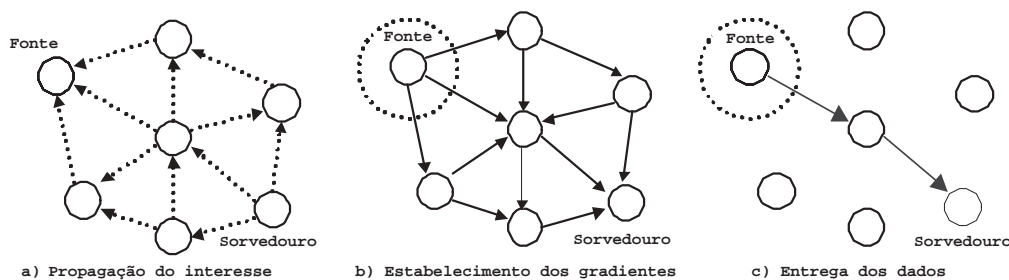


Figura 2.6: Um esquema simplificado para o difusão direcionada

No difusão direcionada, a atividade de monitoramento desejada pelo usuário da aplicação é entendida pela rede como uma tarefa. O **esquema de nomeação de dados** atribui às tarefas uma lista de pares atributo-valor que a descrevem. Por exemplo, a tarefa de monitorar a localização de um animal a ser realizada pelos sensores localizados dentro do retângulo e enviar os dados a cada 20ms por 10s pode ser descrita da seguinte forma 2.7(a).

Intuitivamente, a descrição da tarefa especifica o interesse para os dados que corres-

pondam aos atributos. Por essa razão, a descrição da tarefa é denominada *interesse*. Os dados enviados em resposta aos interesses são também nomeados utilizando um esquema de nomeação similar. Por exemplo, um sensor que detecta um animal pode ser capaz de gerar o seguinte dado 2.7(b).

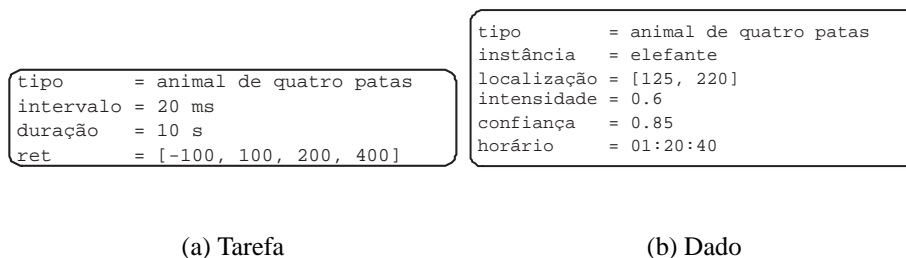


Figura 2.7: Esquema de nomes

Dado um conjunto de tarefas suportadas pela rede de sensores, o esquema de nomeação deve ser simples e atender aos requisitos da rede. O esquema utilizado pelo Direced Diffusion é baseado no par atributo-valor para os interesses e para os dados. Em geral, cada atributo possui associado a ele um conjunto de valores. Por exemplo, o conjunto associado ao atributo tipo é um grupo de valores que representam objetos móveis (animal, veículo, ser humano). O valor de um atributo pode ser qualquer subconjunto desse conjunto. No exemplo, o valor do atributo tipo no interesse é animal de quatro patas.

O **interesse** é a descrição da tarefa nomeada. Um interesse é introduzido na rede por um nó especial, denominado nó sorvedouro, do inglês *sink*. Esse nó envia mensagens em broadcast periodicamente para cada um de seus vizinhos. Esse interesse inicial é exploratório, ou seja, o intuito é de verificar se existe algum nó capaz de atender à tarefa desejada.

Como pode haver perda de mensagens, o nó sorvedouro re-transmite uma outra mensagem de interesse periodicamente similar à primeira mensagem, apenas variando o atributo hora. A taxa de re-envio das mensagens é um parâmetro do protocolo que aumenta a robustez à perda dos interesses, mas eleva também o overhead de mensagens e consequentemente o consumo dos nós.

Cada nó mantém um cache de interesse, onde cada item corresponde a um interesse distinto e não contém informação sobre o nó sorvedouro. Cada entrada no cache contém

vários campos, tais como o horário, que indica quando o último interesse foi recebido, e os gradientes, um para cada vizinho. Cada gradiente contém um campo de taxa de envio de dados (derivado do atributo intervalo), e o campo duração (derivado dos atributos do interesse horário de geração e tempo de vida). Quando um gradiente expira, ele é removido da entrada de interesse. Quando todos os gradientes para uma entrada de interesse expiram, o interesse por completo é removido do cache.

Quando um nó recebe um interesse, ele verifica se o interesse existe no cache. Se não existe o batimento com nenhum dos interesses distintos armazenados, o nó cria uma nova entrada de interesse. Cada entrada no cache do nó possui um gradiente que aponta para o vizinho que o enviou o interesse, com a taxa de envio dos dados. Para tanto, deve ser possível distinguir os vizinhos individualmente através de identificadores como o endereço MAC 802.11, ou endereçamento em cluster do Bluetooth. Caso exista uma entrada de interesse, mas não exista gradiente para o emissor do interesse, o nó adiciona um gradiente com o valor especificado. E, finalmente, se existir tanto uma entrada e um gradiente, o nó simplesmente atualiza os campos de horário e duração da mensagem armazenada.

Após receber um interesse, um nó pode decidir re-enviar o interesse. Para os seus vizinhos, o interesse parecer ser originado pelo nó emissor da mensagem, embora possa vir de um outro nó distante. São utilizadas apenas interações locais para a disseminação dos interesses. E como consequência, todo par de nós vizinhos estabelece um gradiente entre si, como ilustrado na figura 2.8. A existência desses gradientes pode causar o recebimento de mensagens duplicadas. No entanto, esta técnica permite a seleção de caminhos empiricamente melhores e o restabelecimento rápido do fluxo de dados no caso de falhas nas rotas.

Um nó sensor, ao detectar um alvo, busca no seu cache de interesse por uma entrada correspondente. Caso haja um casamento de interesse, o nó, que se torna nó fonte, computa a maior taxa de requisição de dados dentre todos os gradientes e ajusta o sensor para gerar os eventos nesta taxa. O nó fonte envia uma descrição do evento para cada vizinho que ele tenha um gradiente. É iniciada, então, a fase de **propagação dos dados**.

Ao receber uma mensagem de dados dos seus vizinhos, o nó tenta encontrar uma en-

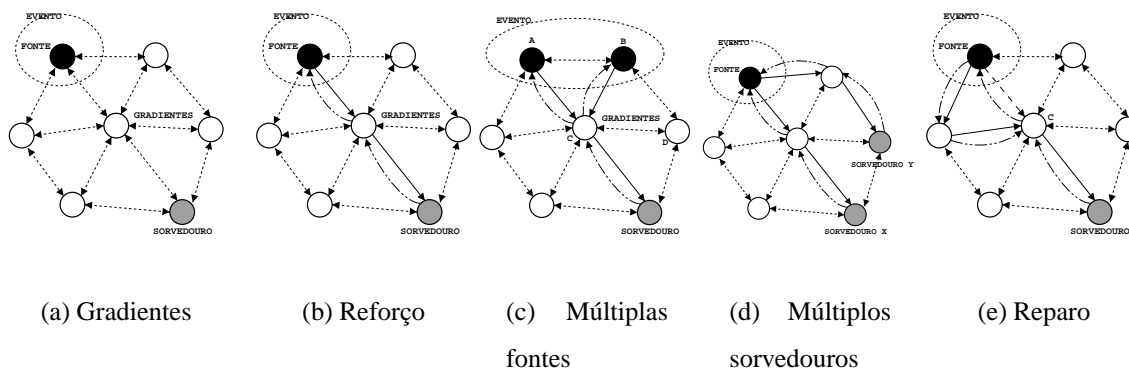


Figura 2.8: Aspectos da difusão direcionada

trada de interesse correspondente na sua lista. Se não existir, a mensagem de dados é descartada silenciosamente. Caso exista, o nó verifica em outra lista, que contém um histórico das mensagens de dados recentemente enviadas. Essa lista de dados possui várias funções, uma delas é prevenir a formação de laços. Se a mensagem de dados recebida estiver na lista de dados, a mensagem é descartada. Caso contrário, a mensagem recebida é adicionada à lista de mensagens e é re-enviada para os vizinhos do nó que possuem um gradiente ativo. Assim, os dados fluem das fontes até o sorvedouro percorrendo múltiplos caminhos pela rede. Essas mensagens de dados iniciais são denominadas **exploratórias**, pois destinam-se à configuração e descoberta das rotas. Após o sorvedouro começar a receber esses eventos de dados com baixa taxa, ele reforça um vizinho específico de modo a obter eventos com taxa de dados maior.

Para reforçar um caminho, o sorvedouro re-envia a mensagem de interesse original, porém com uma taxa de dados maior. Esse reforço é denominado **reforço positivo**. Quando o nó vizinho recebe esse interesse, ele verifica se já possui um gradiente para esse nó emissor e se a taxa de envio de dados é maior que a de qualquer gradiente existente, então reforça ao menos um vizinho. A regra de reforço positivo é localizada, isto é, cada nó decide qual vizinho terá seu gradiente reforçado. Através dessas interações locais é estabelecida uma rota para a transmissão de dados entre o nó fonte e o sorvedouro. A regra implementada no protocolo original reforça o gradiente do primeiro nó vizinho que responder a um interesse propagado pela rede. Nesse caso, a regra de reforço positivo seleciona empiricamente a rota de menor retardo.

2.4.4 Camada Transporte

Existem ainda poucos estudos realizados para a camada de transporte para as redes de sensores sem fio. A camada de transporte assume maior importância quando a aplicação realizará acesso a redes externas, como a Internet. Em RSSF, a comunicação entre o usuário e o sorvedouro seria realizada utilizando-se o TCP (Transmission Control Protocol) ou o UDP (User Datagram Protocol) via Internet, e entre os nós sensores e o sorvedouro utilizaria-se o UDP somente. Entretanto os problemas de transporte seguro e controle de congestionamento, ainda são tópicos que necessitam de discussões e propostas de soluções para atender aos requisitos das RSSF.

O protocolo ESRT (*Event-to-Sink Reliable Transport in Wireless Sensor Network*) proposto por Yogesh et al. [41] controla a comunicação fim-a-fim pelo estado corrente da rede, baseando-se no nível de confiabilidade e na condição de congestionamento. O objetivo é de manter a rede na região ótima, onde a confiabilidade da comunicação é atingida com o mínimo consumo de energia e sem congestionamento da rede.

O protocolo RMST (*Reliable Multi-Segment Transport*) foi desenvolvido para operar em conjunto com o Directed Diffusion. O RMST realiza a fragmentação e remontagem das mensagens, adicionando atributos específicos aos dados que fluem de uma fonte para o sorvedouro.

2.5 Comentários

O presente capítulo apresentou uma visão geral sobre as redes de sensores sem fio, contextualizando-as dentro das redes móveis. Foram descritas as principais características de *hardware* e *software* para as RSSF, e os protocolos de comunicação utilizados pelos nós sensores. Alguns trabalhos propostos na área também foram apresentados. Além disso, foi explicado o mecanismo de funcionamento do protocolo de difusão direcionada original.

Como pode-se notar a energia é o fator mais crítico em uma RSSF. O tempo de vida da rede depende da energia economizada no em cada camada de protocolo desde a camada

física até a de aplicação, sendo que os maiores ganhos são obtidos quando é reduzido o número de transmissões na rede. No próximo capítulo, será descrito o projeto desenvolvido para a camada de rede de um protocolo de roteamento centrado em dados.

Capítulo 3

Protocolo de Roteamento com Métrica de Energia

NESTE capítulo, é proposto um mecanismo para a escolha de rotas em RSSF utilizando métrica de energia para evitar que alguns nós na rede tenham seus níveis de energia esgotados prematuramente. Para tanto, é proposta uma extensão a um protocolo de roteamento centrado em dados, de forma a permitir que os nós decidam localmente por qual rota encaminhar os dados sem incorrer em altos custos relacionados ao conhecimento de toda a topologia da rede.

Este capítulo está estruturado da seguinte forma. Na seção 3.2 é descrita a extensão feita para o protocolo original. Depois de descrever o problema e a solução na seção 3.2.1, através de um tratamento matemático, é mostrado na seção 3.2.2 detalhes da implementação no simulador de redes ns-2.

3.1 Introdução

Algumas aplicações de redes de sensores como monitoramento de temperatura de uma região, por exemplo, precisam que todos os nós estejam operantes durante o maior tempo possível. Não é desejável que alguns nós tenham seus níveis de energia diminuídos antes de outros. As depressões de energia podem prejudicar a qualidade do sensoriamento

como um todo. Nesse trabalho, todos os nós sensores são importantes para o usuário quer estejam sensibilizados pelo fenômeno de interesse, ou seja gerando dados, quer estejam servindo apenas de roteadores. O intuito da proposta é de prolongar o tempo de vida dos nós sensores que compõem a rede, através da distribuição do consumo de energia pela rede sem incorrer em altos custos de comunicação e de computação.

O protocolo centrado nos dados utilizado foi o difusão direcionada. O mecanismo proposto estende o tempo de sobrevivência das RSSF através da descoberta de novos caminhos pela rede, distribuindo o gasto de energia. Em uma rede composta por muitos nós, há elevada perda de energia empregada na comunicação necessária para centralizar o processo de decisão do algoritmo. Protocolos pró-ativos mantêm antecipadamente em suas tabelas de roteamento caminhos para os destinos, e para manter essas entradas atualizadas necessitam trocar mensagens frequentemente. Neste trabalho, o protocolo realiza apenas interações locais e é reativo. A cada rodada do protocolo é computada a energia residual dos nós que compõem o caminho e a distância em número de saltos entre o destino e a origem. A partir dessas informações, é escolhida a rota a ser utilizada pelo fluxo de dados.

3.2 Extensão ao protocolo difusão direcionada

O difusão direcionada utiliza como métrica para a escolha da rota àquela que possui menor retardo, o que geralmente leva à rota com menor quantidade de saltos. A rota selecionada é alterada somente na próxima rodada do protocolo. Isso pode levar ao esgotamento de energia os nós escolhidos para fazerem parte da rota. O filtro Energia foi desenvolvido para modificar o roteamento original da difusão direcionada, escolhendo caminhos que levem a um balanceamento de energia entre os nós. Esse filtro é instanciado em todos os nós da rede. A modelagem do problema é mostrada na seção a seguir.

3.2.1 Modelagem

Pode-se modelar o problema de minimizar o consumo da rede como a seguir. Seja $G = (V, E)$ um grafo direcionado onde V é um conjunto de vértices (nós sensores) e E é

o conjunto de pares (u, v) não ordenados de elementos em V (conexões). Um grafo é dito direcionado, ou digrafo, quando o sentido das ligações entre os vértices é considerado. Nesse caso, denomina-se de arco a aresta direcionada e E é uma relação binária em V , i. e. um conjunto de pares ordenados, das ligações.

Inicialmente, cada vértice u possui uma bateria com energia $E_u \geq 0$. Para cada arco $e = (u, v)$ associamos o custo de envio da mensagem s_e (ou $s_{u,v}$), que corresponde a energia usada pelo nó ao enviar a mensagem. Os nós que não participam efetivamente da comunicação ficam com seus circuitos receptores desligados, não consumindo energia. O consumo de energia para a recepção da mensagem é considerado baixo e a perda de energia associada ao mesmo pode ser incluída no custo do envio da mensagem. Enviar um pacote por $e = (u, v)$ reduzirá a E_u , a energia de u , em s_e . O envio das mensagens somente ocorre caso E_u seja superior a s_e .

Ao enviar um pacote de um vértice s para um vértice t em um grafo G , é possível freqüentemente escolher entre alguns caminhos diferentes uma rota através da qual a mensagem é encaminhada. O intuito é de utilizar os caminhos que conservem a energia residual da rede, isto é, evitar caminhos de roteamento de mensagens que levem ao esgotamento das reservas de energia dos nós da rede.

Se for encaminhada uma mensagem por um caminho $C = (v_1, v_2, \dots, v_{k-1}, v_k)$ em G , onde v_1, \dots, v_k são vértices e $(v_1, v_2), \dots, (v_{k-1}, v_k)$ são arestas, então cada nó em C perde uma parcela de sua energia associada ao custo de envio de mensagens. Seja R_{v_i} a energia residual de v_i após o envio de uma mensagem. Então, $R_{v_i} = E_{v_i} - s_{v_i, v_{i+1}}$ para $i = 1, \dots, k - 1$. Define-se o caminho que maximiza a energia da rede, denotado por $M(C)$ como $M(C) = \max_{i=1, \dots, k-1} (\sum_{i=1}^{k-1} R_{v_i})$.

A regra implementada no protocolo não escolhe o caminho que possui o maior valor de energia residual disponível por todos os nós. A razão está em evitar que os caminhos mais longos possam ser favorecidos. Caminhos mais longos são aqueles que possuem um maior número de saltos entre o nó fonte e o nó sorvedouro. Rotear um pacote por um caminho longo implica excessivo gasto de energia em transmissões, o que é indesejável. Dessa forma, o critério utilizado para escolher o melhor caminho foi calcular a distribuição de energia disponível nos nós ao longo do caminho. Como a solução utiliza apenas

interações localizadas, introduziu-se a razão K obtida através da divisão entre o somatório da energia residual e o número de nós que compõem o caminho $K = \frac{M(C)}{k-1}$. Considera-se, portanto, que o melhor caminho é aquele que dentre todos os caminhos disponíveis de um dado par de vértices de origem e destino obtiver a maior razão K .

3.2.2 Implementação

O protocolo de difusão direcionada possui uma arquitetura dividida em filtros e em núcleo. Os filtros são módulos de software adicionados ao código original com finalidades de interagir com o núcleo do protocolo. Pode-se alterar ou acrescentar funcionalidades específicas ao protocolo original, como agregação de dados e alteração do mecanismo de roteamento [42]. A interface entre os filtros e o núcleo do protocolo é feita através da API de filtros descrita em [43]. Os filtros possuem prioridades, que são inicializadas via código. As prioridades são utilizadas para definir a ordem da passagem das mensagens pelos filtros. Filtros com prioridades mais altas receberão as mensagens primeiro, podendo assim, alterá-las antes de repassarem a mensagem para o núcleo do protocolo.

O módulo de roteamento do protocolo de difusão direcionada denomina-se *Gradiente*. Este módulo define as rotas utilizadas para enviar as mensagens de dados. Essas rotas foram usadas para o encaminhamento de mensagens exploratórias provenientes das fontes e de reforços positivos enviados pelo sorvedouro. Já o núcleo do protocolo tem como funções o armazenamento das mensagens e a manutenção da lista dos vizinhos dos nós, dentre outras.

O módulo desenvolvido foi nomeado filtro Energia. A interface entre o filtro Energia e o núcleo do difusão direcionada foi realizada utilizando a API de filtros. Como padrão, o filtro *Gradiente* é inicializado com uma prioridade 75 pelo simulador. Então, o filtro Energia recebeu um valor de prioridade igual a 85. O filtro Energia, ao ser instanciado pelos nós, está configurado para tratar todas as mensagens recebidas do tipo exploratórias. As mensagens de interesse e as mensagens de dados não são processadas.

Por ter uma prioridade maior que o Gradiente, quando uma mensagem exploratória é recebida pelo nó, ela é encaminhada pelo núcleo da difusão direcionada para o filtro Ener-

gia. Este, ao invés de imediatamente encaminhá-la ao originador do interesse, configura um temporizador associado à mensagem recebida e armazena a mensagem em um cache local. Quando o temporizador expira é feita uma comparação entre as informações contidas nas mensagens exploratórias existentes localmente para se determinar qual mensagem será encaminhada para o núcleo do protocolo. A mensagem encaminhada ao núcleo será posteriormente tratada pelo filtro *Gradiente*, conforme mostrado na figura 3.1.

A noção de mensagens entrantes ou saintes no protocolo de difusão direcionada é um pouco diferente do fluxo em uma arquitetura de redes. As mensagens podem chegar ao núcleo do protocolo, representado pelo ponto 1 na figura 3.1, tanto por uma aplicação local quanto pela rede. Após passar pelos filtros, o destino da mensagem pode ser de novo ou a rede ou uma aplicação local, ponto 6 na figura 3.1. Todas as mensagens passam pelo filtro na mesma ordem independente da fonte. No processamento original da difusão direcionada, o filtro *Gradiente* encaminha a primeira mensagem exploratória que chega ao nó e descarta as subsequentes. Nessa extensão, o filtro *Energia* por ser mais prioritário que o do *Gradiente* será tratado primeiro. Graficamente, na figura 3.1, filtros mais prioritários são posicionados mais a esquerda.

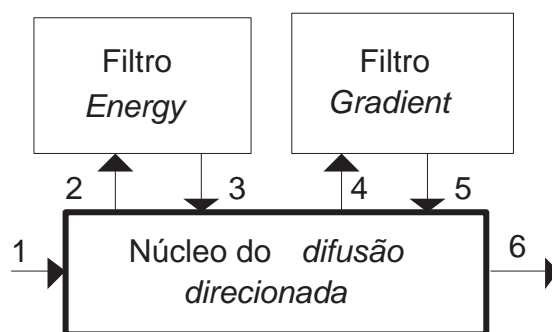


Figura 3.1: Interação do filtro Energy com o núcleo do protocolo

O armazenamento de todas as mensagens exploratórias em um cache local pelo filtro *Energia* é necessário para que seja feita a seleção de rotas segundo os critérios descritos em 3.2.3. A seleção da melhor rota em termos de energia é feita em duas etapas. A primeira etapa consiste no armazenamento das mensagens exploratórias e a criação de um temporizador incremental. A segunda fase é iniciada após o término do temporizador. Nessa fase é feita a seleção da mensagem exploratória que será encaminhada ao núcleo do protocolo de difusão direcionada.

Mensagem	Descrição
1 :	Recepção das mensagens exploratórias
2 :	Inicialização do temporizador e armazenamento das mensagens exploratórias localmente
3 e 4 :	O filtro Energia envia para o núcleo do difusão direcionada apenas a mensagem selecionada. O núcleo encaminha a mensagem para o próximo filtro, o Gradiente
5 :	O filtro Gradiente encaminha a mensagem para os vizinhos que têm interesse naquela informação
6 :	O núcleo do difusão direcionada envia as mensagens exploratórias que estabelecerão os caminhos reforçados para a entrega das mensagens de dados

Tabela 3.1: Tabela descritiva do comportamento do filtro Energia

Ao receber uma mensagem, o filtro Energia processa a mensagem para verificar se a mensagem é do tipo Exploratória. Caso seja, ele temporiza um contador, o associa a mensagem recebida e armazena localmente os candidatos a caminhos com maior energia disponível. Ao expirar o temporizador, são adicionados novos campos na mensagem, conforme mostrado em 3.2.1.

3.2.3 Configuração do Filtro

O filtro Energia é instanciado em todos os nós sensores que compõem a rede. O filtro modifica o conteúdo de dois atributos adicionados ao cabeçalho das mensagens originais do difusão direcionada:

- *dist_fonte*: contém o número de saltos entre a fonte e o destino;
- *energia_caminho*: armazena o somatório das energias residuais de cada nó que compõe o caminho entre a fonte e o destino

É utilizado o atributo *dist_fonte* para o temporizador instanciado nos nós. O temporizador possui um papel muito importante no mecanismo proposto. O filtro Energia

armazena as mensagens exploratórias, que são responsáveis pelo estabelecimento das rotas na rede. Se o valor do temporizador for constante para todos os nós, o resultado do mecanismo não seria satisfatório. As mensagens exploratórias dos nós muito distantes do nó coletor poderiam não ter tempo de chegar ao nó sorvedouro para serem selecionadas, conforme será ilustrado em 3.2.

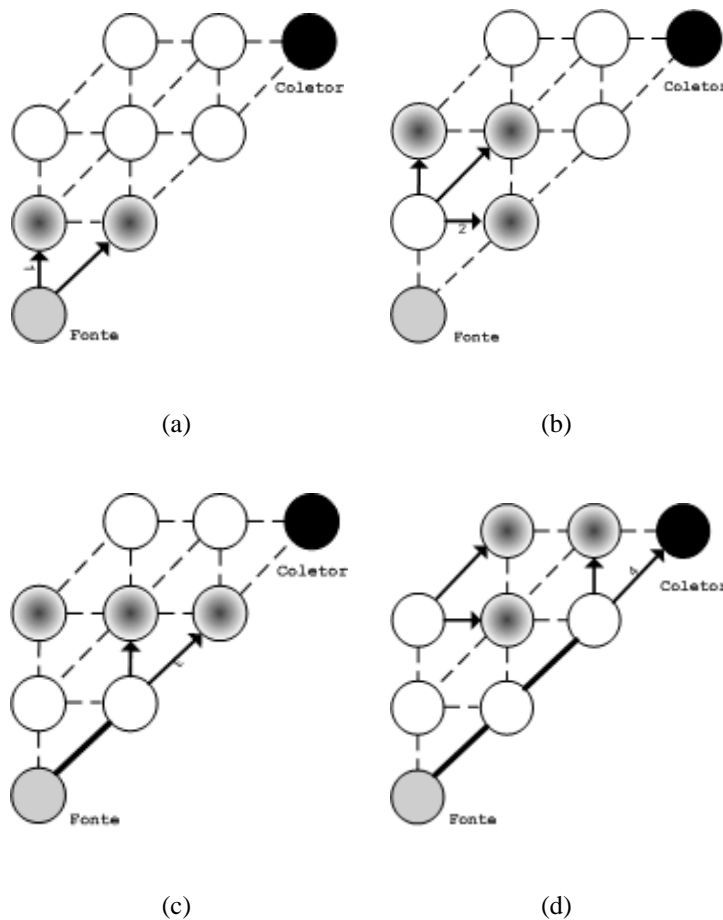


Figura 3.2: Estabelecimento das rotas com temporizador constante

Na figura 3.2, a fonte envia as mensagens exploratórias para seus dois vizinhos, o que torna seus temporizadores ativos. Supondo que a mensagem indicada pela seta com o subscrito 1 chegue primeiro ao seu destino, ela ativará o nó vizinho da esquerda antes do nó da direita, figura 3.2 (a). Dessa forma, o temporizador do nó da esquerda expira antes e envia a mensagem dois para seus três vizinhos, 3.2 (b). A seguir, em 3.2 (c), o temporizador do nó da direita vizinho a fonte expira e é enviada a mensagem três para seus dois vizinhos. O gradiente escolhido, indicado na figura 3.2 por uma linha mais

forte, aponta diretamente para o nó fonte. Isso é devido ao critério de escolha das rotas descrito anteriormente, em 3.2.1. Supondo que a mensagem três chegue antes no nó de destino, esse nó terá seu temporizador sensibilizado. Novamente, a escolha do gradiente está indicada na figura pela linha mais forte. Dessa forma, o caminho mais curto entre a fonte e o destino pode ser o escolhido, caso o temporizador seja único em toda a rede. Para evitar esse problema, o temporizador é variável em cada nó da rede. O valor calculado para o temporizador é diretamente proporcional a distância dos nós até a fonte. Isso implica que quanto mais distante o nó estiver da fonte maior será seu tempo de espera para o recebimento das mensagens exploratórias.

A fonte f atribui o valor de sua energia residual ao campo $Energia_Caminho$ e atribui o valor 1 ao campo $Distncia$. Assim, a mensagem enviada pela fonte v_1 , m_{v_1} , conterá energia do caminho, $E_C = R_{v_1}$, e distância até a fonte $D_C = 1$.

A cada rodada do protocolo são executados três passos:

- **Fase de Início** : O(s) nó(s) sorvedouro(s) envia(m) mensagens contendo os descritores dos dados de interesse para cada vizinho. Essas mensagens de interesse descrevem uma tarefa solicitada por exemplo por um usuário conectado a Internet. Após o recebimento do interesse, o nó re-envia esse interesse para os seus vizinhos, como a implementação original da difusão direcionada descrito no capítulo 2. O mecanismo proposto não altera a característica do protocolo de somente realizar interações localizadas. Na figura fig 3.3 ilustra-se uma rede de sensores composta de 6 nós, sendo que um deles é o sorvedouro, e o v_1 é o nó sensibilizado pelo fenômeno. O número dentro dos nós corresponde à sua energia residual em joules.
- **Fase de estabelecimento das rotas** : Quando o interesse atinge uma região apropriada, um ou mais sensores são ativados tornando-se fontes. Esses nós fontes começam a enviar mensagens de dados na forma atributo-operação-valor pelos gradientes configurados na etapa inicial. Na implementação original da difusão direcionada, os nós intermediários re-enviam a primeira mensagem exploratória recebida, até que se chegue ao(s) nó(s) sorvedouro(s). São favorecidos dessa forma, os caminhos de baixa latência pela rede. Na extensão realizada, as mensagens recebidas são armazenadas localmente no nó e é associado um temporizador que controlará

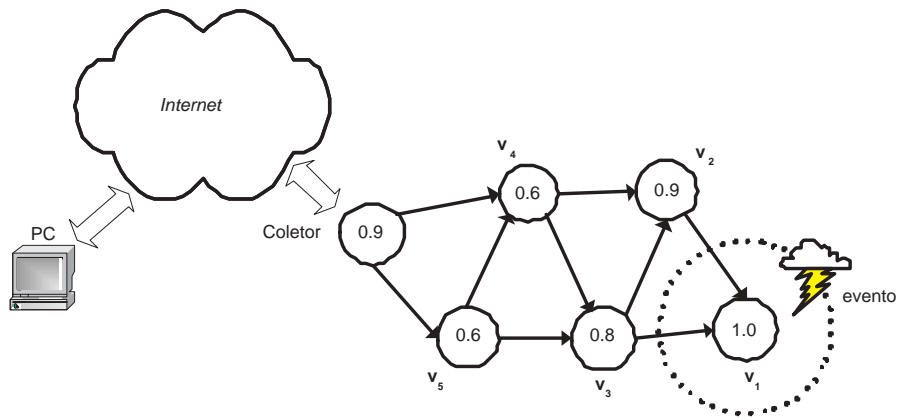


Figura 3.3: Propagação das mensagens de interesse

o envio das mensagens. Observe que existe um comprometimento entre atraso e consumo de energia. Dependendo da aplicação em RSSF (p. ex. monitoramento remoto do meio ambiente), o aumento da latência no recebimento dos dados não é tão prejudicial quanto à ausência dos mesmos devido a desconectividade da rede.

- **Envio das mensagens exploratórias** : Cada nó intermediário armazena localmente as mensagens exploratórias recebidas. Suponha que o vértice v_2 receba uma mensagem exploratória vinda do vértice v_1 . É feita a temporização da mensagem m_{v_1} em uma estrutura local do nó v_2 . Caso v_2 receba outra mensagem exploratória oriunda de outro vértice, p. ex. v_3 também é armazenado em seu cache m_{v_3} e atribuído um temporizador. Ao expirar o temporizador é feita uma busca no cache para determinar qual é a melhor mensagem recebida. É escolhida a mensagem que possuir a maior razão entre a energia do caminho e a distância em saltos até a fonte, $K = \frac{E_C}{D_C}$, dentre as mensagens armazenadas. Observe na figura 3.4 que o nó v_3 considerou a mensagem enviada pelo vértice v_1 e descartou a mensagem vinda por v_2 .

Re-envio das mensagens exploratórias : A partir da seleção feita no passo anterior, são computados novos valores para E_C e D_C . Caso a mensagem vinda do nó v_2 obtenha a maior razão K são calculados novos valores para $E_C = R_{v_1} + R_{v_2}$ e $D_C = d_{v_2,f} + 1$, onde $d_{v_2,f}$ representa a distância entre o vértice v_2 e a fonte f . Pode-se escolher caminhos com D_C maiores para proporcionar o balanceamento da energia consumida pela rede.

Fase de envio dos dados Através do envio das mensagens exploratórias é es-

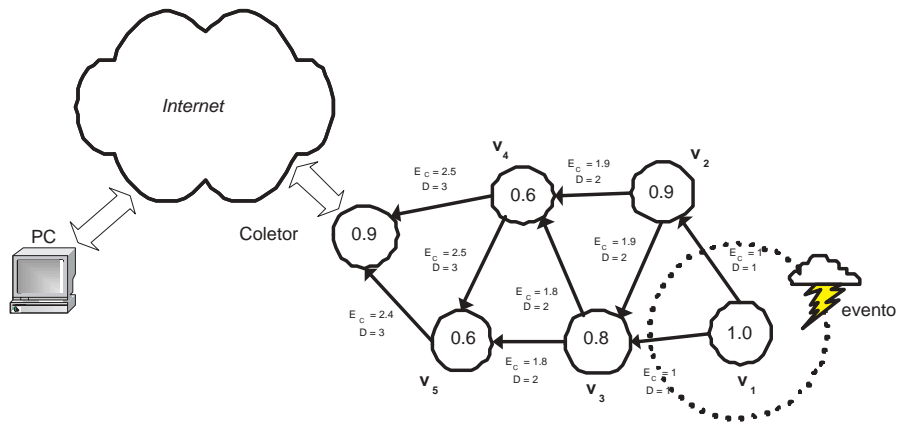


Figura 3.4: Propagação das mensagens exploratórias

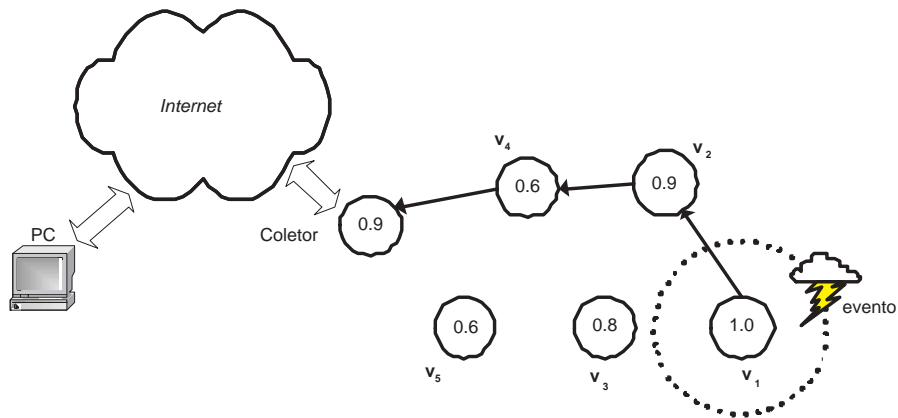


Figura 3.5: Propagação dos dados

tabelecido um gradiente entre o nó receptor e o nó que enviou a mensagem. Assim que o nó fonte começar a enviar seus dados, as mensagens são encaminhadas pelos nós intermediários segundo a direção apontada pelo gradiente, ilustrado na figura 3.5.

3.3 Comentários

Em uma rede de sensores sem fio é desejável que o protocolo de roteamento seja baseado em atributos ao invés de ser baseado em endereços. Manter uma lista de identificadores únicos em uma rede de sensores pode ser custoso em termos de gastos de energia devido ao dinamismo da rede. Dentre os protocolos de roteamento baseados em atributos, o protocolo de difusão direcionada recebe destaque na área. Para as aplicações em

RSSF onde todos os nós da rede são importantes, o protocolo original mostra-se pouco adequado. No protocolo original, os caminhos com baixo retardo são privilegiados. Tal fato pode levar alguns nós à exaustão precoce.

O filtro Energia proposto nesta dissertação tem como objetivo balancear o consumo de energia pelos nós da rede. Através da implementação do filtro Energia, pretende-se modificar o comportamento original do protocolo de difusão direcionada e aumentar o tempo de vida da rede. O próximo capítulo traz os resultados obtidos com o filtro Energia para a topologia simulada.

Capítulo 4

Simulação e Análise

NESTE capítulo são apresentadas as simulações realizadas para analisar o desempenho do protocolo proposto. É feita uma análise comparativa entre o protocolo de difusão direcionada original e o filtro Energia proposto. Na seção 4.1 é descrito o ambiente de simulação utilizado. As métricas utilizadas, descritas na seção 4.2, buscam quantificar os ganhos obtidos com a proposta.

4.1 Ambiente de simulação

O ambiente de simulação utilizado é o NS-2 (*Network Simulator*) [44], um simulador dirigido a eventos discretos, que oferece suporte para a simulação de serviços e protocolos de rede. O NS-2 está em desenvolvimento no projeto VINT (*Virtual InterNet Testbed*) desenvolvido pelo LBNL (*Lawrence Berkley National Laboratory*), o ISI (*Information Sciences Institute*) na Universidade da Califórnia do Sul (USC) e o laboratório da Xerox PARC. O NS-2 utiliza as linguagens C++ e OTcl (*Object Tool Command Language*) [45], que permite a interação com o código em C++ dinamicamente. O protocolo foi implementado em C++ e a configuração da topologia, dos nós, e dos parâmetros do protocolo foi feito em scripts OTcl.

Para os experimentos descritos nesta dissertação, foi utilizado tanto o modelo de recepção em espaço aberto quanto o modelo de múltiplos caminhos, dependendo da dis-

tância entre o transmissor e o receptor, conforme definido pelo modelo de propagação do canal no NS-2. Não importante qual modelo seja utilizado (comunicação direta ou efeito de múltiplos caminhos), a potência recebida diminui com o aumento da distância entre o emissor e o receptor. Se a distância entre o transmissor e o receptor é inferior a uma distância limiar de *cross-over* ($d_{crossover}$), então o modelo de espaço aberto de Friss é utilizado (atenuação d^2), e se a distância é superior a $d_{crossover}$, o modelo de propagação por múltiplos caminhos é utilizado (atenuação d^4). O ponto de *cross-over* é definido por:

$$d_{crossover} = \frac{4\pi\sqrt{L}h_r h_t}{\lambda} \quad (4.1)$$

Para as simulações das redes de sensores e do mecanismo proposto foi utilizada a extensão *diffusion3* da difusão direcionada, presente na distribuição do ns-2 versão 27. Essa extensão foi desenvolvida pelo grupo da USC/ISI no projeto SCADDS (*Scalable Coordination Architectures for Deeply Distributed Systems*) [46]. Foi utilizada a versão 9.0.1 da API da difusão direcionada.

O modelo de energia do ns-2 possui foi utilizado nesse trabalho. Esse modelo de energia é um atributo do nó e representa o nível de energia do mesmo. O nó possui um valor inicial, *initialEnergy_*, que é a energia disponível no início da simulação. As energias utilizadas para a transmissão e recepção de mensagens são nomeadas *txPower_* e *rxPower_*, respectivamente. A energia inicial é decrementada a cada pacote transmitido, conforme as equações 4.2 e 4.3. Os parâmetros utilizados para a configuração do modelo de energia são definidos através da interface em OTcl.

$$Energia = Energia - (P_{tx} * Tempo_{tx}) \quad (4.2)$$

$$Energia = Energia - (R_{rx} * Tempo_{rx}) \quad (4.3)$$

4.2 Cenários e Métricas utilizadas

A topologia da simulação é um campo de sensores disposto na forma de uma *Grid* (grade) modificada para permitir a comunicação na diagonal entre os nós igualmente espaçados. Essa topologia foi escolhida por atender às aplicações de monitoramento de maquinário em pátios industriais ou monitoramento das vibrações em plataformas petrolíferas [47]. Foram construídos dois cenários de simulação. No cenário 1, a grade é composta por 9 nós. No cenário 2, a grade é composta por 100, 144, 196 e 256 nós.

As métricas utilizadas para verificar o desempenho do filtro são:

- rotas descobertas - representa a capacidade do protocolo em descobrir caminhos ou rotas de forma a distribuir o tráfego de dados pelos nós da rede;
- número de transmissões - é o número de transmissões de dados efetuadas por todos os nós da rede;
- energia residual de cada nó - representa a energia residual de cada nó da rede;
- energia residual total da rede - é o somatório da energia residual de todos os nós da rede;
- custo do recebimento das mensagens - é a razão entre a quantidade de mensagens recebidas pelo nó sorvedouro e a energia residual total da rede;
- tempo de vida da rede - definido como o tempo de simulação em que o primeiro nó da rede fica com energia residual nula.

O modelo de energia do simulador foi configurado de modo a caracterizar os nós como MicaMotes. As descrições técnicas mais importantes estão descritas na tabela 4.1.

4.3 Cenário 1 : Grade com 9 nós

Com o intuito de entender o comportamento do filtro, inicialmente é utilizada a grade com 9 nós. Existe um único nó sorvedouro ou coletor, que é o nó 0 (zero), localizado

	Cenário 1	Cenário 2
Quantidade de nó fonte	1	3
Quantidade de nó sorvedouro	1	1
Energia Inicial de todos os nós(J)	0.8	0.8
Potência Consumida para a Transmissão(mW)	660	660
Potência Consumida para a Recepção(mW)	0	0
Alcance efetivo do rádio(m)	30	30
Distância entre os nós	20	20
Tempo de envio de dados Exploratórios(s)	150	150
Tempo de envio de dados(s)	1	1
Variação do tempo (jitter) de envio de dados(s)	0.4	0.4
Timeout do Gradiente(s)	100	100
Quantidade de pacotes no buffer	50	50

Tabela 4.1: Parâmetros das simulações.

no vértice inferior esquerdo e um único nó fonte, nó 8 (oito), localizado no vértice superior direito. O nó 0 dissemina os interesses correspondentes às tarefas de sensoriamento solicitadas pelo usuário, e recebe os dados originados da fonte distante.

Nesse cenário, a simulação durou 900 segundos. Todas as medidas foram calculadas após 20 rodadas de simulação e para um intervalo de confiança 95%, em alguns gráficos as barras de erro são omitidas para maior clareza da figura.

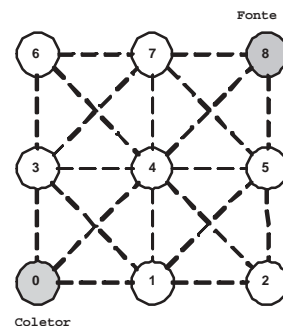


Figura 4.1: Topologia em Grid 3x3.

4.3.1 Análise do protocolo de roteamento para a difusão direcionada

A seguir, são apresentados os valores obtidos para o protocolo para a difusão direcionada original. Cada uma das métricas são analisadas e os resultados, discutidos.

Nr nós	Quantidade de vezes
3	63
4	30
5	3

Tabela 4.2: Difusão pura: Análise das árvores descobertas.

Métrica 1: Análise das árvores de roteamento

Na tabela 4.2 são apresentadas as rotas construídas pelo protocolo original. A primeira coluna mostra a quantidade de nós que compõem a rota e a segunda coluna relaciona a quantidade de vezes durante toda a simulação que aquelas rotas foram construídas. Como pode ser observado, cerca de 65% das rotas descobertas pelo protocolo original têm três nós. Essa rota com 3 nós é compreendida pelos nós 0, 4 e 8, que formam a diagonal. Como foi dito, o protocolo de difusão direcionada original privilegia os caminhos de baixa latência. Em uma topologia em grade como a simulada, a diagonal é o caminho mais utilizado.

Métrica 2: Número de transmissões

A quantidade de transmissões realizadas pela rede está muito relacionada ao número de saltos que compõem as rotas descobertas. Quanto mais saltos tiver maior será o número de transmissões realizadas pelos nós sensores. No protocolo original, durante as 20 rodadas de 900s de duração cada, o número de transmissões de dados realizadas foi de 1132 ± 20.8 .

Métrica 3: Energia residual de cada nó

Os gráficos 4.2(a), 4.2(b) e 4.2(c) mostram a energia residual da rede em 51s, 591s e 891s de simulação, respectivamente. Os eixos x e y representam o posicionamentos dos nós. Foram dispostos 9 nós, separados de 20m. No eixo z é exibido o valor da energia residual individual de cada nó em Joules. A energia inicial de todos os nós é de 0.8J.

Inicialmente, em 51s de simulação, a superfície do gráfico é plana, indicando que os nós estão com o nível de energia residual quase idêntios. Aos 591s de simulação, o

mapa de energia da rede apresenta uma depressão central. Esse gráfico indica que há um consumo excessivo de energia do nó 4 e seus vizinhos. Com o passar da simulação esse nó central será esgotado. O protocolo original iniciará, então, nova fase de descoberta de rotas, inundando novamente a rede. Tal fato, irá levar a um consumo excessivo da rede, conforme pode ser observado no mapa de energia ao fim da simulação, mostrado em 4.2(c).

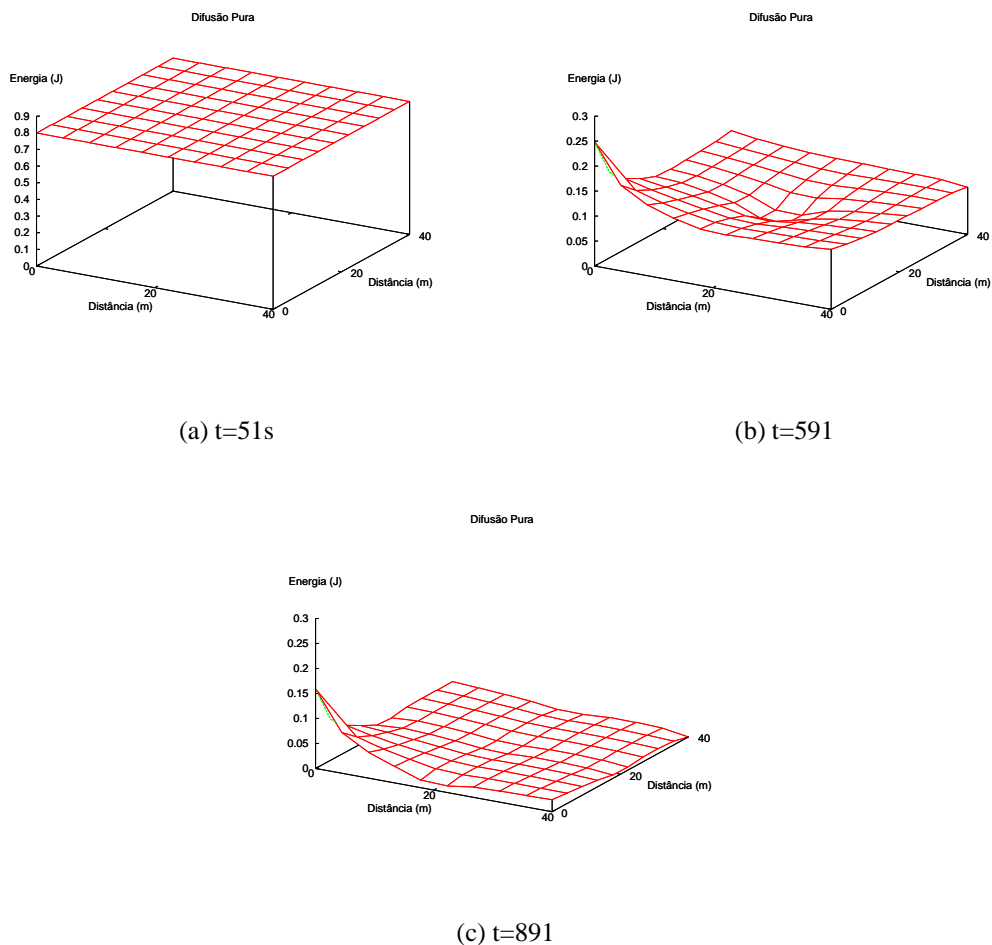


Figura 4.2: Difusão pura: Energia residual dos nós.

Métrica 4: Energia residual total da rede por nó

Na figura 4.3 é mostrada a energia restante em cada nó da rede ao fim da simulação. Observa-se o formato em "V" formando uma pequena depressão no centro do gráfico. Isso denota um consumo excessivo de energia em alguns nós, principalmente nos nós centrais,

e uma reserva maior de energia nos demais nós da rede. O consumo da energia na rede de sensor deve ser balanceado entre os nós, evitando este tipo de formação em "V".

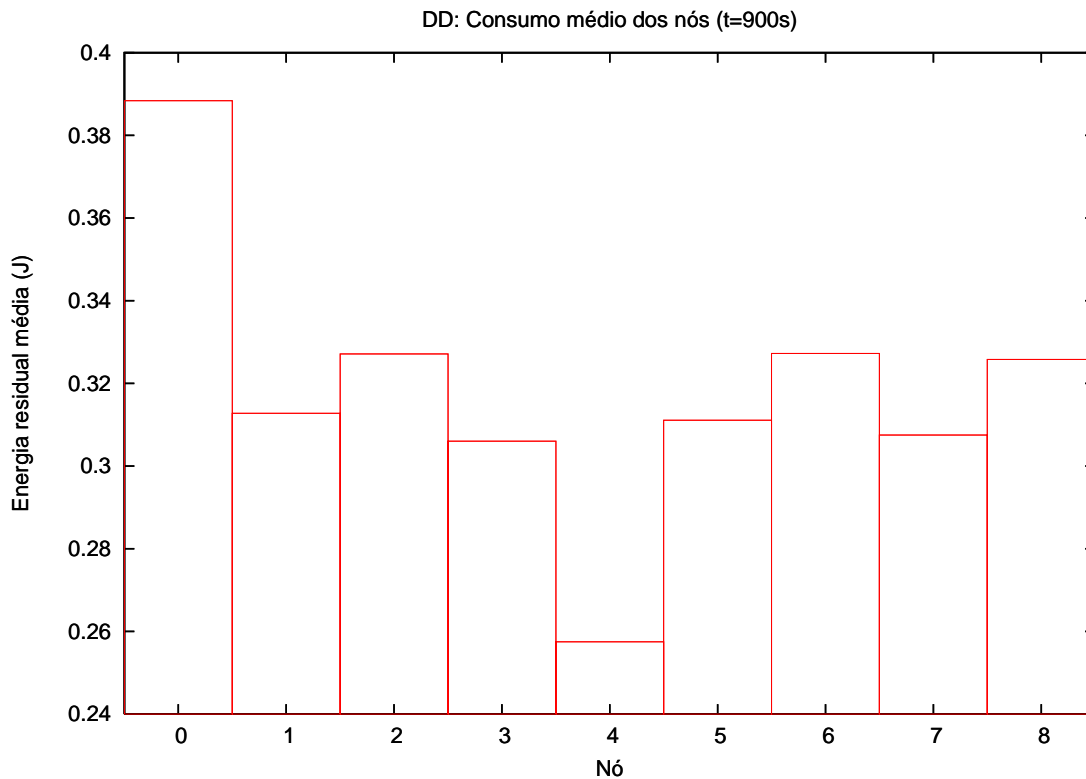


Figura 4.3: Difusão Pura: Energia residual final por nó.

Métrica 5: Consumo total da rede

A seguir, na figura 4.4, é calculado o valor em Joules correspondente a toda energia residual disponível na rede. Esse valor corresponde à soma dos valores de energia residual de cada nó que compõe a topologia.

Métrica 6: Custo do recebimento das mensagens

O custo para o recebimento de mensagens mede quantas mensagens são entregues para cada unidade de energia (J) gasta por todos os nós da rede. Foram recebidas, em média, 460,75 mensagens pelo nó sorvedouro. O consumo final da rede foi de 7.11J. Dessa forma, calcula-se que a rede consegue entregar 64,74 mensagens/J.

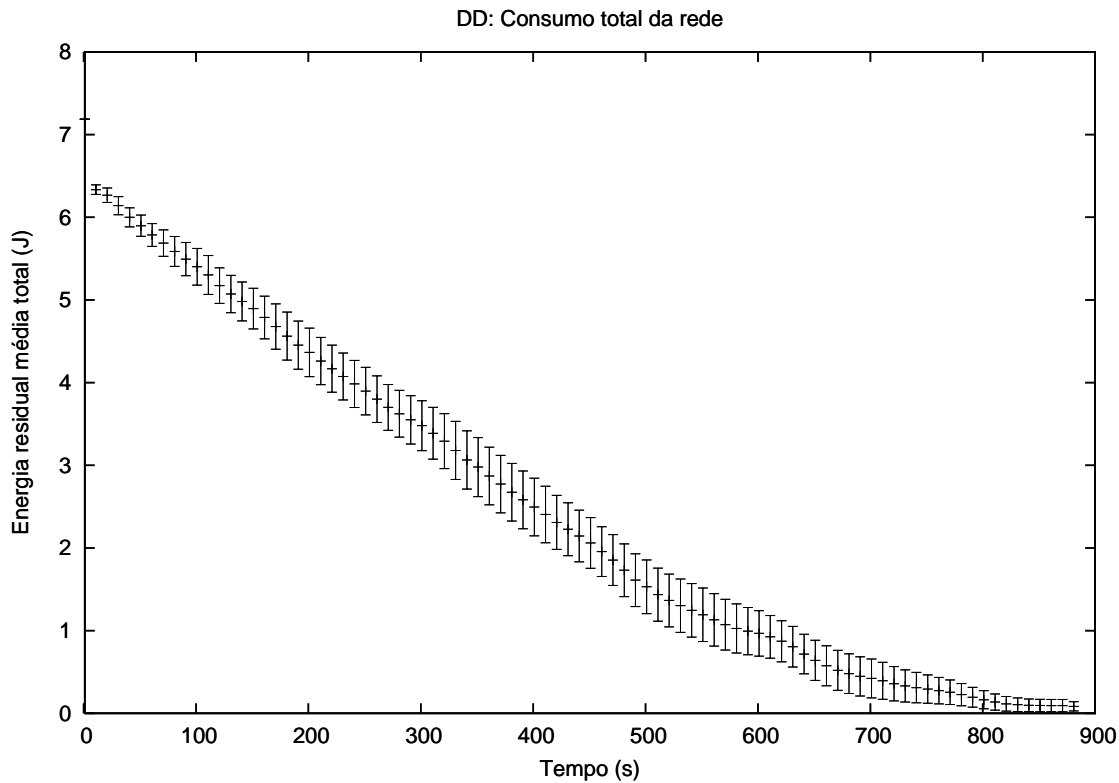


Figura 4.4: Difusão Pura: Energia residual da rede.

Métrica 7: Tempo de vida da rede

O tempo médio em segundos até que o primeiro nó tenha sua energia residual nula é de 500s. Nas aplicações em que todos os nós desempenhem papéis importantes, a perda de um nó significa que a informação coletada pela rede está incompleta, ou até inútil.

4.3.2 Protocolo de Difusão Direcionada com o filtro Energia

Conforme explicado anteriormente, o filtro Energia trata apenas as mensagens Exploratórias. A ideia é de balancear entre todos os nós da rede o consumo de energia. O trabalho apresentado em [48] mostra alguns resultados iniciais desta tese. Neste capítulo será feita uma análise mais detalhada dos ganhos obtidos com a introdução do filtro Energia.

Temporizador(ms)	Nr nós	Qtde de vezes	Temporizador(ms)	Nr nós	Qtde de vezes
50	3	37	500	3	35
	4	45		4	51
	5	8		5	3
100	3	36	1000	3	35
	4	49		4	53
	5	3		5	6
200	3	36	1500	3	35
	4	53		4	48
	5	5		5	4

Tabela 4.3: Filtro Energia: Análise das rotas descobertas.

Métrica 1: Análise das árvores de roteamento

A descoberta das rotas possui duas etapas: a primeira é o envio das mensagens exploratórias pelos nós e a segunda é o envio/recebimento dessas mensagens. O protocolo proposto controla apenas a etapa de envio das mensagens exploratórias. Para o envio/recebimento das mensagens é utilizado o padrão IEEE 802.11. Em um ambiente sem-fio existe por exemplo a questão da disputa pelo acesso ao meio e perdas de pacotes que influenciam no tempo de recebimento das mensagens. Na arquitetura do protocolo difusão direcionada, o impacto está na perda das mensagens exploratórias e consequentemente na descoberta das rotas. Com o temporizador dos nós com valor baixo, o filtro Energia se aproxima do protocolo original pois a quantidade de mensagens exploratórias enviadas/recebidas fica reduzida.

Na tabela 4.3 são apresentadas as rotas construídas após a introdução do filtro Energia. A primeira coluna exibe o valor do temporizador, a segunda, a quantidade de nós que compõem as rotas construídas e a terceira coluna relaciona a quantidade de vezes durante toda a simulação que aquelas rotas foram construídas. Após a introdução do filtro Energia, as rotas construídas passaram a ter um maior número de saltos. O filtro Energia passa a indicar as rotas com 4 saltos, ao invés de 3 saltos. Quando o temporizador possui o valor de 200ms, 56% dos caminhos descobertos possuem 4 saltos. A partir desse valor, se aumentarmos o temporizador, os ganhos obtidos para a descoberta de caminhos mais longos são pequenos. Além disso, o atraso no descobrimento das rotas passa a ser maior, conforme ilustrado em 4.5.

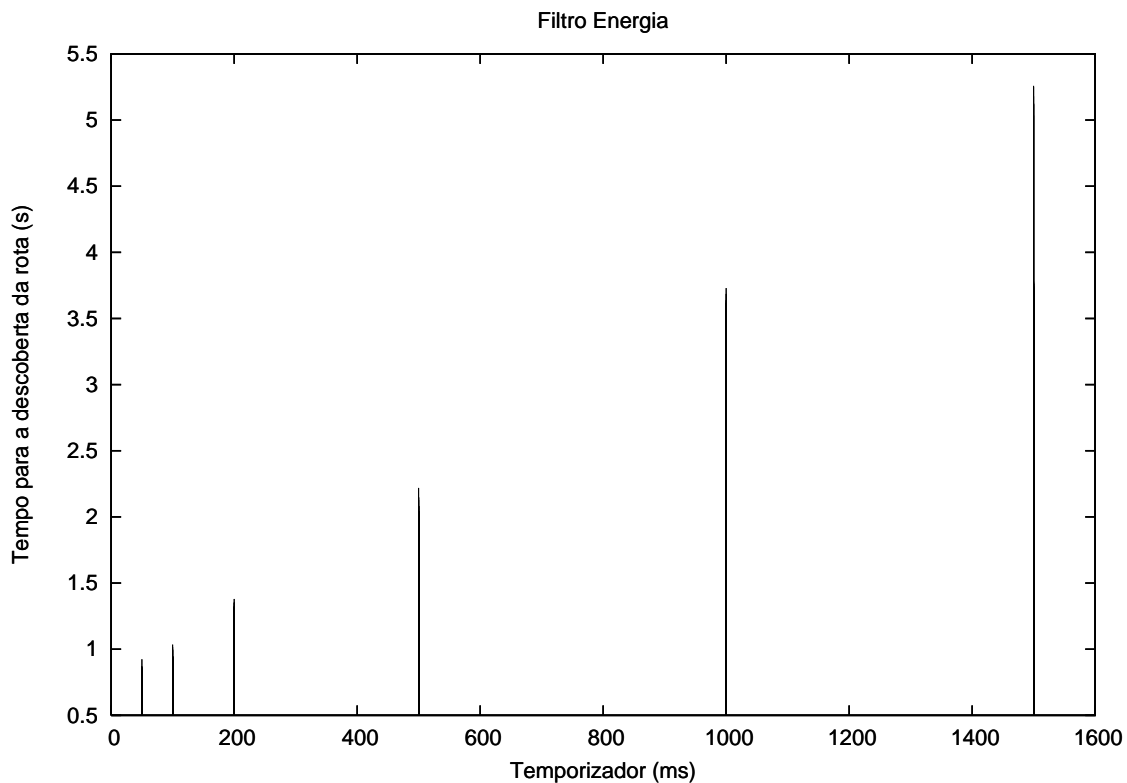


Figura 4.5: Filtro Energia: Tempo de descoberta das rotas.

Métrica 2: Número de transmissões

A figura 4.6 mostra a variação no número de transmissões efetuadas pela rede para a transmissão dos dados. Pode-se observar que a variação do temporizador dos nós não influenciou significativamente na variação do número de transmissões de dados realizadas. Se compararmos o valor anterior a introdução do filtro Energia, que era uma média de 1132, percebe-se um pequeno aumento no número de transmissões de dados. Esse pequeno aumento pode ser justificado se observarmos que o número de nós que passaram a compor as rotas aumentou em 1(um), que é um valor pequeno.

Métrica 3: Energia residual de cada nó

No início da simulação, em 51s, o mapa da energia residual da rede com o filtro Energia é bem parecido ao mapa do protocolo original. Esse gráfico não sofre variação com o aumento do temporizador, conforme mostrado em 4.7(a) com o temporizador de

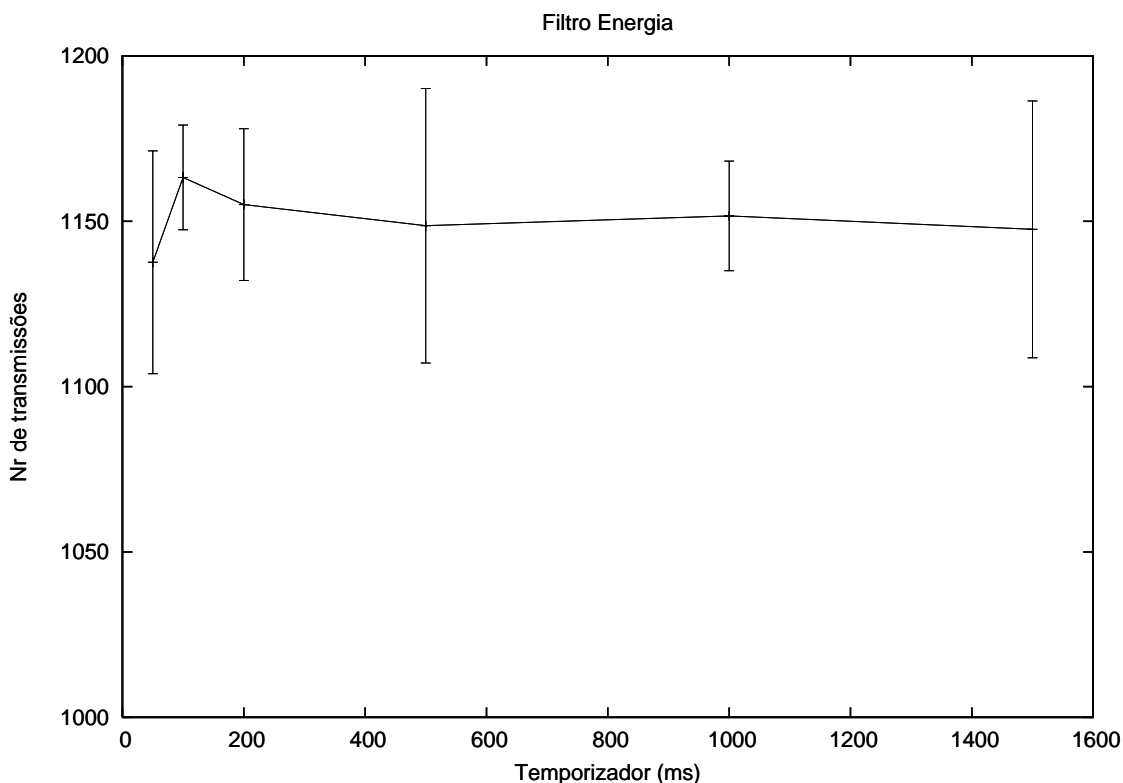


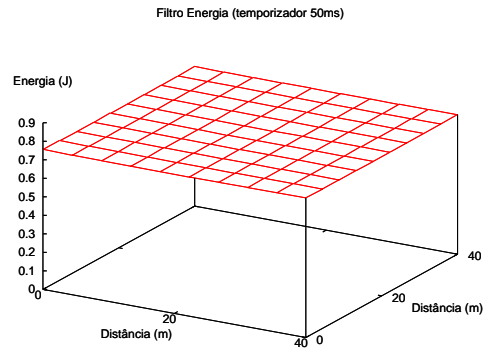
Figura 4.6: Filtro Energia: Número de transmissões de dados.

50ms e em 4.7(b) onde o temporizador é de 1500ms.

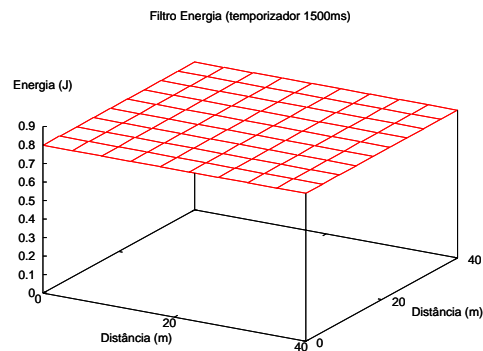
De modo análogo, no fim da simulação, em 891s, o mapa com ou sem o filtro Energia também são parecidos. Esse mapa também sofre variação pequena com o aumento do temporizador, conforme são mostradas nas figuras 4.8(a) e 4.8(b).

A seguir, em 4.9, serão apresentadas as distribuições energéticas da rede em 591s, um pouco após a metade da simulação. São exibidas as distribuições energéticas da rede variando o temporizador em 50ms, 100ms, 200ms, 500ms, 1000ms e 1500ms.

Novamente, não são observadas grandes diferenças entre os mapas de energia apresentados quanto à variação do temporizador. Entretanto, se compararmos os mapas da energia da rede após a introdução do filtro Energia com o mapa do protocolo original, percebem-se diferenças. A parte central do mapa original é muito mais acentuada do que o mapa obtido com o filtro Energia. Essa parte central da rede ainda é bastante utilizada, mesmo após a introdução do filtro Energia. A utilização do nó central é explicada, pois o critério utilizado para o balanceamento do consumo de energia utiliza a distância entre



(a) temporizador 50ms

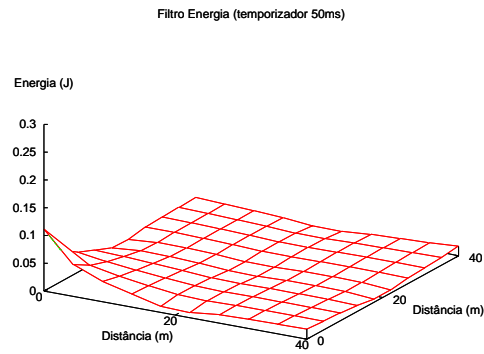


(b) temporizador 1500ms

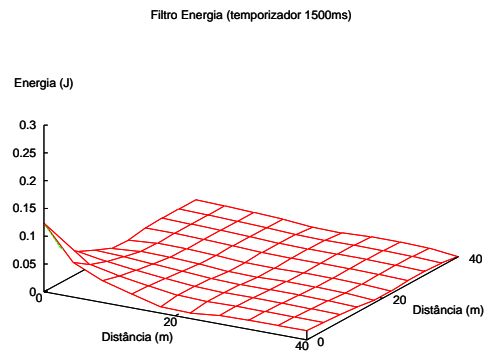
Figura 4.7: Filtro Energia: Energia residual dos nós em 51s.

a fonte e o destino. Se considerarmos a razão entre energia residual e número de saltos entre os nós fonte e destino, as árvores descobertas incluem os nós centrais.

As bordas da superfície do mapa de energia mostrado na figura 4.9 são um pouco mais baixas do que as bordas da superfície do protocolo original. Isso indica que os nós 2 e 6, que compõem os vértices da topologia, são mais utilizados para o encaminhamento das mensagens. Com isso, o mapa de energia obtido é mais plano, pois ocorre o balanceamento do consumo de energia entre os nós.

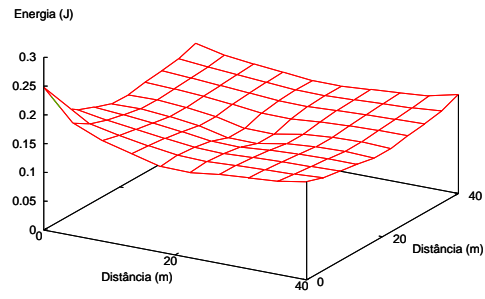


(a) temporizador 50ms

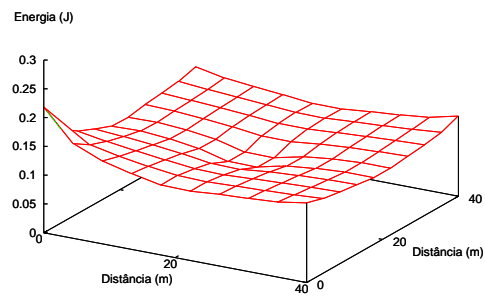


(b) temporizador 150ms

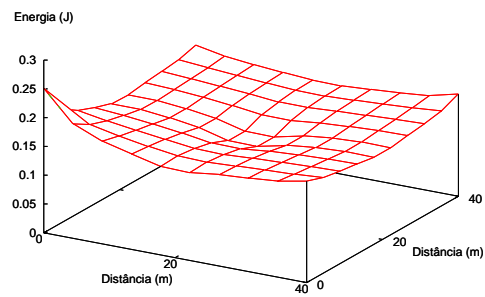
Figura 4.8: Filtro Energia: Energia residual dos nós em 891s.



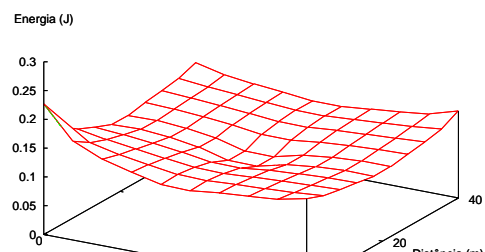
(a) temporizador 50ms



(b) temporizador 100ms



(c) temporizador 200ms



Métrica 4: Energia residual total da rede por nó

Com o temporizador em 200ms, observa-se na figura 4.10 que há uma distribuição melhor da energia consumida por nó. O gráfico do protocolo original tinha um formato em "V", que com o filtro Energia deixa de existir, como era o esperado.

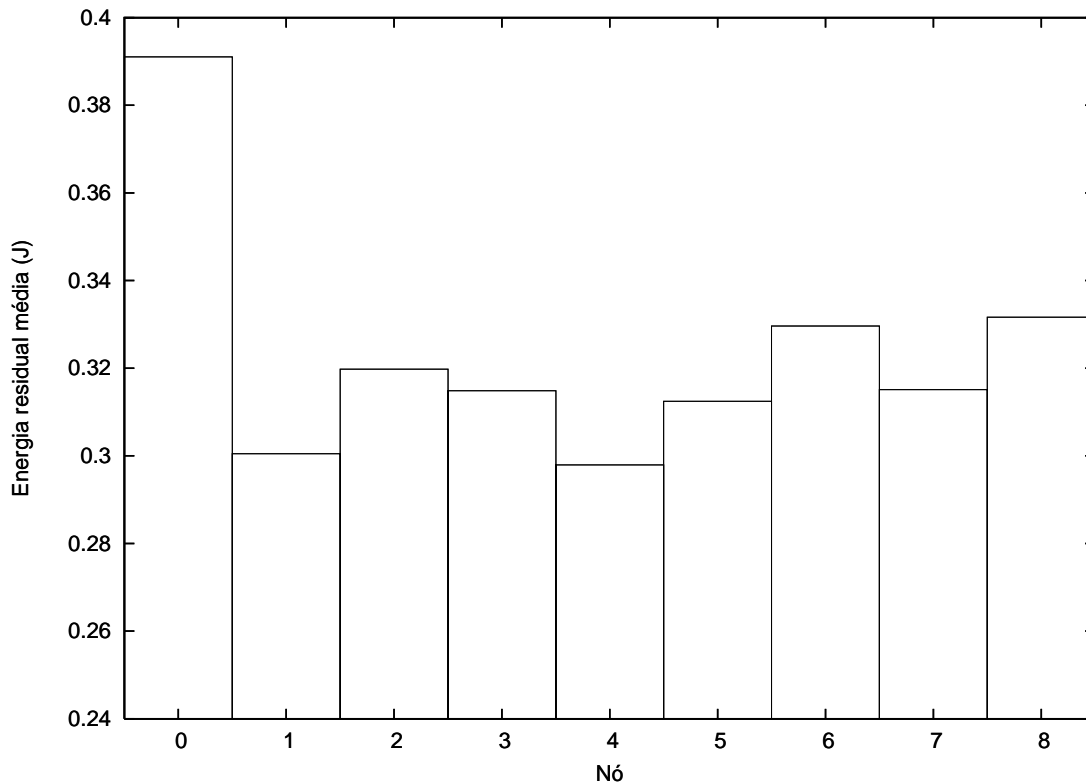


Figura 4.10: Filtro Energia: Energia residual por nó (temporizador 200ms).

Métrica 5: Consumo total da rede

Na figura 4.11, será mostrada a energia residual total da rede. O valor exibido corresponde à soma das energias residuais de todos os nós da rede. A variação do temporizador instanciado nos nós pouco influenciou no valor da energia residual total da rede.

Na figura 4.12 pode se observar que a energia residual total disponível na rede é maior após a introdução do filtro Energia. O protocolo para a difusão direcionada original utiliza em excesso o nó central 4 o que provoca a sua exaustão precoce. A detecção da perda do nó não é tão imediata, levando à transmissão de dados que são perdidos por falta de

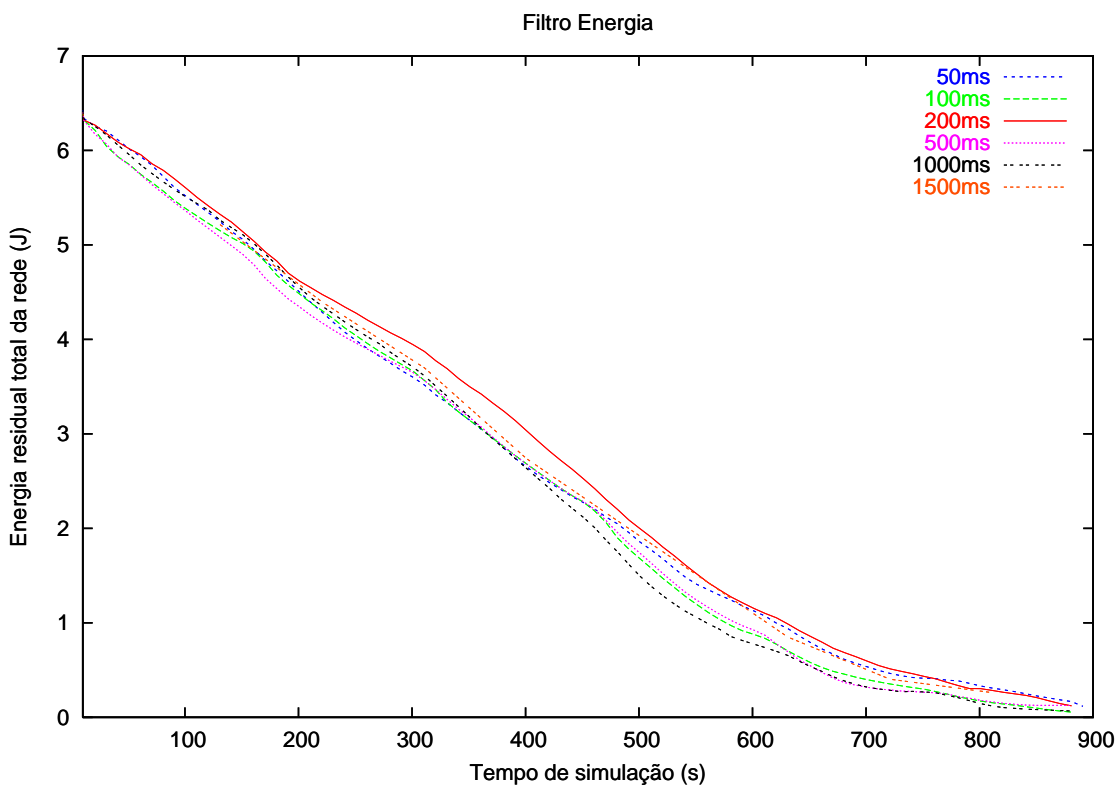


Figura 4.11: Filtro Energia: Energia residual da rede.

rota. Com o intuito de reparar o caminho, o protocolo original inunda a rede com várias mensagens exploratórias novas, elevando o consumo de energia dos nós. Como o filtro energia retarda a morte dos nós, é reduzida o número de mensagens exploratórias para reparar os caminhos de dados.

Métrica 6: Custo do recebimento das mensagens

O custo para o recebimento de mensagens mede quantas mensagens são entregues para cada unidade de energia (J) gasta por todos os nós da rede. A quantidade de mensagens recebidas pelo nó sorvedouro está na figura 4.13.

O custo para o recebimento das mensagens foi calculado e está na figura 4.14. O valor obtido é próximo ao custo do protocolo original. Após a introdução do filtro Energia, houve um decréscimo na quantidade de mensagens entregues no mesmo espaço de tempo da simulação. O filtro Energia aumentou o número de saltos necessários para que a mensagem chegue ao destino e introduz um atraso no estabelecimento de novas rotas, o que

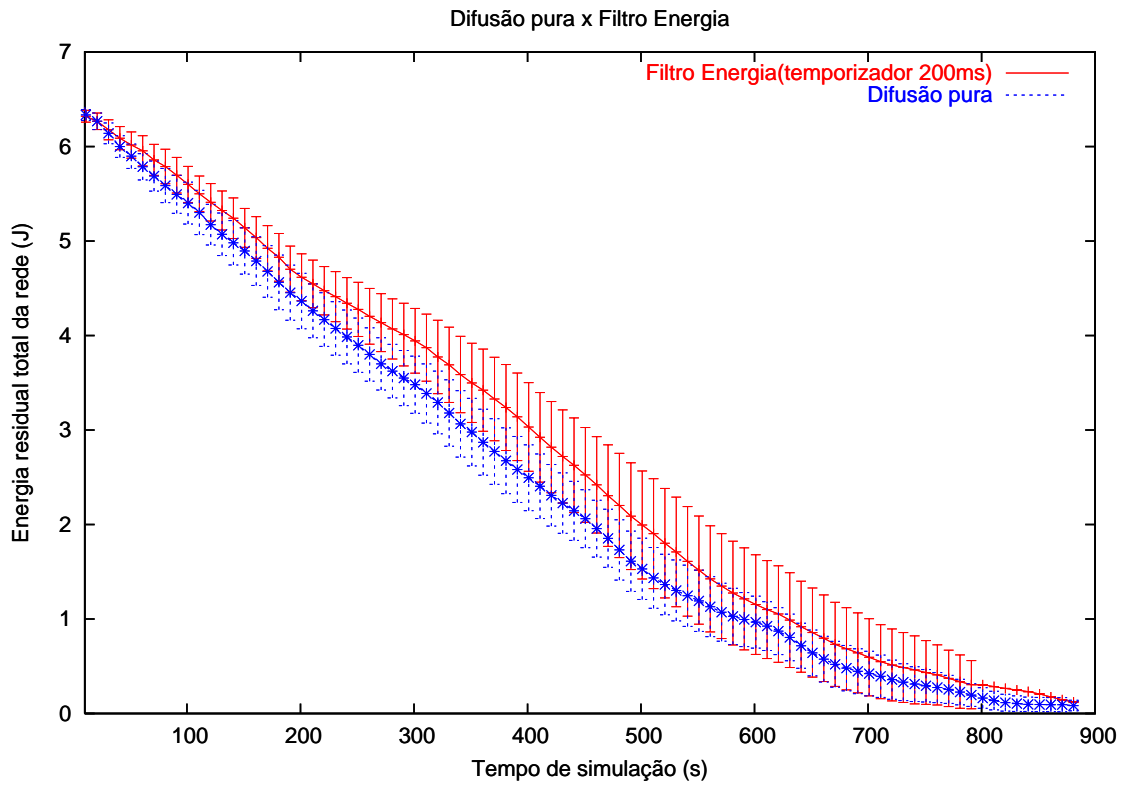


Figura 4.12: Filtro Energia x Difusão pura: Energia residual da rede.

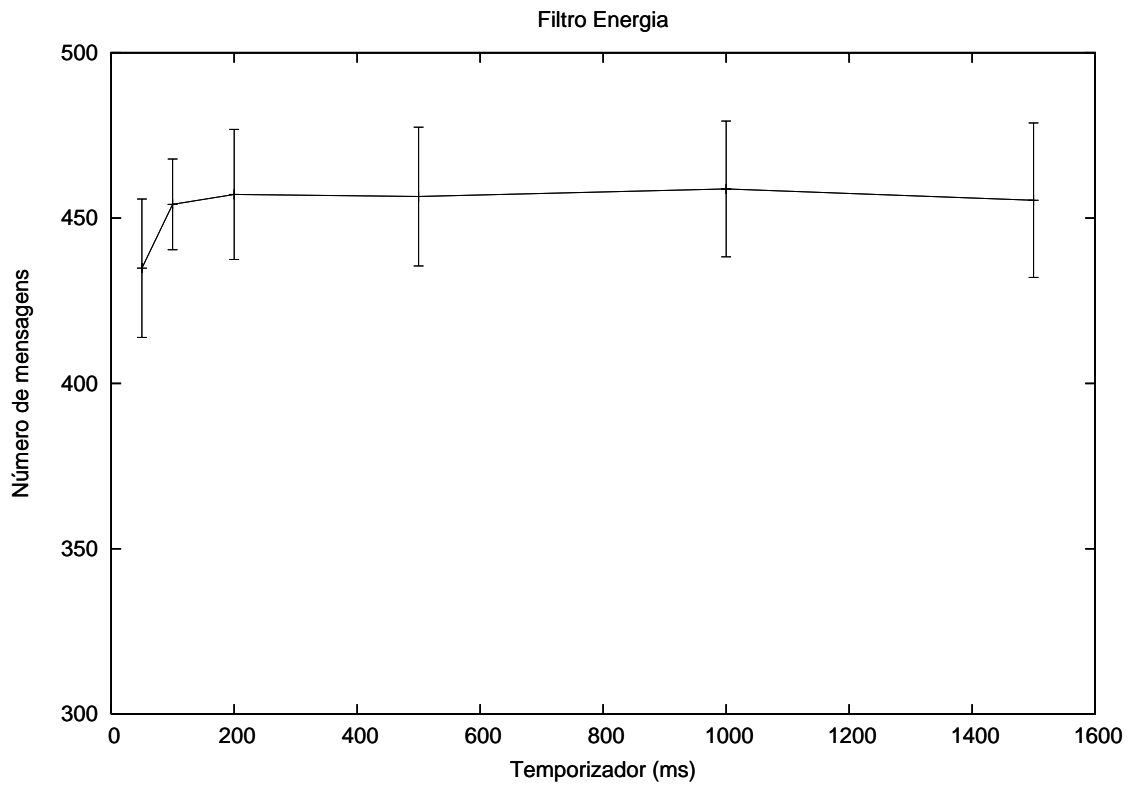


Figura 4.13: Filtro Energia: Quantidade de mensagens de dados recebidas.

provocou tal redução de mensagens. Apesar disso, diminuiu o consumo total da rede, o que compensou o custo do recebimento das mensagens.

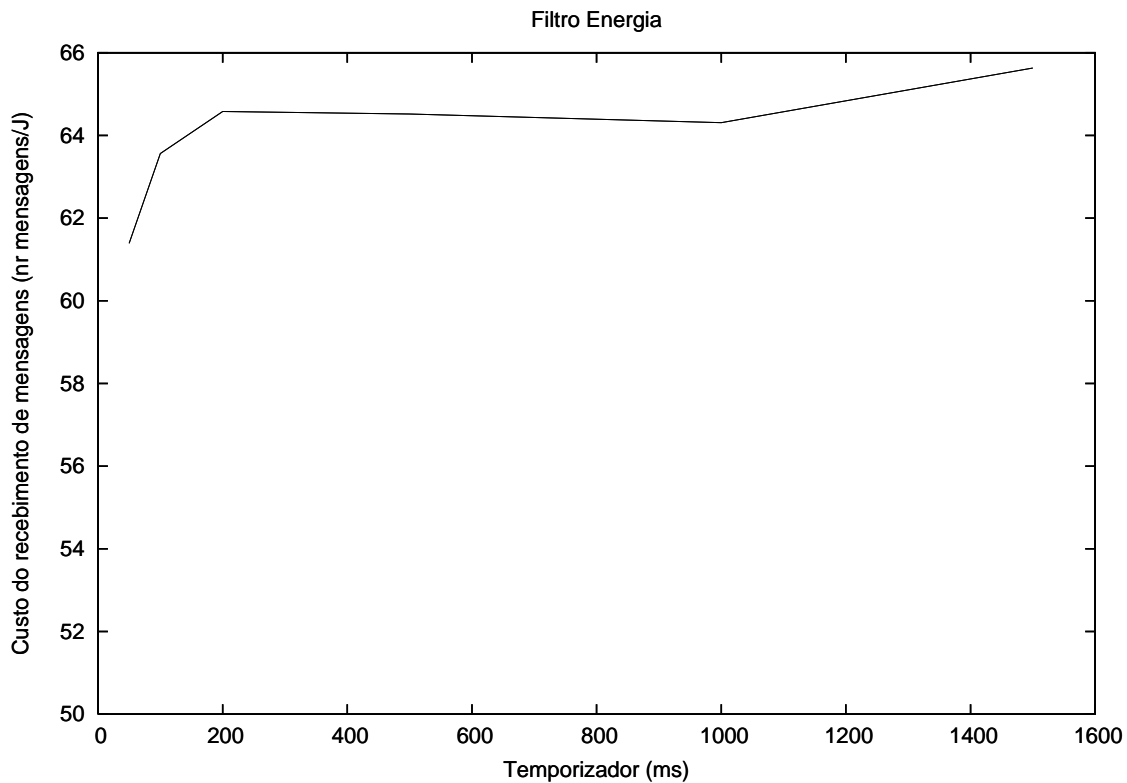


Figura 4.14: Filtro Energia: Custo do recebimento das mensagens.

Métrica 7: Tempo de vida da rede

Em 4.15 tem-se o gráfico do tempo de vida da rede. O tempo de vida da rede foi definido como o tempo transcorrido até que o primeiro nó tivesse sua energia residual nula. O tempo de vida da rede ficou em torno de 600s, um aumento de 20%.

4.4 Cenário 2 : Grades com maior número de nós

As próximas simulações aumentam o número de nós sensores. São feitas simulações para topologias em grade de dimensão 10x10, 12x12, 14x14 e 16x16, com 100, 144, 196 e 256 nós, respectivamente. São colocados três nós fontes e um nó coletor das informações. Os nós fontes são os nós sensores numerados como 99, 95 e 59 - grade 10x10, 143, 137 e

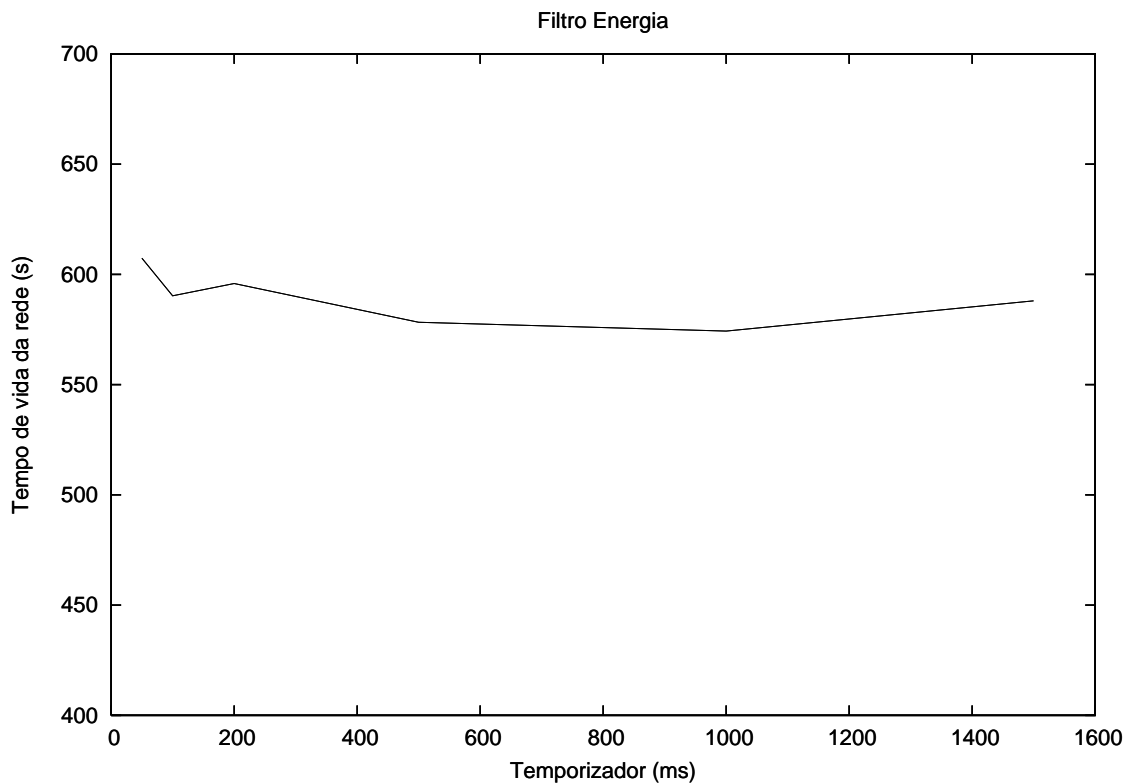


Figura 4.15: Filtro Energia: Tempo de vida da rede.

83 - grade 12x12, 195, 188 e 111 - grade 14x14, 255, 247 e 143 - grade 16x16. São feitas análises de desempenho do protocolo original e comparados os resultados obtidos com o filtro Energia. A topologia da grade 10x10 é mostrada na figura 4.16.

e nas topologias com maior número de nós a simulação durou 600s. Os resultados que serão apresentados a seguir representam um cenário com maior número de nós. As simulações duraram 600s e foram executadas 20 rodadas. O filtro Energia foi instanciado em todos os nós e o temporizador utilizado foi de 200ms, pois foi o que teve melhor desempenho no cenário anterior com 9 nós. A distância entre os nós permanece de 20m. As mensagens exploratórias continuam sendo enviadas a cada 150s.

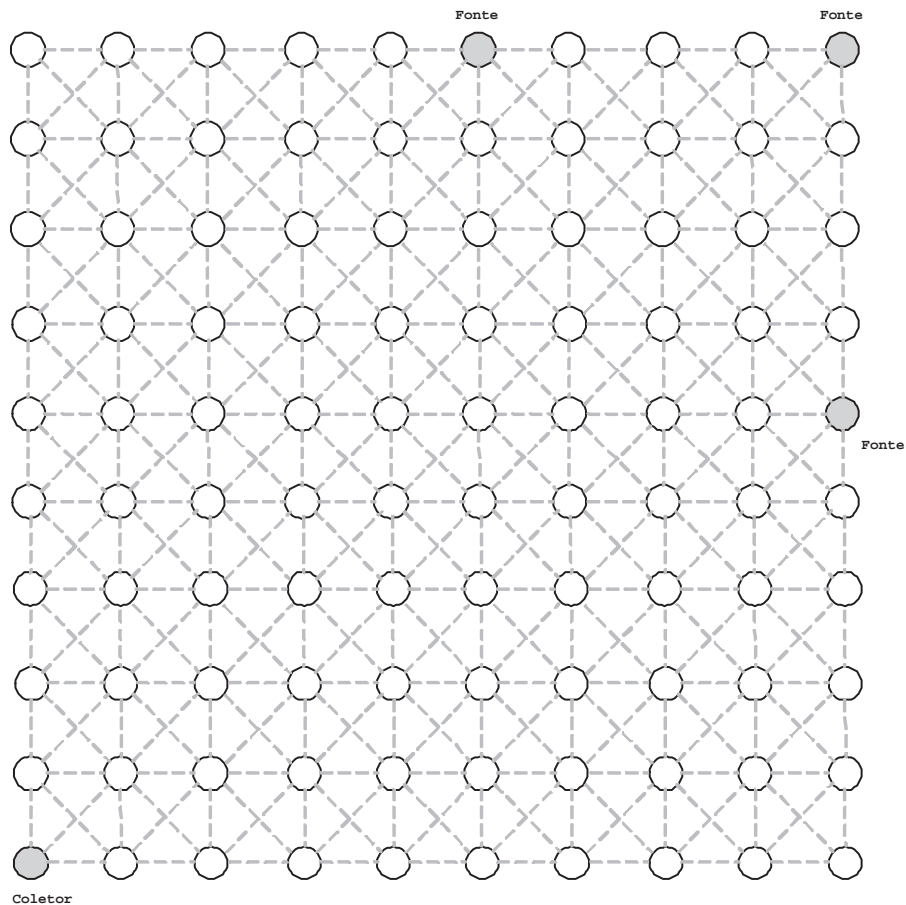


Figura 4.16: Topologia em Grid 10x10.

4.4.1 Comparativo entre o protocolo original e o Filtro Energia

Métrica 1: Número de transmissões

Na figura 4.17, é mostrado o número de transmissões de dados efetuado entre os nós fontes e o nó sorvedouro. Como era de se esperar, o aumento do número de nós provoca um aumento no número de transmissões. O número de transmissões de dados efetuado com a introdução do filtro Energia foi maior que no protocolo original porque os caminhos utilizados para o roteamento são mais longos.

Métrica 2: Energia residual de cada nó

Nas figuras 4.18 e 4.19 a seguir serão mostrados os mapas de energia da rede aos 51s de simulação com intervalo de confiança 95%. Pode-se observar em 4.18(b) que

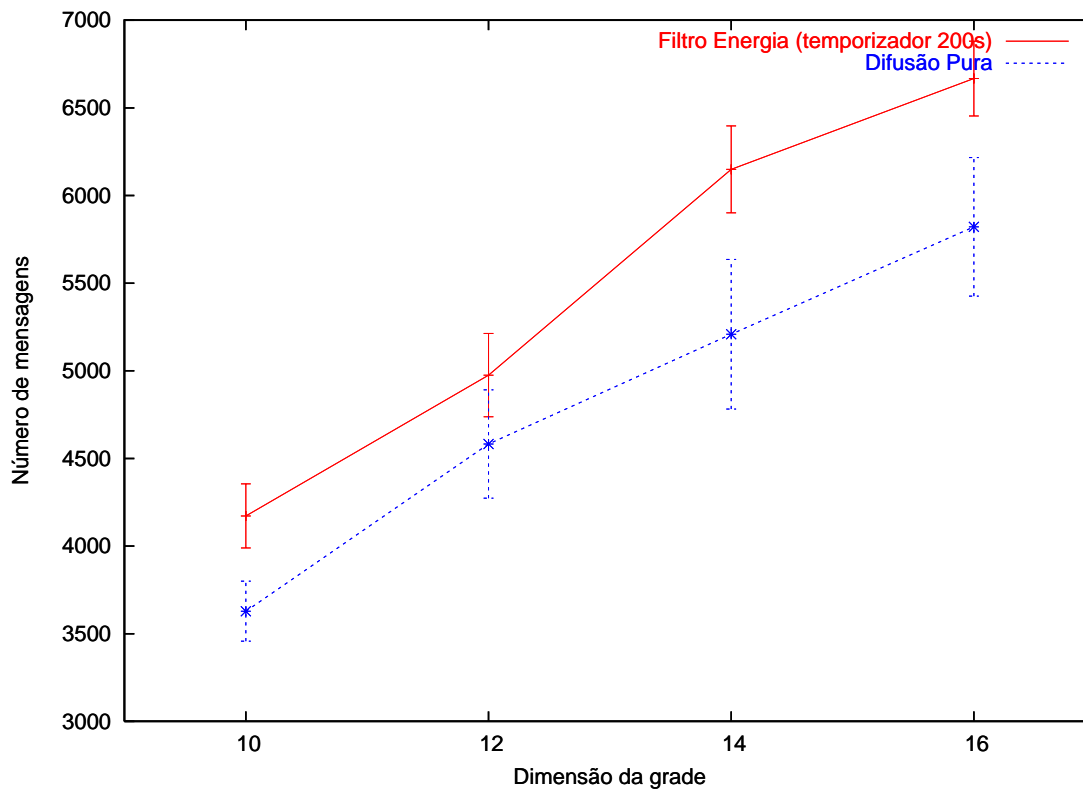


Figura 4.17: Comparativo Difusão pura x Filtro Energia: Número de transmissões de dados.

a superfície do gráfico com o filtro Energia é mais elevada na parte central do que a superfície do gráfico com o protocolo original (figura 4.18(a)). Como era de se esperar, o protocolo original privilegia os caminhos de menor retardo. Por isso existe a depressão na parte central da superfície desenhada.

Métrica 3: Energia residual total da rede por nó

Os gráficos exibidos em 4.20 e 4.21 mostram o valor da energia residual de cada nó que compõe a topologia no final da simulação. Como o protocolo original varia pouco as rotas utilizadas para o encaminhamento das mensagens, a tendência do gráfico é a de possuir formações em "V". Por exemplo, na topologia em grade de 10x10 nós, os nós fontes foram os nós 99, 95 e 59. Observa-se que o protocolo original, mostrado na figura 4.20(a), prefere alguns nós em detrimento de outros. Enquanto que, com o filtro Energia, na região do gráfico entre os nós 90 e 99, por exemplo, há uma linha quase reta,

indicando que houve a distribuição de energia entre os nós, figura 4.20(b). Além disso, pode-se observar que, na topologia em grade 12x12, o valor da energia residual dos nós com a utilização do protocolo original, figura 4.20(c), é mais baixo que o valor da energia residual após a utilização do filtro Energia, figura 4.20(d). Nos gráficos gerados para as topologias em grade 14x14, figuras 4.21(a) e 4.21(b), e 16x16, figuras 4.21(c) e 4.21(d) os mesmos resultados podem ser observados.

Métrica 4: Custo do recebimento das mensagens

A figura 4.22 mostra a quantidade de mensagens recebidas pelo nó sorvedouro com e sem o filtro Energia. Conforme definido, o custo do recebimento das mensagens é a razão entre a quantidade de mensagens recebidas pelo nó sorvedouro e a energia residual total da rede. Analisando a figura 4.23, observa-se que a razão entre a quantidade de mensagens coletas e a energia consumida pela rede diminui à medida que há o aumento do número de nós. Isso se deve ao fato de haver um aumento no número de transmissões de dados de acordo com os resultados mostrados anteriormente.

Além disso, o filtro Energia consegue entregar mais mensagens por unidade de energia consumida pela rede. Apesar de aumentar o número de saltos entre as fontes e o destino, o filtro reduz as mensagens perdidas durante o tempo entre o esgotamento do nós e a descoberta de uma nova rota.

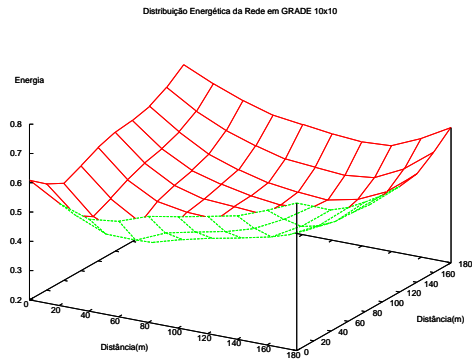
Métrica 5: Tempo de vida da rede

Na figura 4.24 é exibida a variação no tempo de vida da rede em função do aumento da dimensão da grade da topologia. Na topologia simulada, o tempo de vida da rede com a utilização do protocolo original diminui à medida que aumenta o número de nós da rede. Isso se deve ao fato de haver um aumento excessivo nas inundações feitas pelos nós para a descoberta de rotas. Com o filtro Energia, o aumento do número de nós favorece a distribuição do consumo de energia porque aumenta a possibilidade de caminhos de roteamento. O aumento no tempo de vida da rede chega a quase 60% na topologia com 256 nós.

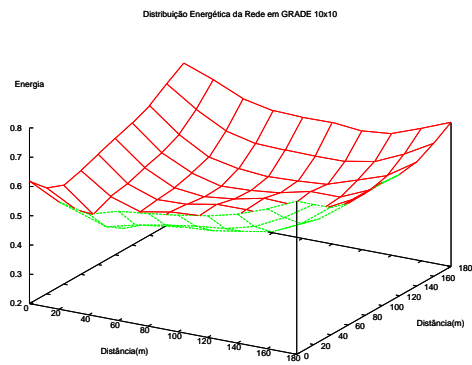
4.5 Comentários

Através das simulações realizadas tanto com poucos nós quanto com maior quantidade de nós, observam-se ganhos após a utilização do filtro Energia. Com a distribuição do consumo de energia por diversos caminhos da rede, consegue-se reduzir as perdas de mensagens entre o esgotamento energético de um nó e a descoberta de um caminho alternativo. Há, portanto, um aumento no número de mensagens recebidas pelo sorvedouro.

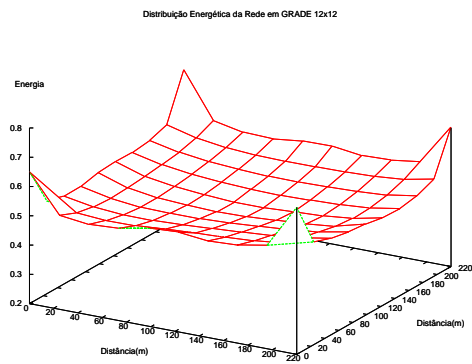
Outra métrica importante foi o aumento da ordem de 20% no tempo transcorrido até que o primeiro nó tenha seu nível de energia nulo. O aumento no tempo de vida da rede foi conseguido através da distribuição do consumo da energia entre os nós, o que foi verificado através da análise das rotas descobertas.



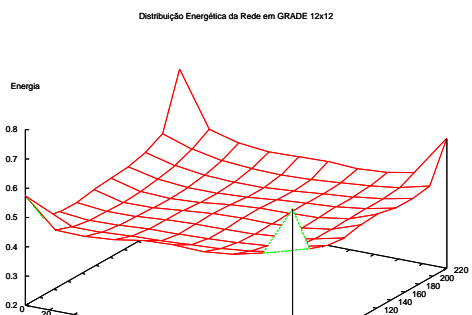
(a) Grade 10x10 - Difusão original

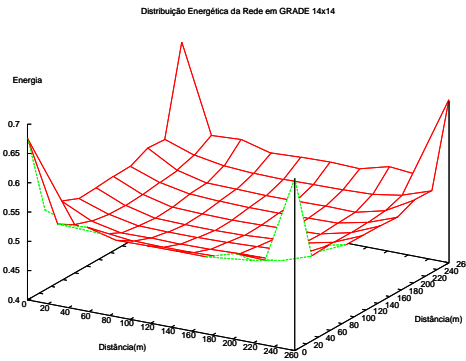


(b) Grade 10x10 - Filtro Energia

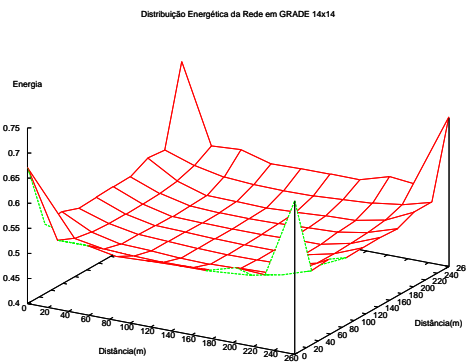


(c) Grade 12x12 - Difusão original

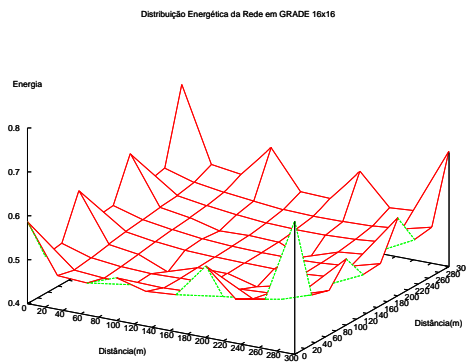




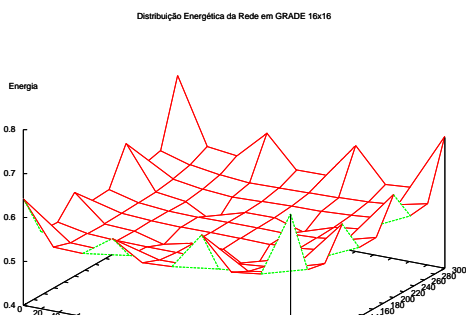
(a) Grade 14x14 - Difusão original

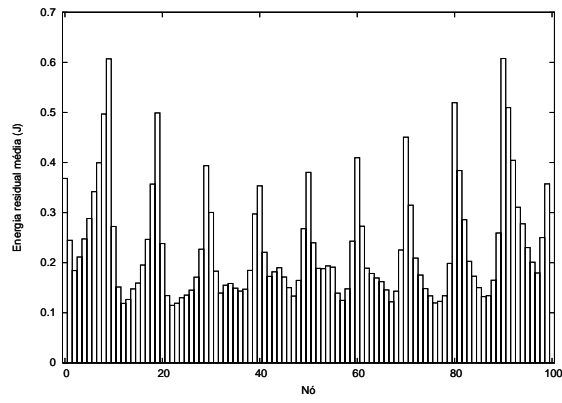


(b) Grade 14x14 - Filtro Energia

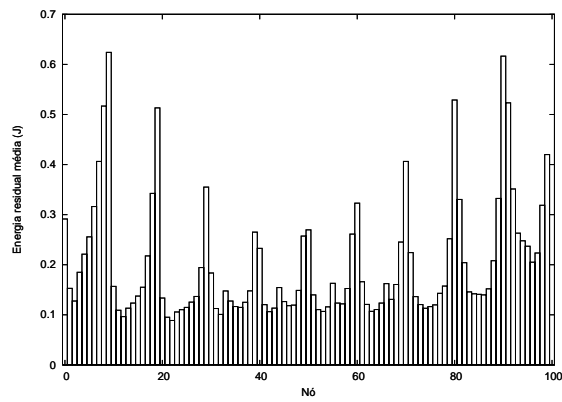


(c) Grade 16x16 - Difusão original

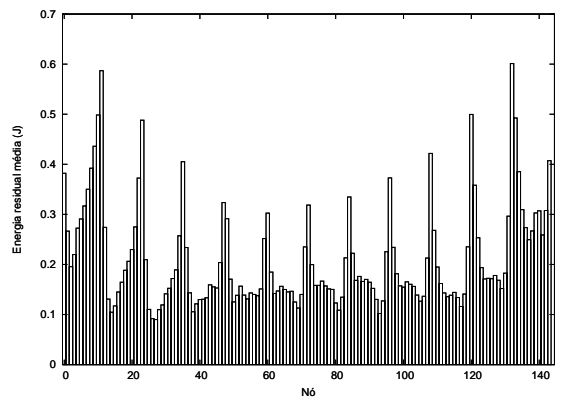




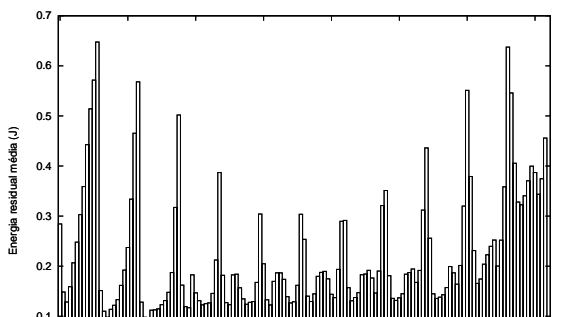
(a) Grade 10x10 - Difusão original

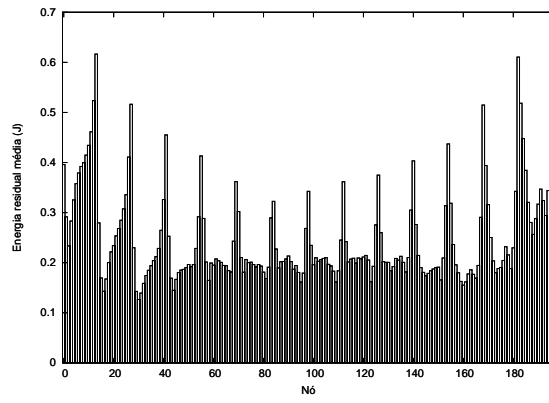


(b) Grade 10x10 - Filtro Energia

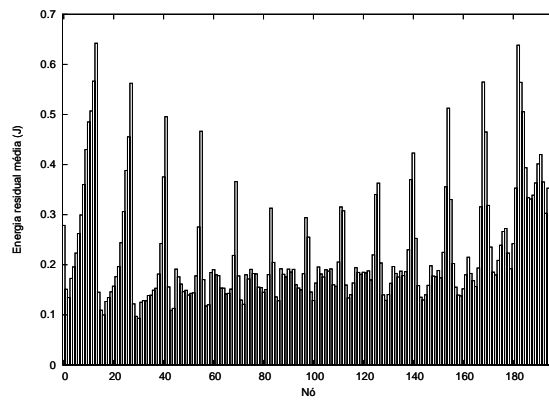


(c) Grade 12x12 - Difusão original

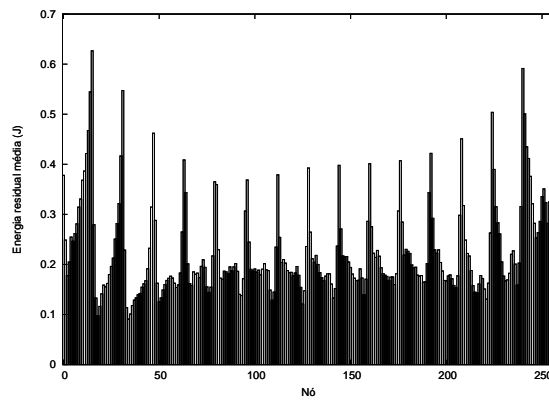




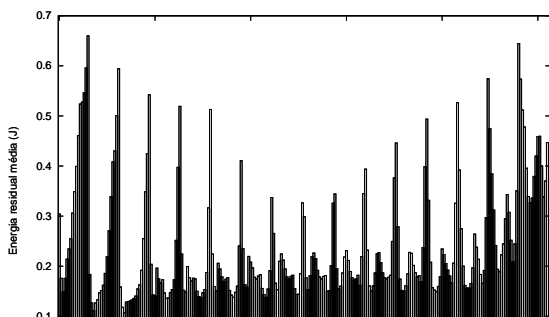
(a) Grade 14x14 - Difusão original



(b) Grade 14x14 - Filtro Energia



(c) Grade 16x16 - Difusão original



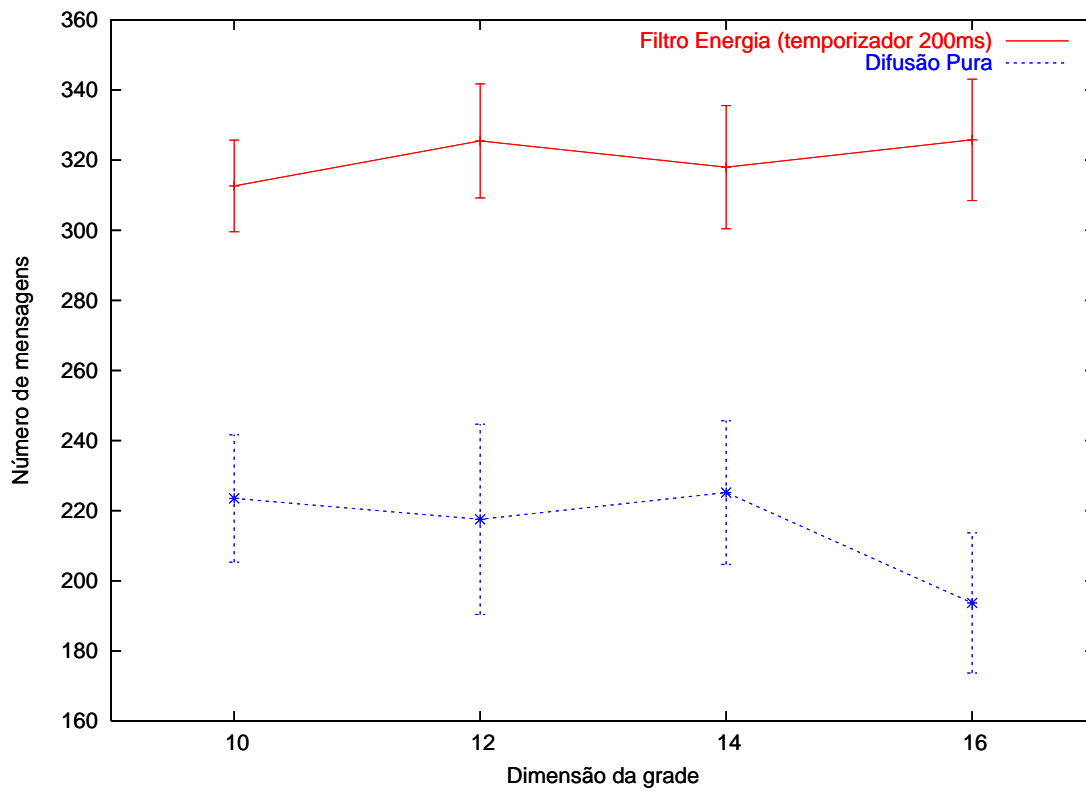


Figura 4.22: Comparativo Difusão pura x Filtro Energia: Número de mensagens recebidas.

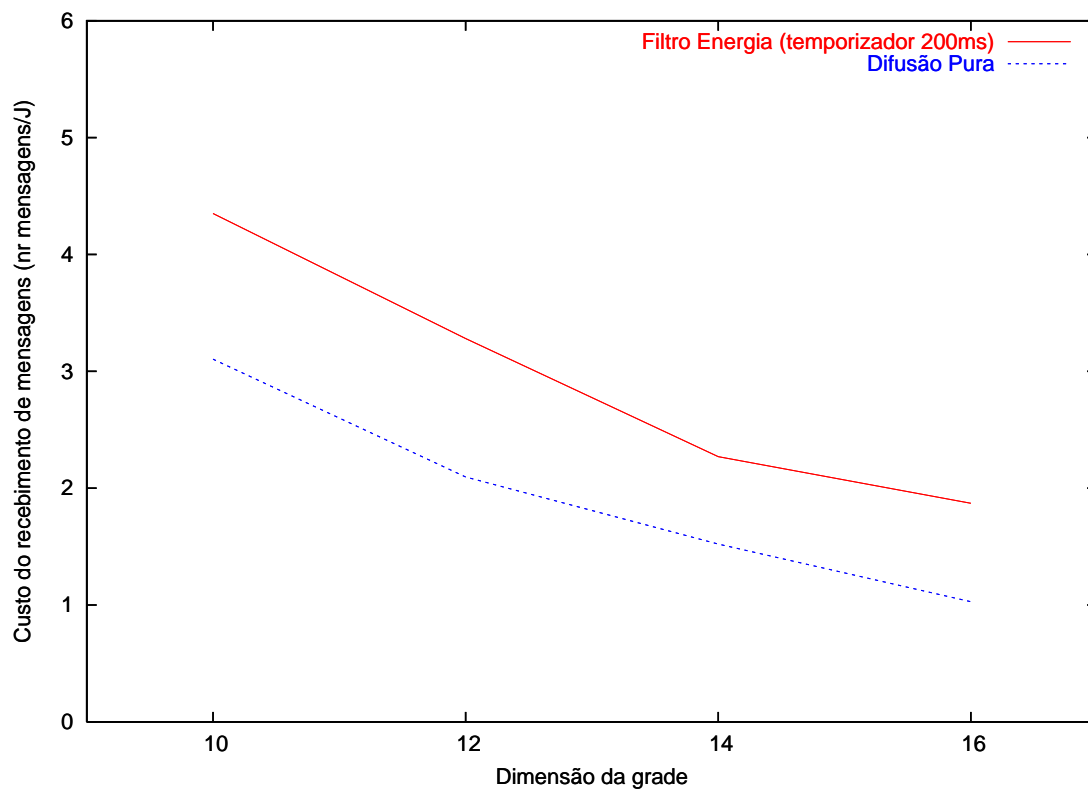


Figura 4.23: Comparativo Difusão pura x Filtro Energia: Custo do recebimento das mensagens.

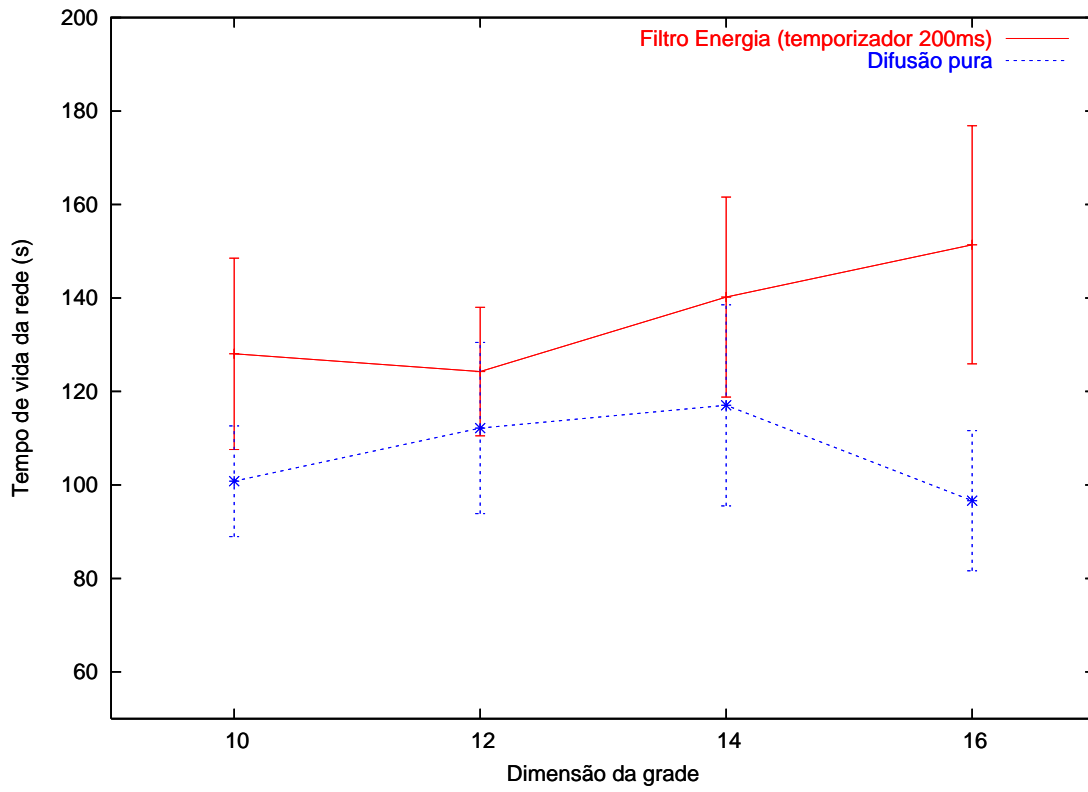


Figura 4.24: Tempo de vida da rede.

Capítulo 5

Conclusões e Trabalhos Futuros

PROTOCOLOS de roteamento para as RSSFs devem possuir mecanismos que se adequem às fortes restrições de energia dos nós. Dependendo da aplicação, algumas métricas de desempenho dos protocolos são mais importantes do que outras. Quando o tempo de sobrevivência da rede é a métrica de interesse, os protocolos de roteamento que utilizam como métrica a energia residual proporcionam ganhos para a rede.

A principal contribuição deste trabalho foi a proposta de um mecanismo de roteamento que altera o protocolo original para a escolha das rotas. Para os cenários simulados houve um aumento no tempo de vida da rede e um aumento no número de mensagens recebidas. A seção 5.1 apresenta as conclusões da tese e na seção 5.2 são feitas algumas sugestões de trabalhos futuros.

5.1 Conclusões

Dentre os protocolos de roteamento propostos para as RSSFs, a Difusão Direcionada recebe destaque por ser um protocolo centrado nos dados e relativamente simples para operação. Além disso, esse protocolo utiliza apenas interações locais para a escolha das rotas e é reativo. Essas características evitam com que haja o desperdício de energia entre os nós para o prévio estabelecimento das rotas. Por isso, a difusão direcionada é bastante adequado para a maioria das aplicações em RSSFs.

O mecanismo de roteamento proposto implementa uma modificação no protocolo difusão direcionada. Entretanto, foram mantidas as características originais do protocolo como o uso de interações locais. De início, o comportamento do mecanismo proposto não difere do protocolo original. No entanto, com a diminuição do nível energético da rota de menor retardo, há a escolha de novas rotas. Com isso, consegue-se o aumento na sobrevida da rede.

O temporizador introduzido nos nós influencia na capacidade da rede em encontrar uma maior quantidade de rotas. O temporizador implementado é diretamente proporcional à distância em saltos entre os nós fonte e destino. Quanto mais longe o nó se encontra da fonte, maior é o seu temporizador. Essa abordagem é melhor do que o uso de um temporizador constante pela rede, pois permite que nós mais próximos do sorvedouro reúnam maiores informações sobre os níveis de energia dos caminhos. Desta forma, permite que o nó sorvedouro, que irá determinar qual rota utilizar, tenha mais opções de escolha.

A distribuição do tráfego pelos nós da rede utilizando como critério o somatório da energia residual do caminho se mostrou uma métrica interessante para a distribuição do tráfego. Esse critério foi utilizado para se manter as características do protocolo em ter somente interações locais. Através do acumulador de energia do caminho pode-se, de certa forma, ter um conhecimento da rede como um todo, mesmo que de forma parcial. O acumulador de energia não foi utilizado como único critério para evitar que caminhos muito longos tivessem maior peso do que caminhos mais curtos. O uso de caminhos longos demais leva ao esgotamento da energia da rede, uma vez que aumenta em excesso o número de transmissões necessárias para que os pacotes atinjam o seu destino.

Através do uso da energia residual do caminho ponderada com a distância em saltos entre os nós origem e destino das informações conseguiu-se um aumento de até 60% no tempo de vida da rede na topologia com 256 nós. O número de mensagens entregues no destino teve um aumento de mais de 50%. Esses aumentos se justificam pois a utilização dos caminhos entre a fonte e destino foi balanceada. Dessa forma se evitou a formação de buracos pela rede, elevando número de mensagens recebidas.

5.2 Sugestões para Trabalhos Futuros

Como sugestão para trabalhos futuros, recomenda-se um estudo mais profundo dos efeitos do balanceamento do consumo da energia nos nós da rede em outras topologias. Pode-se variar a disposição dos nós sensores, a quantidade de nós tanto os nós fontes quanto os nós sorvedouros.

Um outro ponto a ser explorado é o critério para a escolha das rotas. Poderia ser utilizada a distância euclidiana entre os nós fontes e destino para realizar o balanceamento da energia. A posição geográfica dos nós pode ser obtida através de GPS ou pela utilização de algoritmos de localização. O temporizador dos nós também poderia se beneficiar do conhecimento da posição dos nós para ser implementado, aumentando o número de caminhos possíveis de roteamento. Outra implementação possível do temporizador é ser aleatório de forma a permitir uma maior quantidade de escolhas dos caminhos.

Além disso, o filtro Energia foi instanciado em todos os nós da rede, tanto nos próximos às fontes, como nos próximos ao sorvedouro. Pode-se explorar uma topologia em que alguns nós possuam o filtro e outros não. Dessa forma, possivelmente o tempo de descoberta de rotas seria diminuído, com possível redução no consumo de energia.

O mecanismo proposto foi implementado sobre a difusão direcionada, herdando as suas vantagens e limitações. Essa abordagem pode ser estendida e implementada em outros protocolos centrados em dados.

Referências Bibliográficas

- [1] HSU, V., KAHN, J. M., E PISTER, K. S. J. Wireless communications for smart dust. *Electronics Research Laboratory Technical Memorandum Number M98/2* (fevereiro de 1998).
- [2] TAYLOR, C. What dust can tell you. Página da World Wide Web. http://www.dust-inc.com/news/articles/Jan12_2004.htm, abril de 2005.
- [3] Wireless sensor networks are coming to a machine near you. Página da World Wide Web. <http://www.machinedesign.com>, abril de 2005.
- [4] KHAN, K. Industry focus: Resources the sensor revolution. Página da World Wide Web. <http://www.accenture.com>, abril de 2005.
- [5] Crossbow technology. Página da World Wide Web. <http://www.xbow.com>, abril de 2005.
- [6] LAHIRI, K., RAGHUNATHAN, A., DEY, S., E PANIGRAHI, D. Battery-driven system design: A new frontier in low power design. *Proceedings of International Conference on VLSI Design, Bangalore, India* (janeiro de 2002), 261–267.
- [7] PEDRAM, M. Power optimization and management in embedded systems. *Proceedings of ASP-DAC 2001, Yokohama, Japan* (fevereiro de 2001), 239–244.
- [8] GONÇALVES PINTO, A. J. Mecanismo de agregação de dados baseado em técnicas paramétricas aplicado em redes de sensores. Tese de mestrado, COPPE/UFRJ, Abril de 2004.

- [9] POTTIE, G. J., E KAISER, W. J. Wireless Integrated Network Sensors - WINS. *Communications of the ACM* (maio de 2000), 51–58.
- [10] CHANG, J.-H., E TASSIULAS, L. Energy conserving routing in wireless ad-hoc networks. *Proceedings of IEEE INFOCOM Tel Aviv, Israel* (março de 2000).
- [11] WU, J., DAI, F., GAO, M., E STOJMENOVIC, I. On calculating power-aware connected dominating set for efficient routing in ad hoc wireless networks. *IEEE/KICS Journal of Communications and Networks* (março de 2002).
- [12] CHEN, B., JAMIESON, K., BALAKRISHNAN, H., E MORRIS, R. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *In 7th Annual International Conference on Mobile Computing and Networking* (julho de 2001).
- [13] SHAH, R. C., E RABAEY, J. Energy aware routing for low energy ad hoc sensor networks. *IEEE Wireless Communications and Networking Conference (WCNC)* (2002).
- [14] K PAHLAVAN, P. K., E HATAMI, A. A handoff in hybrid mobile data networks. *IEEE Personal Communications* (abril de 2000), 34–36.
- [15] Ietf mobile ad hoc networks working group. Página da World Wide Web. <http://www.ietf.org/html.charters/manet-charter.html>, abril de 2005.
- [16] Wireless integrated micro sensors project. University of California at Los Angeles UCLA. Electrical Engineering Department. Página da World Wide Web. <http://www.janet.ucla.edu/WINS/>, abril de 2005.
- [17] SOHRABI, K., GAO, J., AILAWADHI, V., E POTTIE, G. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications* (outubro de 2000), 16–27.
- [18] AKYILDIZ, I. F., SU, W., SANKARASUBRAMANIAM, Y., E CAYIRICI, E. A survey on sensor networks. *Communications of the ACM* (agosto de 2002), 102–114.

- [19] The factoid project at compaq research. Página da World Wide Web, <http://www.research.compaq.com/wrl/techreports/abstracts/TN-60.html>, abril de 2005.
- [20] The smart dust project. Página da World Wide Web. <http://robotics.eecs.berkeley.edu/pister/SmartDust/>, abril de 2005.
- [21] Ultra low power wireless sensor project. Página da World Wide Web. http://www-mtl.mit.edu/jimg/project_top.html, abril de 2005.
- [22] Mit uamps project. Página da World Wide Web. <http://www.mtl.mit.edu/research/icsystems/uamps/>, abril de 2005.
- [23] RABAEY, J., AMMER, J., DA SILVA JR., J. L., E PATEL, D. Picoradio: Ad-hoc wireless networking of ubiquitous low-energy sensor/monitor nodes. *WVLSI* (abril de 2000).
- [24] RABAEY, J. M., AMMER, J., KARALAR, T., LI, S., OTIS, B., SHEETS, M., E TUAN, T. Picoradios for wireless sensor networks - the next challenge in ultra-low power design. *Proceedings of the International Solid-State Circuits Conference* (fevereiro de 2002).
- [25] STOCHERO, J. M. Roteamento favorecendo agregação de dados em redes de sensores. Tese de mestrado, COPPE/UFRJ, Junho de 2004.
- [26] Pesquisa em redes de sensores. Página da World Wide Web. <http://www.gta.ufrj.br/wsns>, abril de 2005.
- [27] LORCH, J., E SMITH, A. Software strategies for portable computer energy management. *IEEE Personal Communications*, Vol.5, No.3 (junho de 1998), 60–73.
- [28] BENINI, L., BOGLIOLO, A., E DE MICHELI, G. A survey of design techniques for system-level dynamic power management. *Trans. on VLSI Systems*, Vol.8, No.3 (junho de 2000), 299–316.
- [29] PARK, S., SAVVIDES, A., E SRIVASTAVA, M. B. Simulating networks of wireless sensors. *WSC '01: Proceedings of the 33rd conference on Winter simulation* (2001), 1330–1338.

- [30] HOLLAR, S. Cots dust. Tese de mestrado, University of California/Berkeley, 1996.
- [31] Mica2 - wireless measurement system. Página da World Wide Web. http://www.xbow.com/products/product_pdf_files/wireless_pdf/mica2.pdf, abril de 2005.
- [32] Zigbee alliance. Página da World Wide Web. <http://www.zigbee.org>, abril de 2005.
- [33] LEOPOLD, M., DYDENSBORG, M. B., E BONNET, P. Bluetooth and sensor networks: a reality check. *SenSys'03: Proceedings of the 1st International conference on Embedded networked sensor system* (2003), 103–113.
- [34] YE, W., HEIDEMANN, J., E ESTRIN, D. Medium access control with coordinated, adaptive sleeping for wireless sensor networks. *USC/ISI TECHNICAL REPORT ISI-TR-567* (janeiro de 2003).
- [35] YE, W., HEIDEMANN, J., E ESTRIN, D. An energy-efficient mac protocol for wireless sensor networks. *Proceedings of IEEE INFOCOM* (junho de 2002).
- [36] HEIDEMANN, J., SILVA, F., INTANAGONWIWAT, C., GOVINDAN, R., ESTRIN, D., E GANESAN, D. Building efficient wireless sensor networks with low-level naming. *Proceedings of the Symposium on Operating Systems Principles* (2001), 146–159.
- [37] ADJIE-WINOTO, W., SCHWARTZ, E., BVALAKRISHNAN, H., E LILLEY, J. The design and implementation of an intentional naming system. *ACM Symposium on Operating Systems Principles* (dezembro de 1999).
- [38] HEINZELMAN, W. R., CHANDRAKASAN, A., E BALAKRISHNAN, H. Energy-efficient communication protocol for wireless microsensor networks. *Proceedings of the 33rd IEEE Hawaii International Conference on System Sciences (HICSS)* (janeiro de 2000), 1–10.
- [39] LINDSEY, S., E RAGHAVENDRA, C. S. Pegasus: Power efficient gathering in sensor information systems. *International Conference on Communications* (2001).

- [40] INTANAGONWIWAT, C., GOVINDAN, R., E ESTRIN, D. Directed diffusion: A scalable and robust communication paradigm for sensor networks. *Proceedings of ACM MobiCom'00, Boston, MA* (agosto de 2000), 56–67.
- [41] SANKARASUBRAMANIAM, Y., AKAN, O., E AKYILDIZ, I. Esrt : Event-to-sink reliable transport in wireless sensor networks. *The fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing* (junho de 2003).
- [42] PINTO, A. J. G., STOCHERO, J. M., E DE REZENDE, J. F. Descoberta de rotas e mecanismo de agregação em redes de sensores. *22º Simpósio Brasileiro de Redes de Computadores (SBRC 2004), Gramado, RS, Brasil* (março de 2004).
- [43] HEIDEMANN, J., SILVA, F., YU, Y., ESTRIN, D., E HALDAR, P. *Diffusion Filters as a Flexible Architecture for Event Notification in Wireless Sensor Networks*. USC/ISI Technical Report 2002-556, 2002.
- [44] The network simulator ns-2. Página da World Wide Web. <http://www.isi.edu/nsnam/ns>, abril de 2005.
- [45] Otcl - object tcl: the faq manual. Página da World Wide Web. <ftp://ftp.tns.lcs.mit.edu/pub/otcl/README.html>, abril de 2005.
- [46] Scalable coordination architectures for deeply distributed systems project. University of South California. Information Sciences Institute. Página da World Wide Web. <http://www.isi.edu/scadds/>, abril de 2005.
- [47] DEBORAH ESTRIN, LEROY GIROD, G. P., E SRIVASTAVA, M. Instrumenting the world with wireless sensor networks. *International Conference on Acoustics, Speech and Signal Processing, (ICASSP 01)* (maio de 2001).
- [48] TEIXEIRA, I., DE REZENDE, J. F., E PEDROZA, A. C. P. Wireless sensor networks: Improving the network energy consumption. *Simpósio Brasileiro de Telecomunicações (SBTr)* (setembro de 2004).

Apêndice A

O Filtro Energia

O FILTRO desenvolvido neste trabalho, implementado na linguagem C++ foi incorporado ao código do simulador NS-2. O filtro, que utiliza a API desenvolvida pelo grupo SCADDS [43], tem seu código apresentado na listagem que se segue, relativas aos arquivos energy.cc e energy.hh.

```
// *****  
// Programa de Pós-Graduação em Engenharia Elétrica - COPPE  
// Mestrado em Engenharia Eletrica  
// Ênfase      : Redes de Computadores  
// Aluna       : Ingrid Teixeira  
// Orientadores : Prof Aloysio de Castro P. Pedroza e José Ferreira  
//             : de Rezende  
// Arquivo     : Energy.cc  
// Objetivo    : Tratar as mensagens do tipo Exploratórias recebidas  
//             : pelos nós de modo a estabelecer árvores de roteamento  
//             : capazes de distribuir o consumo de energia pelos nós da  
//             : rede  
//*****  
  
#include "energy.hh"  
#include "attrs.hh"  
#include <algorithm>  
#include <list>  
  
static class EnergyFilterClass : public TclClass  
{  
public:  
    EnergyFilterClass() : TclClass( "Application/DiffApp/EnergyFilter" ) { }  
    TclObject* create ( int argc , const char*const* argv )
```

```

{
    if ( argc == 5 )
    {
        int prio = atoi ( argv[4] ) ;
        return ( new EnergyFilter ( prio ) ) ;
    }
}
} class_Energy_filter ;

int EnergyFilter::command ( int argc , const char*const* argv )
{
    if ( argc == 3 )
    {
        if ( strcasecmp ( argv[1] , "debug" ) == 0 )
        {
            global_debug_level = atoi( argv[2] ) ;
            if ( global_debug_level < 1 || global_debug_level > 10 )
            {
                global_debug_level = DEBUG_DEFAULT ;
                printf ( "Error: Debug level outside range(1-10) or missing !\n" ) ;
            }
        }
    }
    if ( argc == 2 )
    {
        if ( strcmp ( argv[1] , "start" ) == 0 )
        {
            run () ;
            return TCL_OK ;
        }
    }
    return DiffApp::command ( argc , argv ) ;
}

MessageEntry * EnergyFilter::findMessageEntrySender (Message *msg)
{
    MessageList::iterator message_entry_itr;
    MessageEntry *message_entry;

    for (message_entry_itr = message_list_.begin(); message_entry_itr !=
        message_list_.end(); ++message_entry_itr)
    {
        message_entry = *message_entry_itr;
        if ((message_entry->pkt_num_==msg->pkt_num_)&&(message_entry->
            rdm_id_==msg->rdm_id_)&&(message_entry->node_id_==msg->last_hop_))
        {
            return message_entry;
        }
    }
}

```

```

    }
    return NULL;
}

MessageEntry * EnergyFilter::findMessageEntry (Message *msg)
{
    MessageList::iterator message_entry_itr;
    MessageEntry *message_entry;

    for (message_entry_itr = message_list_.begin(); message_entry_itr
         != message_list_.end(); ++message_entry_itr)
    {
        message_entry = *message_entry_itr;
        if ((message_entry->pkt_num_==msg->pkt_num_)&&(message_entry->rdm_id_
            ==msg->rdm_id_)&&(message_entry->node_id_==msg->next_hop_))
        {
            return message_entry;
        }
    }
    return NULL;
}

void EnergyFilter::EnergyTimeout(Message *msg)
{
    MessageList::iterator message_itr;
    MessageEntry *message_entry;

    DiffPrint ( DEBUG_IMPORTANT , "Energy: Node %d Energy Timeout.\n" ,
               msg->next_hop_ ) ;

    message_itr = message_list_.begin();

    while ( message_itr != message_list_.end() ) {
        message_entry = *message_itr;
        if ((message_entry->pkt_num_==msg->pkt_num_)&&(message_entry->rdm_id_
            ==msg->rdm_id_)&&(message_entry->node_id_==msg->next_hop_))
            break;
        else message_itr ++ ;
    }
    // Recupera a message_entry do no e coloca na msg a ser enviada
    msg->next_hop_ = message_entry->next_hop_;
    msg->last_hop_ = message_entry->last_hop_;
    msg->src_id_ = message_entry->src_id_;
    msg->energy_ = message_entry->energy_;
    msg->dist_src_ = message_entry->dist_src_;
    msg->new_message_ = message_entry->new_message_ ;

    ( ( DiffusionRouting * ) dr_ ) -> sendMessage (msg , filter_handle_);
}

```

```

//message_itr = message_list_.erase(message_itr);
//delete message_entry;
}

void EnergyFilter::ProcessMessage ( Message *msg )
{
    TimerCallback *energy_timer ;
    MessageEntry *message_entry ;
    Node *thisnode ;
    double node_energy ;

    switch ( msg->msg_type_ )
    {
        case DATA:

            DiffPrint( DEBUG_NO_DETAILS , "Energy: Received Data !\n" ) ;
            ( ( DiffusionRouting * ) dr_ )->sendMessage( msg , filter_handle_ ) ;
            break ;

        case PUSH_EXPLORATORY_DATA:

            DiffPrint ( DEBUG_NO_DETAILS , "Energy: Received a Push
                Exploratory Data !\n" ) ;
            ( ( DiffusionRouting * ) dr_ ) -> sendMessage ( msg , filter_handle_ ) ;
            break ;

        case EXPLORATORY_DATA:
            if (msg->last_hop_ == LOCALHOST_ADDR){
                ((DiffusionRouting *)dr_)->sendMessage(msg, filter_handle_);
            }
            else
            {
                // Recupera o ID do no que recebeu a mensagem
                thisnode = Node::get_node_by_address( ((DiffusionRouting *)dr_)->getNodeId() );
                //thisnode = Node::get_node_by_address( msg->next_hop_ );

                // Recupera a energia residual deste no
                node_energy = thisnode->energy_model()->energy();

                DiffPrint ( DEBUG_NO_DETAILS, "Energy: No %d recebendo ED de no %d\n"
                    , msg->next_hop_, msg->last_hop_);

                // Incrementa os campos energia do caminho e distancia
                msg->energy_ += node_energy;
                msg->dist_src_ ++;

                // Procura pelo no que enviou a mensagem na nossa lista de message_entry

```

```

// Teoricamente somente o no fonte que envia a primeira ED atende esta condicao
message_entry = findMessageEntrySender (msg);
if (!message_entry){
    // Inere uma nova message_entry na lista
    DiffPrint (DEBUG_IMPORTANT,"Energy: PRIMEIRA Inserindo nova ED na
                message_list do no %d\n",msg->last_hop_);
    message_entry = new MessageEntry (msg->last_hop_,
                                      msg->rdm_id_ , msg->pkt_num_ , msg->last_hop_ ,
                                      0 , msg->dist_src_ , msg->energy_ , msg->aggreg_ID_,
                                      msg->src_id_ , msg->next_hop_);
    message_entry->node_id_ = msg->last_hop_;
message_entry->src_id_ = msg->src_id_;
    message_entry->energy_ = msg->energy_;
    message_entry->dist_src_ = msg->dist_src_;
    message_entry->attrs_ = CopyAttrs ( msg->msg_attr_vec_ );
    message_entry->new_message_ = msg->new_message_ ;

    // Inere novo elemento
    message_list_.push_back(message_entry);
}

// Procura pelo no que recebeu a mensagem na nossa lista de message_entry
message_entry = findMessageEntry (msg);
if (!message_entry){
    // Inere uma nova message_entry na lista
    DiffPrint ( DEBUG_IMPORTANT, "Energy: Inserindo nova ED na message_list
                do no %d\n" , msg->next_hop_);
    message_entry = new MessageEntry (msg->next_hop_,
                                      msg->rdm_id_ , msg->pkt_num_ , msg->last_hop_ ,
                                      0 , msg->dist_src_ , msg->energy_ , msg->aggreg_ID_,
                                      msg->src_id_ , msg->next_hop_);
    message_entry->node_id_ = msg->next_hop_;
message_entry->src_id_ = msg->src_id_;
    message_entry->energy_ = msg->energy_;
    message_entry->dist_src_ = msg->dist_src_;
    message_entry->attrs_ = CopyAttrs ( msg->msg_attr_vec_ );
    message_entry->new_message_ = msg->new_message_ ;

    energy_timer = new EnergyExpirationTimer ( this , CopyMessage ( msg ) ) ;
    DiffPrint ( DEBUG_IMPORTANT, "Energy: timer %d\n" , (timer_*msg->dist_src_ ) );
    //message_entry->energy_timer_handle_ = ( (DiffusionRouting *) dr_ ) ->
        addTimer((int)(timer_*msg->dist_src_), energy_timer ) ;

    message_entry->energy_timer_handle_ = ( (DiffusionRouting *) dr_ ) -> addTimer((int)
        timer_ + (int) ((ENERGY_EXPIRATION_JITTER * (GetRand() * 1.0 / RAND_MAX)
        - ( ENERGY_EXPIRATION_JITTER / 2 ) ) ) , energy_timer ) ;

    // Inere novo elemento

```



```

        message_list_.push_back(message_entry);
    }
    else
        // Verifica se a mensagem existente eh melhor em termos de energia
        // Atualiza a message_entry deste no
        if ((float)(msg->energy_/msg->dist_src_) >= (float)(message_entry->energy_
            /message_entry->dist_src_)) {
            DiffPrint ( DEBUG_IMPORTANT, "Energy: mensagem recebida eh melhor que a
                message_list do no %d\n" , msg->next_hop_);
            message_entry->node_id_ = msg->next_hop_;
            message_entry->rdm_id_ = msg->rdm_id_;
            message_entry->pkt_num_ = msg->pkt_num_;
            message_entry->aggreg_ = msg->aggreg_;
            message_entry->aggreg_ID_ = msg->aggreg_ID_;
            message_entry->last_hop_ = msg->last_hop_;
            message_entry->src_id_ = msg->src_id_;
            message_entry->next_hop_ = msg->next_hop_;
            message_entry->energy_ = msg->energy_;
            message_entry->dist_src_ = msg->dist_src_;
            message_entry->attrs_ = CopyAttrs ( msg->msg_attr_vec_ );
            message_entry->new_message_ = 1;
        }
    }
}
}
}

void EnergyFilter::recv ( Message *msg , handle h )
{
    struct timeval tmv;

    GetTime(&tmv);

    pkt_count_++;

    DiffPrint ( DEBUG_NO_DETAILS , "Entering the Energy Filter %d. a %d\n" , pkt_count_ , a ) ;

    ProcessMessage ( msg ) ;
}

int EnergyExpirationTimer::expire()
{
    // Call timeout function
    agent_->EnergyTimeout(msg_);

    // Do not reschedule this timer
    return -1;
}

```

```

handle EnergyFilter::setupFilter()
{
    NRAttrVec attrs ;
    handle h ;

    // For the Energy filter, we are interested only in DATA_CLASS

    attrs.push_back ( NRClassAttr.make ( NRAttribute::EQ , NRAttribute::DATA_CLASS ) ) ;

    h = ( ( DiffusionRouting * ) dr_ ) -> addFilter ( &attrs , filter_priority_ ,
        filter_callback_ ) ;

    ClearAttrs ( &attrs ) ;
    return h ;
}

void EnergyFilter::run()
{
    // Set up the filter
    filter_handle_ = setupFilter() ;

    DiffPrint ( DEBUG_ALWAYS , "Energy filter subscribed to *, received handle %d\n" ,
        filter_handle_ ) ;
    DiffPrint ( DEBUG_ALWAYS , "Energy filter initialized !\n" ) ;
}

EnergyFilter::EnergyFilter ( int prio ) : pkt_count_( 0 )
{
    filter_priority_ = prio;
    a=200;
    // Create callback classes and set up pointers
    filter_callback_ = new EnergyFilterReceive ( this ) ;
    bind ( "timer_" , &timer_ ) ;
    bind ( "energy_delta_" , &energy_delta_ ) ;
    bind ( "nrsrc_" , &nrsrc_ ) ;
}

void EnergyFilterReceive::recv ( Message *msg , handle h )
{
    app_->recv ( msg , h ) ;
}

```

```

// *****
// Programa de Pós-Graduação em Engenharia Elétrica - COPPE
// Mestrado em Engenharia Eletrica
// Ênfase      : Redes de Computadores
// Aluna       : Ingrid Teixeira
// Orientadores : Prof Aloysio de Castro P. Pedroza e José Ferreira de Rezende
// Arquivo     : Energy.hh
// Objetivo    : Declaração das classes, atributos e métodos utilizados
//              pelo filtro Energia
//*****

#ifndef _ENERGY_HH_
#define _ENERGY_HH_

#include "diffapp.hh"

#define ENERGY_FILTER_PRIORITY 10
#define ENERGY_EXPIRATION_DELAY 300 // (msec) bw receive and forward
#define ENERGY_EXPIRATION_JITTER 100 // (msec) jitter

extern NRSimpleAttributeFactory<char *> SrcAttr;
#define Src_KEY 3640

extern NRSimpleAttributeFactory<char *> E1Attr;
extern NRSimpleAttributeFactory<char *> E2Attr;
#define E1_KEY 3650
#define E2_KEY 3660

class MessageEntry {
public:
    MessageEntry(int32_t node_id, int32_t rdm_id, int32_t pkt_num, int32_t last_hop ,
                int aggreg, u_int16_t dist_src, double energy, int64_t aggreg_ID,
                double src_id, int32_t next_hop) :
        node_id_(node_id), rdm_id_(rdm_id), pkt_num_(pkt_num), last_hop_(last_hop), aggreg_
        (aggreg), dist_src_(dist_src), energy_(energy), aggreg_ID_(aggreg_ID), src_id_
        (src_id), next_hop_(next_hop) {};
    int32_t node_id_;
    int32_t rdm_id_;
    int32_t pkt_num_;
    int32_t last_hop_;
    int aggreg_;
    u_int16_t dist_src_;

    NRAttrVec *attrs_;
    handle energy_timer_handle_;
    int new_message_;
    double energy_;
    int64_t aggreg_ID_;

```

```
double src_id_;
int32_t next_hop_;
};

class EnergyFilter;

typedef list<MessageEntry *> MessageList;
// List of all messages stored in a node
MessageList message_list_;
int a;

class EnergyFilterReceive;

class EnergyFilter : public DiffApp {
public:
    EnergyFilter(int prio);

    int command(int argc, const char*const* argv);

    void recv(Message *msg, handle h);

    void run();

    // Timers
    void EnergyTimeout(Message *msg);

protected:

    // int a;

    int pkt_count_;
    int filter_priority_;
    int timer_;
    double energy_delta_ ;
    int nrsrc_;
    //handles
    handle filter_handle_;

    // Receive Callback for the filter
    EnergyFilterReceive *filter_callback_;

    // List of all messages stored in a node
    //MessageList message_list_;

    // Message Processing functions
    void ProcessMessage(Message *msg);

    // Message Entry related functions
```

```
MessageEntry * findMessageEntry (Message *msg);
MessageEntry * findMessageEntrySender (Message *msg);
void insertMessageEntry (int32_t node_id, MessageEntry *message_entry);
void deleteMessageEntry (int32_t node_id);

handle setupFilter();

};

class EnergyExpirationTimer : public TimerCallback {
public:
    EnergyExpirationTimer(EnergyFilter *agent, Message *msg) :
        agent_(agent), msg_(msg){};
    ~EnergyExpirationTimer()
    {
        delete msg_;
    };
    int expire();

    EnergyFilter *agent_;
    Message *msg_;
};

class EnergyFilterReceive : public FilterCallback {
public:
    EnergyFilterReceive(EnergyFilter *app) : app_(app) {};
    void recv(Message *msg, handle h);

    EnergyFilter *app_;
};

#endif // !_ENERGY_HH_
```