

DANIEL LUDOVICO GUIDONI

**DIFUSÃO DE DADOS BASEADA EM ATRASO E
ENERGIA PARA REDES DE SENSORES SEM FIO**

Belo Horizonte
28 de fevereiro de 2007

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

**DIFUSÃO DE DADOS BASEADA EM ATRASO E
ENERGIA PARA REDES DE SENSORES SEM FIO**

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

DANIEL LUDOVICO GUIDONI

Belo Horizonte
28 de fevereiro de 2007



UNIVERSIDADE FEDERAL DE MINAS GERAIS

FOLHA DE APROVAÇÃO

Difusão de Dados Baseada em Atraso e
Energia para Redes de Sensores Sem Fio

DANIEL LUDOVICO GUIDONI

Dissertação defendida e aprovada pela banca examinadora constituída por:

Dr. ANTONIO ALFREDO FERREIRA LOUREIRO – Orientador
Universidade Federal de Minas Gerais - UFMG

Dra. RAQUEL APARECIDA DE FREITAS MINI – Co-orientadora
Pontifícia Universidade Católica de Minas gerais - PUC Minas

Dr. EDUARDO FREIRE NAKAMURA.
Fundação Centro de Análise Pesquisa e Inovação Tecnológica - FUCAPI, AM

Dra. JUDITH KELNER
Universidade Federal de Pernambuco - UFPE

Belo Horizonte, 28 de fevereiro de 2007

*Aos DEBBpais
Severino e Marlene
Se aos DEBBamigos
Renato e Otávio.*

Agradecimentos

Primeiramente, eu agradeço aos meu pais Severino e Marlene, não apenas pela ajuda financeira pois, sei que foi feita de coração, mas pelo incentivo em todas as horas. Horas essas em que eu tive que ficar mais de três meses sem vê-los. Mas sei que mesmo de longe, muito longe, eles estiveram sempre próximos, ou num telefonema, ou em uma prece antes de dormir. Obrigado e espero não tê-los decepcionado. Não posso esquecer é claro, dos ensinamentos do meu irmão Gustavo, que no começo da graduação teve um papel importante para que eu aprendesse os conceitos básicos de computação e pelo tempo em que utilizava o seu computador para fazer os trabalhos práticos. Agradeço também a minha irmã querida, que mesmo de longe, sempre me apoiou e me deu forças para realizar esse sonho.

Agradeço em especial a um amigo, que me introduziu na pesquisa, fornecendo os ensinamentos básico para essa arte. Me mostrou o caminho para que eu chegasse hoje onde eu estou. Obrigado Fabrício! Seremos sempre amigos.

Agradeço em especial também, a um amigo que fiz quando tinha 14 anos no CEFET-ES. Desde então, passei a ter mais um irmão, chamado Renato. Agradeço pelo seu apoio e atenção, me ligando nos momentos que sabia que eu não estava muito bem para conversarmos e dar conselhos, esses que eu sempre dei muito valor. Não posso esquecer dos outros amigos, mas não menos importantes: Geléia, Buraco, Zacarias, Sungão, Maumita, Mingau, Ximits entre outros que estarão sempre em meu coração.

Agradeço aos meus Mestres, Antonio A. F. Loureiro e Raquel. A. F. Mini. À Raquel pela paciência e dedicação na iniciação científica que fizemos na graduação. Ao Loureiro por me dar o prazer de trabalhar ao seu lado. Agradeço aos dois pelos importantes ensinamentos na pesquisa ou pessoais/profissionais que recebi nesse período e espero receber nos próximos 4 anos.

Ao aluno de Doutorado Max do Val Machado, que teve a paciência e confiança em me ajudar nesse período, sendo para me ensinar o NS ou me ensinando a fazer pesquisa. Obrigado.

Agradeço aos amigos de república e aos colegas do laboratório. Aos amigos que fiz nas disciplinas, em especial o Anísio, Lara, Mário Sérgio e aos demais amigos que fiz

nesse período.

Resumo

Um dos recursos mais importantes em redes de sensores é a energia, porque, em geral, as baterias dos nós não podem ser recarregadas. Por isso, quaisquer soluções para essas redes devem ser eficientes em termos de energia. O objetivo deste trabalho é projetar protocolos para disseminar uma informação para todos os nós da rede, ou seja, soluções para realizar difusão de dados em redes de sensores sem fio. Com esse objetivo, é proposto o “*Delay and Energy Based Broadcasting*” (DEBB), um protocolo que utiliza o mapa de energia e o conceito de propagação com atrasos para realizar a difusão de dados eficiente em termos de energia em redes de sensores sem fio. O funcionamento básico do DEBB é simples e otimizado para reduzir a carga de trabalho dos nós sensores. Resultados de simulação mostram a eficiência do protocolo em relação à cobertura da rede, ao número de transmissões e ao consumo de energia. O protocolo apresenta a habilidade de contornar regiões de baixa energia e seu comportamento é escalável quando o número de regiões de baixa energia aumenta. Além disso, o DEBB apresenta um mecanismo para a redução da latência, principal problema dos algoritmos baseados em propagação com atraso.

Abstract

One of the most important resources in wireless sensor networks is energy, since, in general, the nodes batteries cannot be recharged. Then, all solutions to these networks must be efficient in terms of energy. The goal of this work is to design protocols to disseminate a data to all network nodes, i.e., solutions to perform broadcasting in wireless sensor networks. With this goal, it is proposed the Delay and Energy Based Broadcasting (DEBB), a protocol that combines the concepts of forwarding delay with the energy map to perform the energy-efficient broadcasting in wireless sensor networks. The basic operating of DEBB is simple and optimized to reduce the workload in sensor nodes. Simulation results revealed the efficiency of the proposed protocol in relation to the network coverage, number of transmissions and the energy consumption. The protocol presents the ability to avoid low energy regions and it is scalable when the number of low energy regions increases. Besides, DEBB presents a mechanism to reduce the latency that is the main problem of the forwarding delay algorithms.

Sumário

1	Introdução	1
1.1	Motivação	1
1.2	Definição do Problema	2
1.3	Objetivos	3
1.4	Organização do Trabalho	5
2	Trabalhos Relacionados	6
2.1	Roteamento em Redes de Sensores Sem Fio	6
2.2	Difusão em Redes de Sensores Sem Fio	8
2.2.1	<i>Trajectory and Energy Based Data Dissemination</i> (TEDD)	8
2.2.2	<i>Dynamic Delayed Broadcasting</i> (DDB)	10
2.2.3	<i>Broadcast Protocol for Sensor Networks</i> (BPS)	13
2.2.4	<i>Flooding e Gossiping</i>	14
2.3	Mapa de Energia	15
2.4	<i>State-based Energy Dissipation Model</i> (SEDM)	16
2.5	Conclusão	18
3	Delay and Energy Based Broadcasting (DEBB)	20
3.1	Introdução	20
3.2	Energia de Corte	21
3.3	Funcionamento Básico do DEBB	22
3.4	DEBB Probabilístico	27
3.5	Conclusão	30
4	Resultados de Simulação	32
4.1	Introdução	32
4.2	Avaliação da Probabilidade de Transmissão do DEBB-P	34
4.3	Difusão de Dados em Rede com Energia Homogênea	37
4.4	Difusão de Dados Evitando Uma Região de Baixa Energia	41
4.5	Difusão de Dados Evitando Quatro Regiões de Baixa Energia	44

4.6 Conclusão	47
5 Conclusões e Direções Futuras	49
Referências Bibliográficas	51

Lista de Figuras

1.1	Formas de comunicação de dados em redes de sensores sem fio.	4
2.1	Pontos de referência.	10
2.2	Funcionamento básico do TEDD.	10
2.3	Exemplo da área adicional estimada pelo nó A quando o DDB1 é utilizado.	11
2.4	Funcionamento básico do DDB.	12
2.5	Funcionamento básico do BPS.	14
2.6	Mapa de energia de uma rede de sensores sem fio.	15
2.7	Diagrama do <i>State-based Energy Dissipation Model</i>	18
3.1	Escolha da energia de corte.	22
3.2	Funcionamento básico do DEBB.	23
3.3	Funcionamento básico do DEBB-P.	28
4.1	Cenários de simulação.	33
4.2	Latência nas diferentes probabilidades do DEBB-P.	37
4.3	Resultados na rede com energia homogênea.	38
4.4	Latência dos protocolos.	39
4.5	Resultados dentro da região de baixa energia.	42
4.6	Resultados fora da região de baixa energia.	43
4.7	<i>Snapshot</i> dos nós cobertos pelos diferentes protocolos utilizando o cenário da figura 4.1 (b). Os quadrados brancos representam nós que receberam o pacote de difusão. Os quadados pretos representam os nós que não receberam o pacote de difusão.	44
4.8	Resultados dentro das regiões de baixa energia.	46
4.9	<i>Snapshot</i> dos nós cobertos pelos diferentes protocolos utilizando o cenário da figura 4.1 (c). Os quadrados brancos representam nós que receberam o pacote de difusão. Os quadados pretos representam os nós que não receberam o pacote de difusão.	47
4.10	Resultados fora das regiões de baixa energia.	48

Lista de Tabelas

4.1	Parâmetros utilizados nas simulações.	35
4.2	Média e intervalo de confiança (valores absolutos) da transmissão, cobertura, energia e nós mortos, variando a probabilidade de transmissão máxima do DEBB-P. Tx e Rx são a quantidade de nós que receberam pacotes transmitidos e recebidos respectivamente em uma difusão. En e Nm são energia média dos nós e a quantidade de nós mortos no final da simulação e IC é o intervalo de confiança.	36
4.3	Média do número de operações, transmissões e cobertura da rede em cada difusão de dados.	40

Lista de Algoritmos

1	Algoritmo do DEBB.	25
2	Algoritmo do DEBB-P.	29

Capítulo 1

Introdução

1.1 Motivação

Redes de Sensores Sem Fio (RSSFs) são redes compostas por um grande número de dispositivos com capacidade de realizar processamento, comunicação e sensoriamento distribuído (Estrin et al., 1999; Intanagonwiwat et al., 2002, 2000; Pottie e Kaiser, 2000a,b). Nesse cenário, sensores inteligentes coletam dados do ambiente em que foram inseridos e, em seguida, processam e propagam esses dados para um nó *sink*. O *sink* apresenta uma capacidade computacional superior a dos demais nós e sua responsabilidade é processar as informações de sensoriamento fornecidas pela rede, disponibilizá-las para observadores externos e disseminar informações de controle para a rede. As RSSFs podem ser aplicadas no monitoramento, rastreamento, coordenação e processamento de diversas aplicações. Por exemplo, sensores podem ser interconectados para monitorar e controlar condições ambientais como em florestas, oceanos e planetas. Outras aplicações possíveis dessa tecnologia estão nas áreas médica, militar, espacial, industrial, urbana e rural. A interconexão desses sensores através de redes sem fio, com o objetivo de realizar qualquer tarefa de sensoriamento, possivelmente, revolucionará os processos de obtenção e processamento de informações.

As redes de sensores sem fio podem ser projetadas para coletar dados em diferentes ambientes e mesmo em regiões de difícil acesso ou perigosas. Na área ambiental, as redes de sensores serão importantes na prevenção e no auxílio às vítimas de catástrofes naturais tais como terremotos, vulcões, *tsunamis* e furações. As RSSFs podem monitorar variáveis ambientais em locais internos como prédios e residências. Aplicações rurais podem ser desenvolvidas para o controle de plantações e de criações de animais. Na área urbana, as redes de sensores podem ajudar a melhorar as condições de tráfego e de segurança. É possível monitorar o tráfego de veículos em rodovias, malhas viárias urbanas e prover segurança em centros comerciais. Na área médica, as RSSFs podem

ser utilizadas para monitorar o funcionamento de órgãos como o coração, detectar a presença de substâncias que indicam a existência ou o surgimento de um problema biológico, seja no corpo humano ou animal. Na área militar, as redes de sensores sem fio são capazes de detectar movimentos inimigos, explosões, a presença de material perigoso como gás venenoso ou radiação. Nesse tipo de aplicação, os requisitos de segurança são fundamentais. Além disso, o alcance das transmissões dos sensores é geralmente reduzido para economizar energia e evitar escutas clandestinas. Na área espacial, as RSSFs podem auxiliar em tarefas relacionadas à exploração inicial de um planeta ou estrela. Nesse caso, a rede pode ter como objetivo avaliar os perigos existentes para uma eventual visita humana. Na área industrial, as RSSFs podem prover mecanismos de controle industrial. Por exemplo, micro sensores sem fio podem ser embutidos em “peças” na linha de montagem com objetivo de realizar testes no processo de manufatura. A produção industrial pode ser otimizada a partir do monitoramento em indústrias petroquímicas, fábricas, refinarias e siderurgias.

Os requisitos de projeto de um nó sensor levam a várias restrições. Os dispositivos devem apresentar tamanhos reduzidos e um custo de produção mínimo. A energia é o recurso mais escasso já que o nó sensor utiliza como fonte de energia uma bateria cuja recarga nem sempre é possível. Em muitas situações, o acesso aos nós sensores pode ser difícil, quando não inviável. Mesmo se a recarga de bateria for realizada, observar-se-á um problema de escalabilidade devido ao grande número de nós sensores em uma RSSF.

Além da restrição de energia, o projeto de soluções para as redes de sensores sem fio deve ser robusto na presença de uma topologia dinâmica. Nós sensores podem deixar de operar por falta de energia ou por problemas mecânicos ou físicos, por falhas de comunicação no enlace sem fio ou por estarem adormecidos para economizar energia. A topologia da rede também pode ser alterada quando novos nós são adicionados à rede já existente.

1.2 Definição do Problema

Nesse cenário de desafios caracterizado pelas restrições dos nós sensores e pela topologia dinâmica, a atividade de comunicação de dados torna-se um dos principais tópicos de pesquisa. Isso ocorre porque o custo de comunicação é normalmente o mais significativo nas RSSFs, sendo aproximadamente três ordens de grandeza superior ao de processamento (Pottie e Kaiser, 2000a). Logo, qualquer solução para a comunicação de dados nesse tipo de rede deve ser eficiente, em termos de energia, para aumentar o tempo de vida da rede.

A comunicação de dados em RSSFs pode ser dividida em três casos (Goussevskaia et al., 2005b), como mostrado na figura 1.1: dos sensores para o *sink* (coleta de dados), entre os nós sensores vizinhos (cooperação), e do nó *sink* para os nós sensores (disseminação de dados). Este trabalho aborda um tipo específico de disseminação de dados que acontece quando o nó *sink* envia uma informação para todos os nós da rede. Esse tipo de disseminação de dados é denominada difusão de dados (*broadcasting*). Outros tipos de disseminação de dados são *unicasting*, quando a informação disseminada é destinada a um único nó, e *multicasting*, quando a informação é enviada para vários destinatários. A partir de algoritmos de difusão de dados eficientes, o nó *sink* pode realizar atividades tais como alterar o modo de operação da rede, disseminar um novo interesse, ativar/desativar um ou mais sensores e enviar consultas para a rede. A difusão de dados é fundamental para o funcionamento básico de vários protocolos e aplicações em RSSFs. Além disso, alguns protocolos para a coleta de dados, como o *Directed Diffusion* (Intanagonwiwat et al., 2002), também são dependentes de um protocolo de difusão de dados.

Vários algoritmos de difusão de dados têm sido propostos na literatura (Orecchia et al., 2004; Chang et al., 2005; Rahnavard e Fekri, 2006a; Durresi et al., 2005). Dentre eles, o mais simples é o *flooding* no qual, sempre que um nó recebe um pacote pela primeira vez, ele o retransmite para seus vizinhos. A desvantagem desse algoritmo é o elevado número de transmissões realizadas que o torna ineficiente em termos de energia. Uma versão otimizada do *flooding* é o *gossiping*, ou *flooding* probabilístico, no qual cada nó transmite uma mensagem com uma probabilidade p . O desempenho do algoritmo em relação ao número de nós cobertos e ao número de transmissões é sensivelmente dependente da escolha do valor de p . Alguns protocolos para difusão de dados utilizam o conceito de *forwarding delay* para reduzir o número de transmissões. Nesses algoritmos, quando um nó recebe um pacote, ele espera um determinado intervalo de tempo para decidir se irá retransmiti-lo ou não. Após esse intervalo, o nó propaga o pacote se nenhum vizinho o tiver retransmitido. O ponto importante dessa política é a definição do intervalo de espera, que é proporcional a alguma métrica a ser otimizada como a distância ou a energia. No caso da distância, deseja-se que os nós mais próximos (ou os mais distantes) a um determinado ponto transmitam o pacote.

1.3 Objetivos

Devido à importância do uso eficiente de energia em RSSFs, a informação sobre a quantidade de energia disponível em cada parte da rede pode ser de suma relevância

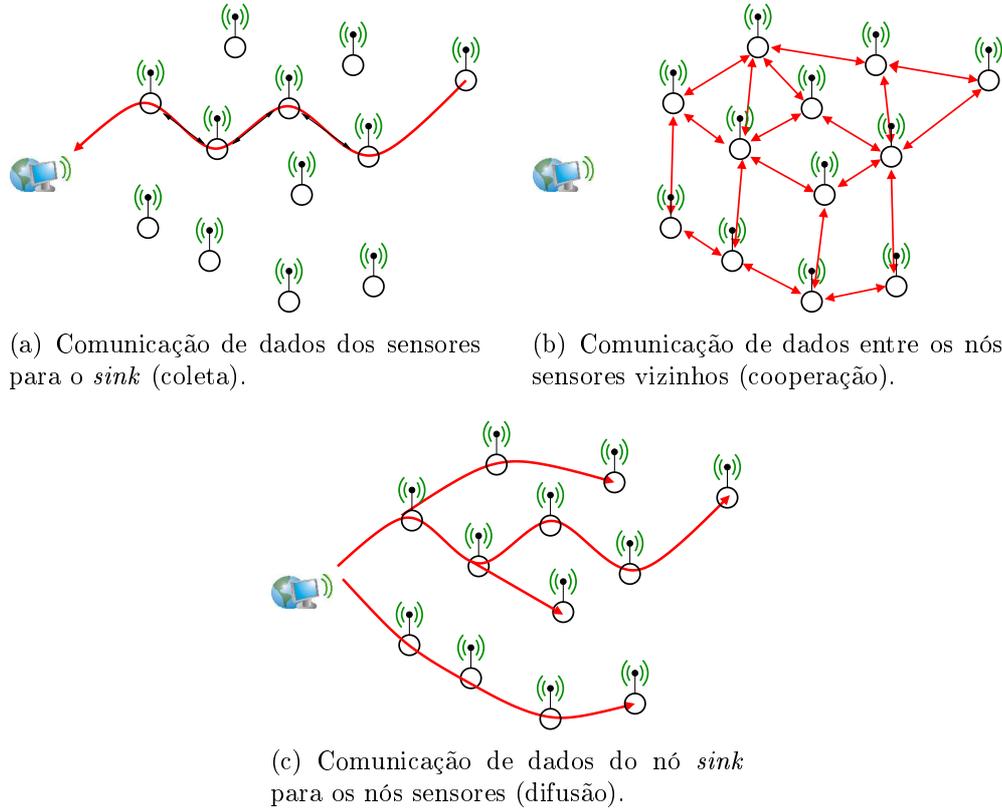


Figura 1.1: Formas de comunicação de dados em redes de sensores sem fio.

para prolongar o tempo de vida da mesma. Essa informação é denominada mapa de energia (Mini et al., 2005) e pode ser utilizada por diferentes atividades com o objetivo de otimizar o consumo de energia disponível na rede. Algoritmos distribuídos cientes do mapa de energia podem alterar seu modo de funcionamento quando forem executados em nós sensores localizados em regiões de baixa energia. Por exemplo, protocolos de difusão de dados podem evitar rotas que passam por tais regiões.

Neste trabalho, é proposto o *Delay and Energy Based Broadcasting* (DEBB) (Guidoni et al., 2006), um protocolo que combina os conceitos de *forwarding delay* e de mapa de energia para realizar a difusão de dados eficiente em termos de energia em RSSFs. O DEBB é composto por dois módulos e o primeira é o de determinar a energia de corte. O *sink* executa essa função para analisar o mapa de energia e detectar regiões de baixa energia na rede. Se alguma região de baixa energia for detectada, um parâmetro contendo informações sobre a região de baixa energia é enviado no pacote a ser difundido na rede. Assim, os nós saberão se eles pertencem ou não a uma região de baixa energia.

O segundo módulo é o de *forwarding delay*. Nesse caso, o nó sensor irá gerar um atraso para enviar o pacote. Esse atraso é proporcional à métrica de distância e energia. Ou seja, quanto maior a energia e a distância para o nó que enviou o pacote, menor será

o atraso inserido na propagação desse. O objetivo do DEBB é reduzir a quantidade de pacotes transmitidos, por conseguinte, a energia consumida na rede, maximizar o número de nós que recebeu o pacote de dados difundido e fazer com que os nós que possuem baixa reserva de energia não participem da difusão. Como o DEBB insere atrasos na propagação do pacote, a latência na difusão é significativamente aumentada. Assim, é proposto uma versão probabilística do DEBB, chamado DEBB-P ou DEBB probabilístico. Nesse caso, o nó calcula uma probabilidade para enviar o pacote. Essa probabilidade é calculada segundo a métrica de distância. O objetivo do DEBB-P é diminuir a latência causada pelo uso da técnica de *forwarding delay*.

O presente trabalho apresenta os dois algoritmos com suas avaliações feitas com simulações, em que várias difusões de dados são realizadas pelo nó *sink*. As métricas de avaliação desses algoritmos são o consumo de energia, número de nós que receberam o pacote difundido, número de nós mortos, operações aritméticas e latência. Resultados de simulação mostram que, em cenários de difusão de dados, o DEBB é mais adequado que os demais protocolos comparados. Nesses cenários, o DEBB apresenta um custo de execução reduzido para os nós sensores e possui um melhor desempenho em termos de número de nós cobertos, número de transmissões realizadas e energia consumida. Além disso, o DEBB apresenta um comportamento escalável e robusto para difusão de dados em cenários contendo regiões de baixa energia. Nesse caso, o DEBB faz com que os dados sejam integralmente roteados pelos nós com maiores reservas energia, mesmo quando o número de regiões de baixa energia aumenta. Por outro lado, o DEBB-P consegue reduzir significativamente a latência causada pela inserção de atrasos mantendo as mesmas características de nós que receberam o pacote de difundido, energia consumida e números de pacotes transmitido do DEBB.

1.4 Organização do Trabalho

Este trabalho está organizado da seguinte maneira. O capítulo 2 apresenta os trabalhos relacionados aos algoritmos projetados nesta dissertação. O capítulo 3 apresenta o protocolo *Delay and Energy Based Broadcasting* (DEBB) e sua versão probabilística. O capítulo 4 apresenta os resultados de simulação em um cenário em que várias difusões de dados são realizadas pelo nó *sink* para todos os nós sensores da rede. Nesse capítulo, o DEBB é comparado com outros protocolos encontrados na literatura. Finalmente, o capítulo 5 apresenta as conclusões e os trabalhos futuros.

Capítulo 2

Trabalhos Relacionados

Este capítulo apresenta os trabalhos relacionados aos algoritmos propostos nesta dissertação. Ele está organizado como descrito a seguir. A seção 2.1 apresenta os protocolos clássicos de roteamento propostos para redes de sensores sem fio. A seção 2.2 apresenta de forma específica e detalhada, os protocolos de difusão mais relacionados com o protocolo proposto neste trabalho. Como o protocolo proposto neste trabalho utiliza informações de energia da rede, a seção 2.3 apresenta o conceito de mapa de energia proposto em (Mini, 2004). A seção 2.4 apresenta o modelo de energia *State-based Energy Dissipation Model* (SEDM), proposto em (Mini et al., 2005). Finalmente, a seção 2.5 apresenta as conclusões do presente capítulo.

2.1 Roteamento em Redes de Sensores Sem Fio

Na comunicação de dados, o roteamento é um problema fundamental que pode ser definido como o processo de enviar pacotes de um elemento origem até um elemento destino através de uma rota pela qual o pacote deve percorrer. Algoritmos de roteamento tem como objetivo obter a melhor rota a ser percorrida pelos pacotes considerando as informações de roteamento coletadas na rede, como por exemplo, informações sobre atraso e congestionamento de um enlace entre pares de nós roteadores. Normalmente, é possível determinar o “custo” de se enviar um pacote através de um enlace ou de um roteador. Esse custo pode ser utilizado pelos algoritmos de roteamento para determinar as melhores rotas.

Assim como em redes *ad hoc*, o roteamento é um problema importante em RSSFs. Em ambos os tipos de redes, a grande dificuldade existente para a realização do roteamento ocorre em função da topologia dinâmica. Nas redes *ad hoc*, a topologia dinâmica é uma consequência da mobilidade dos nós. Dessa forma, as rotas válidas tornar-se-ão inválidas em um futuro próximo devido à mobilidade dos nós. Em RSSFs, apesar dos

nós serem geralmente estáticos, a topologia da rede também é dinâmica. Nós desligam o rádio para economizar energia, nós morrem (por exemplo, por falta de energia ou por um problema físico), novos nós são adicionados, e o *sink* muda sua localização. Na literatura, vários protocolos de roteamento para RSSFs foram propostos (Mann et al., 2005; Braginsky e Estrin, 2002; Su e Akyildiz, 2005; Ganesan et al., 2001; Heinzelman, 2000; Wood et al., 2006; Intanagonwiwat et al., 2002; Servetto e Barrenechea, 2002; Yu et al., 2001), a maior parte deles utiliza uma ou mais técnicas para economizar energia.

Devido a sua natureza, as RSSFs requisitam técnicas de roteamento escaláveis e robustas para disseminação de dados (Ganesan et al., 2001). Os algoritmos para essas redes devem ser projetados objetivando elevar o tempo de vida da rede. Logo, os mesmos devem prover um mecanismo de comunicação robusto e com consumo de energia eficiente. A seguir, são descritos alguns algoritmos propostos na literatura.

Intanagonwiwat et al. (2000) propõem o *Directed Diffusion* um novo paradigma para a comunicação entre os nós sensores. O objetivo desse protocolo é estabelecer uma comunicação eficiente entre os nós sensores e o *sink*. O modelo proposto introduz dois conceitos. O primeiro é o *data-centric* em que os dados gerados pelos sensores são identificados por um par de valor-atributo. Nesse caso, quando o *sink* deseja saber informações sobre um dado específico, ele envia uma identificação do par valor-atributo desejado para a rede – esse processo é chamado envio de interesse. O outro conceito é o *data aggregation* em que os nós intermediários procuram agregar os eventos recebidos em um único evento com o objetivo de reduzir o número de transmissões realizadas e a quantidade de dados armazenados na rede. O funcionamento do *Directed Diffusion* pode ser resumido da seguinte forma. O *sink* dissemina uma tarefa de sensoriamento ou interesse para a rede e os nós intermediários propagam esse interesse através de interações locais. O caminho da propagação do interesse estabelece o caminho reverso para os dados coletados localmente que “casam” com aquele interesse. Essa disseminação cria um gradiente a partir da topologia da rede que se torna direcionada a eventos. O gradiente faz com que os dados coletados percorram caminhos únicos na volta. O *Directed Diffusion* é aprimorado em (Intanagonwiwat et al., 2002, 2003) em que os autores propõem uma abordagem de energia mais eficiente para a agregação de dados.

Em (Heinzelman et al., 1999) e (Kulik et al., 1999), os autores apresentam uma família de protocolos adaptativos, denominada SPIN (*Sensor Protocols for Information via Negotiation*), que dissemina informação de forma eficiente em RSSFs. A família de protocolos SPIN incorpora duas inovações importantes com objetivo de superar as deficiências das abordagens para a disseminação de dados existentes na literatura: negociação e adaptação de recursos. Os nós negociam entre si antes das transmissões de dados. Essa estratégia ajuda assegurar que apenas informações úteis serão transmitidas e que os nós não gastem energia com transmissões desnecessárias. A adaptação de

recursos ocorre quando os nós conhecem os recursos existentes nas regiões onde se encontram e, assim, eles podem planejar melhor suas respectivas atividades de forma a aumentar o tempo de vida da rede.

Alguns algoritmos de roteamento assumem a existência de um sistema de localização que permite aos nós conhecerem suas respectivas posições. Um exemplo deste tipo de algoritmo é o *Geographic and Energy Aware Routing* (GEAR) proposto por (Yu et al., 2001). O GEAR usa a energia e o sistema de localização geográfico para rotear os pacotes até suas respectivas regiões de destino. Esse protocolo procura balancear o consumo de energia de forma a elevar o tempo de vida da rede.

(Heinzelman, 2000; Heinzelman et al., 2000) propõem o algoritmo LEACH (Low-Energy Adaptive Clustering Hierarchy), um protocolo baseado em *clustering* que minimiza o consumo de energia em RSSF. Esse algoritmo realiza rotações aleatórias do líder do *cluster* (*cluster-head*) para distribuir o consumo de energia entre os sensores da rede. Compressões locais também são utilizadas para reduzir as comunicações globais.

2.2 Difusão em Redes de Sensores Sem Fio

Recentemente, alguns autores têm proposto eficientes protocolos para difusão em redes de sensores (Cartigny et al., 2005; Ovalle-Martinez, 2005; Ingelrest et al., 2006; Wu e Dai, 2005, 2004; Rahnavard e Fekri, 2006b; Durresi et al., 2005; Heissenbüttel et al., 2006; Goussevskaia et al., 2005b; do Val Machado et al., 2005). Entre os algoritmos existentes, os que possuem maior relação com esse trabalho são o *Trajectory and Energy Based Data Dissemination* (Goussevskaia et al., 2005b; do Val Machado et al., 2005), o *Dynamic Delayed Broadcasting* (Heissenbüttel et al., 2006), o *Broadcast Protocol for Sensor Networks* (Durresi et al., 2005), *flooding* e o *gossiping* e eles são estudados a seguir.

2.2.1 *Trajectory and Energy Based Data Dissemination* (TEDD)

Algoritmos para roteamento em curva utilizam a técnica de disseminação de dados na qual pacotes são disseminados pelo nó *sink* para um conjunto de nós ao longo de uma curva pré-definida. A idéia principal consiste em inserir uma equação de curva no pacote e os nós intermediários propagam esse pacote para seus vizinhos que estiverem localizados próximos à curva. O TBF (*Trajectory Based Forwarding*), proposto por (Niculescu e Nath, 2003), foi o primeiro algoritmo a utilizar essa idéia para rotear pacotes. O TEDD (Goussevskaia et al., 2005b; do Val Machado et al., 2005) é um protocolo que combina os conceitos de roteamento em curva e de mapa de energia

para disseminar informações de forma eficiente, em termos de energia, em RSSFs. O protocolo é composto de duas partes. Inicialmente, quando deseja disseminar uma informação, o nó *sink* aciona um módulo de geração dinâmica de curvas que realiza a leitura do mapa de energia e gera curvas que passam por regiões com maiores reservas de energia e evitam o fluxo de dados dentro das regiões de menor energia. A segunda parte do algoritmo consiste em uma política de disseminação do tipo *receiver-based*, ou seja, quando um nó recebe um pacote, ele próprio decide se deve ou não propagá-lo. O método de decisão utiliza uma técnica de *forwarding delay*. Nesse caso, quando um nó recebe um pacote ele espera um determinado intervalo de tempo para decidir se irá retransmití-lo. Após esse intervalo, o nó propaga o pacote se nenhum vizinho tiver retransmitido. O ponto importante dessa política é a definição do intervalo de espera. Em (Goussevskaia et al., 2005a), quanto menor a distância do nó em relação a um ponto a frente na curva (ponto de referência), menor o tempo de espera. Os resultados obtidos pelo TEDD são significativos porque ele apresenta várias melhorias em relação ao roteamento em curva original (Niculescu e Nath, 2003), utiliza o conceito de mapa de energia e é o primeiro trabalho sobre a geração dinâmica de curvas de roteamento em redes sem fio.

A idéia principal do TEDD é que o ponto de referência sempre seja um ponto mais a frente na curva e que sistematicamente o nó mais próximo a ele propague o pacote. O uso do ponto de referência é feito de duas formas. Na primeira, existe um único ponto de referência localizado próximo à curva e, conseqüentemente, a disseminação ocorre através de um único fluxo de dados que é roteado em cima da curva. Como ilustrado na figura 2.1, quando um único fluxo é utilizado, o nó mais próximo ao ponto de referência (figura 2.1, ponto *B*) propaga o pacote. Na outra forma, existem dois pontos de referência, ambos são pontos à frente na curva e suficientemente distantes para que existam dois fluxos de dados independentes sendo propagados paralelamente à curva (um acima e outro abaixo da mesma). Como mostrado na figura 2.1, quando dois fluxos são utilizados, os dois nós mais próximos respectivamente aos pontos de referência (figura 2.1, pontos *A* e *C*) propagam o pacote. A escolha de quantos fluxos utilizar é conseqüência do objetivo da disseminação. Existe um compromisso entre maximizar a quantidade de nós cobertos pela curva e minimizar o número de transmissões. Em situações em que o mais importante é minimizar o número de transmissões, a política de um fluxo deve ser utilizada. Por outro lado, quando maximizar a cobertura for o objetivo principal, a política de dois fluxos é a mais indicada.

A figura 2.2 ilustra o modo de funcionamento básico do TEDD. Quando um nó recebe um pacote, ele verifica se sua coordenada está dentro do setor de curva recebido (toda curva possui um setor angular e cada curva é válida apenas dentro de seu setor) (figura 2.2, ponto *A*). Se não estiver, o nó descarta o pacote (figura 2.2, ponto *B*).

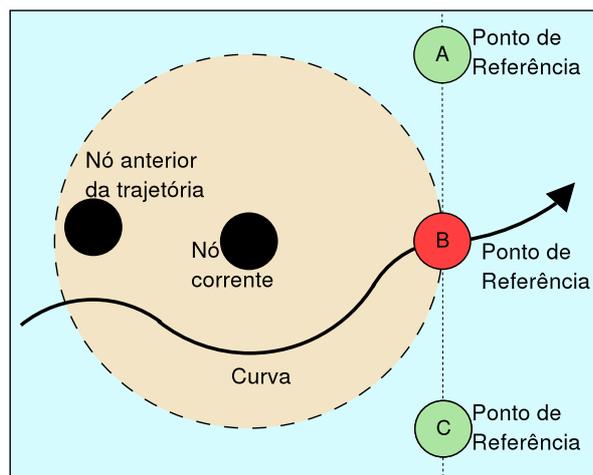


Figura 2.1: Pontos de referência.

Se o nó estiver dentro do setor, é calculado as coordenadas do ponto de referência (figura 2.2, ponto *C*) e verifica se sua distância para o ponto de referência é maior que o raio de comunicação (figura 2.2, ponto *D*). Se sim, o nó descarta o pacote (figura 2.2, ponto *B*). Caso contrário, o nó calcula o tempo de espera (figura 2.2, ponto *E*). Após aguardado o tempo de espera (figura 2.2, ponto *F*), o nó verifica se algum de seus vizinhos retransmitiu o pacote (figura 2.2, ponto *G*). Em caso afirmativo, o nó descarta o pacote (figura 2.2, ponto *B*). Caso contrário, o nó calcula o ponto de referência (para auxiliar na definição do próximo nó da rota) (figura 2.2, ponto *H*) e, em seguida, ele propaga o pacote (figura 2.2, ponto *I*).

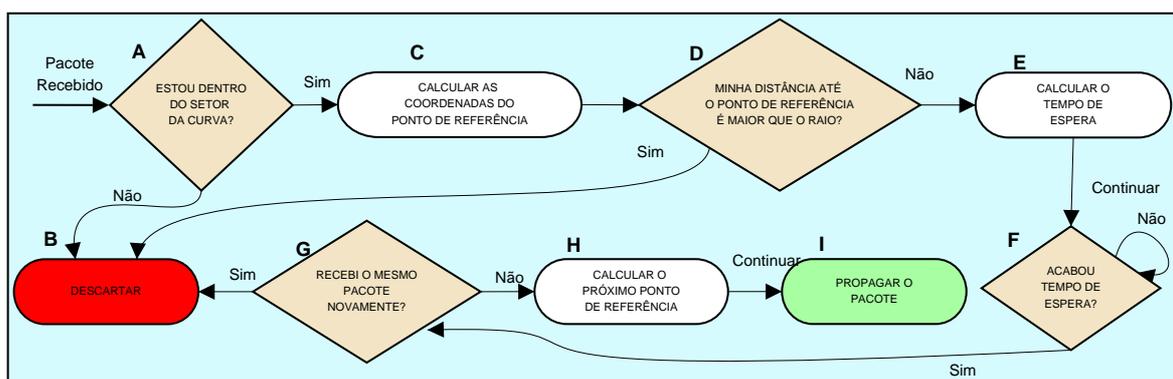
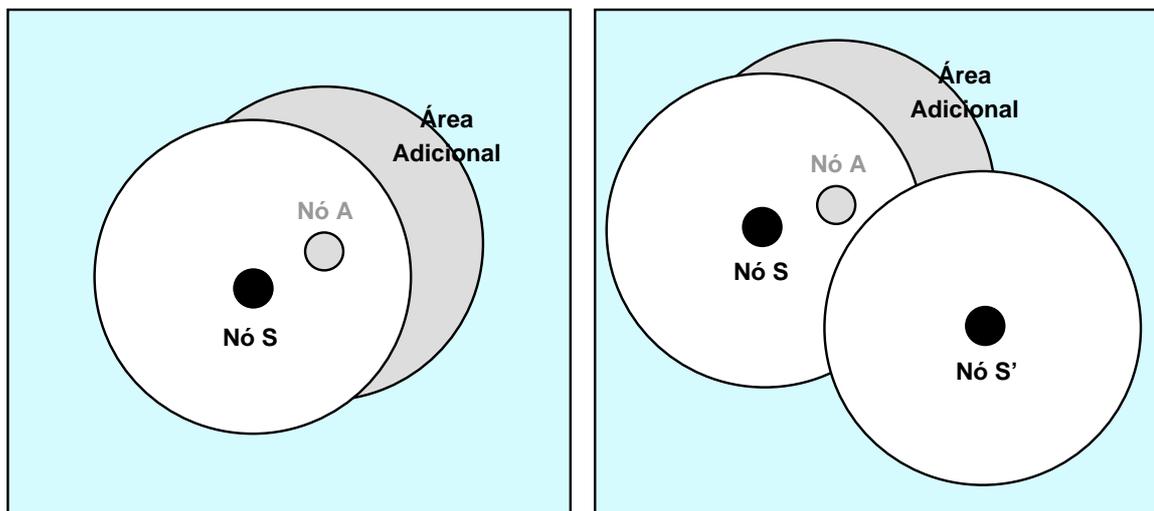


Figura 2.2: Funcionamento básico do TEDD.

2.2.2 *Dynamic Delayed Broadcasting (DDB)*

Em (Heissenbüttel et al., 2006; Heissenbüttel, 2005) os autores apresentam o Dynamic Delayed Broadcasting (DDB), um protocolo eficiente para difusão de dados em RSSF

que utiliza o conceito de *forward delay*. O *forwarding delay* é utilizado com o objetivo de otimizar o número de nós que propagam o pacote de difusão e o tempo de vida da rede. São propostas duas versões do DDB. A primeira chamada de DDB1, quando o nó recebe um pacote, ele calcula o atraso na transmissão baseado na estimativa da área adicional que será coberta com a sua transmissão, ou seja, a métrica utilizada para o cálculo do atraso do pacote é a área que será coberta com a sua transmissão que ainda não foi coberta por nenhuma transmissão de outro nó. A figura 2.3(a) mostra a área adicional estimada pelo nó *A* quando ele recebe um pacote transmitido pelo nó *S*. A figura 2.3(b) mostra a área adicional estimada pelo nó *A* quando ele recebe o mesmo pacote duas vezes, uma pelo nó *S* e outra pelo nó *S'*. Se a área adicional coberta pela transmissão do nó for menor que um *threshold* pré determinado, o nó não retransmite o pacote. Deve-se destacar que essa métrica tem como objetivo reduzir a quantidade de pacotes transmitidos na difusão.



(a) Nó *A* recebe um pacote transmitido pelo nó *S*.

(b) Nó *A* recebe o mesmo pacote duas vezes: uma transmitida pelo nó *S* e outra transmitida pelo nó *S'*.

Figura 2.3: Exemplo da área adicional estimada pelo nó *A* quando o DDB1 é utilizado.

O funcionamento básico do DDB1 é mostrado na figura 2.4. Quando o nó recebe o pacote pela primeira vez, ele calcula a área adicional que será coberta pela sua transmissão (figura 2.4, ponto *A*). Se a área calculada é menor do que um *threshold* pré-definido (figura 2.4, ponto *B*), o nó descarta o pacote (figura 2.4, ponto *C*). Se a área for maior, o nó calcula o tempo de atraso do pacote inversamente proporcional a essa área (figura 2.4, ponto *D*). A equação 2.1 apresenta a fórmula utilizada para o cálculo do tempo de espera. O atraso máximo que um pacote pode ter em cada nó é denominado *Max_Delay*. *AC* é a área coberta se o nó executar a retransmissão e *e* é o Número de Euler.

$$Time_Delay = Max_Delay * \sqrt{\frac{e - e^{\left(\frac{AC}{1.91}\right)}}{e - 1}} \quad (2.1)$$

Após o cálculo do tempo de espera, o nó irá esperar $Time_Delay$ segundos antes de sua transmissão (figura 2.4, ponto *E*). Durante esse período, se o nó receber outra cópia desse mesmo pacote, o agendamento da transmissão será cancelado e a área adicional coberta por sua transmissão é recalculada (figura 2.4, ponto *A*). A nova área é igual a área calculada anteriormente menos a área coberta pelo último pacote recebido. Se o tamanho da nova área for menor que o *threshold* (figura 2.4, ponto *B*) o nó descarta o pacote (figura 2.4, ponto *C*). Senão, um novo tempo de atraso é calculado. Entretanto, o tempo de espera é calculado inversamente proporcional a nova área menos a fração do tempo de espera que o nó já realizou, antes de receber a cópia do pacote pela segunda vez. Esse processo é repetido até que a área coberta pelo nó seja menor que o *threshold*. Quando o tempo de atraso acabar, o nó insere a sua localização geográfica (valores x e y) no pacote (figura 2.4, ponto *F*) e efetua a sua propagação.

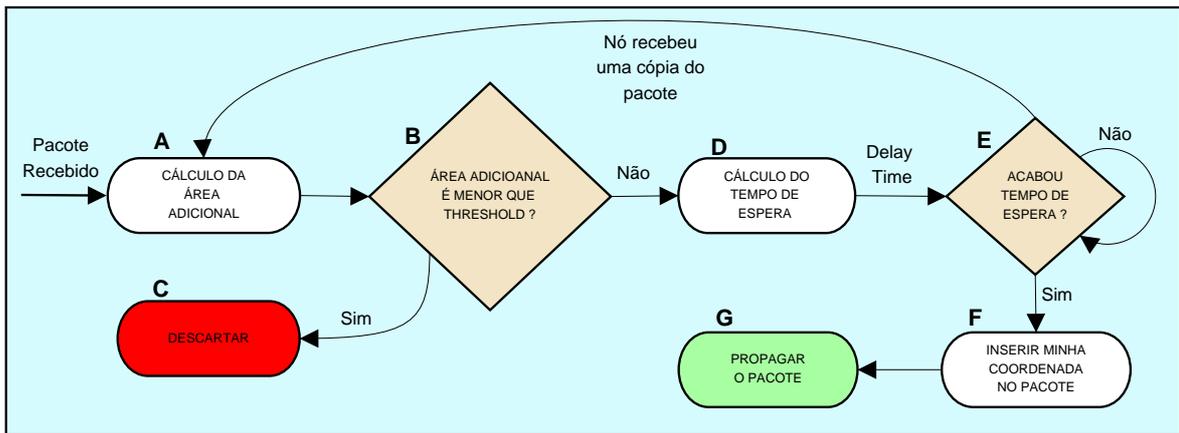


Figura 2.4: Funcionamento básico do DDB.

Segundo Heinssenbüttel, quando o DDB1 é utilizado em cenário como RSSF, onde os nós raramente se movem e tráfego de dados é altamente direcionado ($sink \rightarrow$ rede e rede $\rightarrow sink$), alguns nós consumirão energia mais rapidamente do que outros. Isso acontece pois, o algoritmo não leva em consideração a energia que o nó possui e, conseqüentemente, os mesmo nós transmitirão o pacote de difusão. Para resolver esse problema, os autores propõem a segunda versão do DDB, chamada de DDB2. O funcionamento básico do DDB2 é o mesmo do DDB1, como mostrado na figura 2.4. A diferença entre eles está no cálculo do tempo de espera (figura 2.4, ponto *D*). No DDB2, o tempo de espera é inversamente proporcional a energia corrente do nó, e o conceito de área adicional coberta pela transmissão do pacote continua sendo utilizado, entretanto, é utilizado apenas para verificar quando a área adicional coberta pela transmissão é

menor do que o *threshold*. Se a área coberta for menor que o *threshold*, o nó descarta o pacote. Para calcular o tempo de espera no DDB2 é utilizado a equação 2.2:

$$Time_Delay = Max_Delay * \sqrt{\frac{e - e^{E_B}}{e - 1}} \quad (2.2)$$

onde, e é o número de Euler, Max_Delay é o atraso máximo que um pacote pode ter no nó e E_B é a energia restante no nó em porcentagem da sua energia inicial.

2.2.3 Broadcast Protocol for Sensor Networks (BPS)

O BPS, proposto por (Durrezi et al., 2005), é um protocolo de difusão que utiliza a técnica de *forwarding delay* com a métrica de distância. Seu principal objetivo é minimizar o número de pacotes transmitidos em cada difusão. Quando um nó deseja enviar uma mensagem para todos os outros nós da rede, ele se torna o centro de um hexágono inscrito em seu raio de comunicação e , todos os seus vizinhos calculam sua distância em relação ao vértice mais próximo do hexágono. Um atraso inversamente proporcional a essa distância é calculado e, após esse tempo, o nó retransmite o pacote. Dessa forma, o nó que se encontra mais próximo a algum vértice do hexágono terá um tempo de espera menor e será o primeiro a retransmitir o pacote. Se um nó recebeu um pacote retransmitido por algum vizinho, ele não precisa retransmiti-lo. Uma grande desvantagem do BPS é que, como a informação de energia não é utilizada para efetuar a difusão, regiões de baixa energia na rede não são evitadas na comunicação e, sempre que um determinado nó executar uma difusão, o mesmo subconjunto de nós será escolhido sistematicamente para retransmitir o pacote, proporcionando uma morte prematura dos nós escolhidos.

A figura 2.5 ilustra o funcionamento básico do BPS, onde ‘th’ é a abreviatura para *thresohld*. Quando o nó N recebe o pacote pela primeira vez (figura 2.5, ponto A), ele verifica se o nó que enviou o pacote está a uma determinada distância do nó atual. Se a distância for menor que um determinado *threshold* pré-definido, o pacote é descartado (figura 2.5, ponto B). Caso contrário, verifica se o pacote já foi recebido pelo nó (figura 2.5, ponto C). Se o pacote já foi recebido, ele deve ser descartado (figura 2.5, ponto B). Caso contrário, o nó N calcula a distância dele ao vértice mais próximo do hexágono inscrito em seu raio de comunicação (figura 2.5, ponto D). Após essa etapa, o nó calcula o atraso no envio do pacote (figura 2.5, ponto E). Esse cálculo pode ser feito de duas formas. Na primeira, se o nó que enviou o pacote é a fonte S da difusão, o nó que recebeu o pacote encontra o vértice V mais próximo do hexágono com centro nas coordenadas de S e como exemplo de um vértice desse hexágono tem-se $(S_x + R, S_y)$, onde $S = (S_x, S_y)$ e R é o raio de comunicação do nó. O nó calcula a distância l

para o vértice V e o atraso é dado pela fórmula l/R . Na segunda forma, se o pacote foi enviado por um nó K qualquer da rede, o nó N seleciona o vértice do hexágono mais próximo de sua posição e o atraso do pacote é dado pela fórmula $l/20*R$. Após esse cálculo, o nó espera o tempo necessário para a transmissão do pacote (figura 2.5, ponto F) e, após isso, verifica-se se o pacote já foi recebido (figura 2.5, ponto G). Se o pacote já foi recebido, ele deve ser descartado (figura 2.5, ponto B). Caso contrário, ele insere as suas coordenadas geográficas no pacote (figura 2.5, ponto H) e efetua sua propagação (figura 2.5, ponto I).

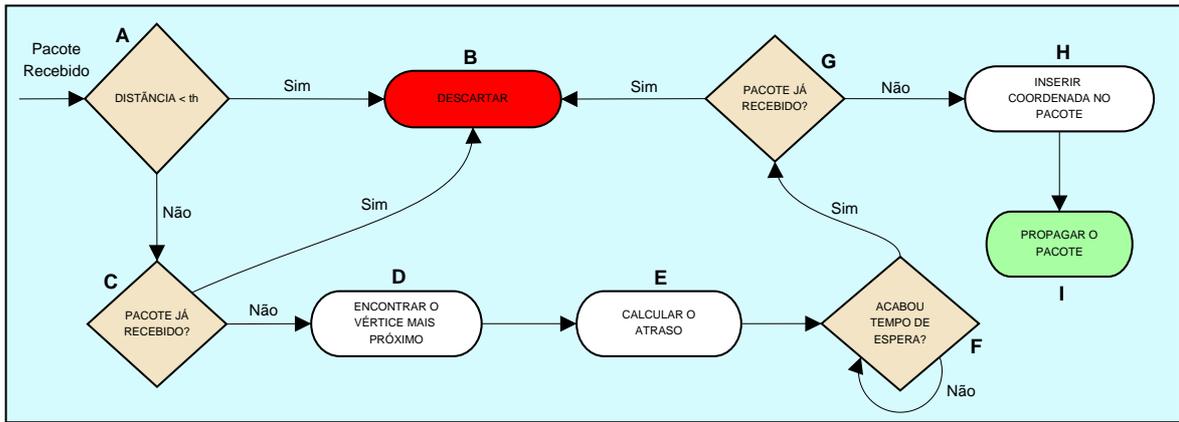


Figura 2.5: Funcionamento básico do BPS.

2.2.4 Flooding e Gossiping

O algoritmo de difusão mais simples é o *flooding* no qual, sempre que um nó recebe um pacote pela primeira vez, ele o retransmite. A desvantagem desse algoritmo é o elevado número de transmissões realizadas que o torna ineficiente em termos de energia. Uma versão otimizada do *flooding* é o *gossiping*, ou *flooding* probabilístico, no qual cada nó transmite uma mensagem com uma probabilidade p . O *gossiping* apresenta um comportamento distinto em função da densidade da rede e da probabilidade utilizada. Se a rede for esparsa ou a probabilidade for pequena, as rotas são quebradas com muita facilidade e poucos nós são cobertos pelo algoritmo. Por outro lado, em redes densas ou quando a probabilidade for suficiente, o protocolo apresenta um desempenho satisfatório em relação ao número de nós cobertos e ao número de transmissões (do V. Machado et al., 2005).

2.3 Mapa de Energia

Em RSSFs, o custo da comunicação de dados pode ser representado pelo consumo de energia. A informação sobre a energia restante em cada parte da rede é denominada de mapa de energia. Esse mapa pode ser representado em tons de cinza como ilustrado na Figura 2.6. As regiões mais claras representam áreas com maior quantidade de energia e as mais escuras correspondem às regiões com menos energia. Utilizando o mapa, é possível determinar se alguma parte da rede pode sofrer falha devido à falta de energia (Zhao et al., 2002). Várias aplicações para as redes de sensores sem fio podem utilizar a informação fornecida pelo mapa, como algoritmos de disseminação de dados, de reconfiguração, de fusão de dados, e de gerenciamento da rede. O ponto importante é que o mapa de energia é fundamental para prolongar o tempo de vida da rede.

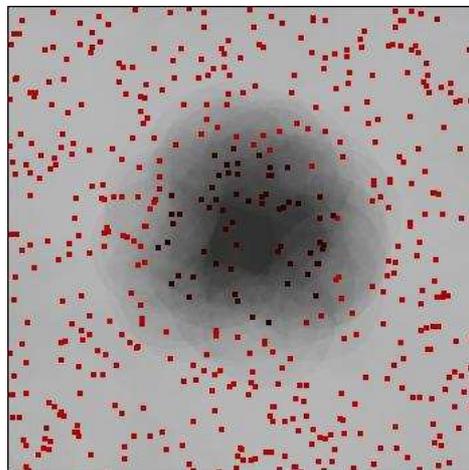


Figura 2.6: Mapa de energia de uma rede de sensores sem fio.

O mapa de energia pode ser construído de várias maneiras. Uma delas é utilizando uma técnica ingênua em que, periodicamente, cada nó sensor envia o seu respectivo valor de energia para o *sink*. Contudo, essa abordagem apresenta um custo elevado, em termos de energia, devido à grande quantidade de transmissões necessárias para atualizar o mapa. Nesse caso, possivelmente, os ganhos advindos do conhecimento fornecido pelo mapa seriam incapazes de cobrir os custos do processo necessário para sua obtenção/atualização.

Zhao et al. (2002) propõem uma abordagem mais interessante em que o mapa é construído utilizando técnicas de agregação. O trabalho proposto por (Zhao et al., 2002) obtém o mapa de energia de uma rede de sensores utilizando uma abordagem baseada em agregação. Um nó sensor apenas precisa enviar para o nó *sink* sua energia local quando existe uma queda significativa quando comparada com a última vez que o nó reportou sua energia disponível. Ao longo do caminho para o *sink*, os nós que

receberem duas ou mais informações de energia podem agregá-las de acordo com várias regras. Se as informações de energia são de áreas topologicamente adjacentes e se têm níveis de energia semelhante, elas podem ser agregadas. O objetivo da agregação é reduzir o custo de coletar o dado de energia, mas mantendo a qualidade da informação obtida. No trabalho de (Zhao et al., 2002), são apresentados resultados de simulação que comparam as abordagens propostas com uma abordagem centralizada. Entretanto, nas simulações, não é levado em consideração o custo da atualização periódica da árvore de agregação.

Mini (2004); Mini et al. (2005, 2003a, 2002, 2003b) propõem outra abordagem eficiente, baseada em mecanismos de cadeias de Markov, para prever o consumo de energia de um nó sensor e com essa informação construir o mapa de energia. A idéia dessa proposta está relacionada à situações em que um nó sensor pode prever seu consumo de energia baseando-se em seu passado. Se um nó pode prever eficientemente a quantidade de energia que ele irá gastar no futuro, ele não precisa transmitir freqüentemente o valor de sua energia. Nesse caso, um nó sensor pode enviar uma única informação contendo o valor de sua energia e os parâmetros que descrevem seu consumo. Usando esses parâmetros, o nó *sink* pode atualizar localmente a informação de energia dos nós sensores. Resultados de simulação apresentados por (Mini et al., 2005) mostram que o uso de modelos baseados em predição apresenta um bom desempenho e diminui a quantidade de energia necessária para a construção do mapa. Além disso, em (Mini et al., 2005), o custo de construção do mapa é mostrado detalhadamente através do número de operações (adição, subtração, multiplicação, divisão, comparação e atribuição) necessárias. Outro ponto importante tratado em (do V. Machado et al., 2005) e em (Mini et al., 2005) é a utilização de técnicas de amostragem para reduzir ainda mais o custo de construção do mapa. É importante destacar que mesmo existindo técnicas otimizadas (em termos de energia) para a construção do mapa, o custo de construção do mesmo pode ser dividido por todas as aplicações e/ou algoritmos que se beneficiam dele. Outra consideração importante é que o *Delay and Energy Based Broadcasting* (DEBB), protocolo de difusão de dados proposto no presente trabalho, utiliza a abordagem baseada em predição para obter o mapa de energia.

2.4 *State-based Energy Dissipation Model* (SEDM)

O uso de simulação para avaliar o desempenho de protocolos de roteamento ou de qualquer outro algoritmo relacionado a energia em redes de sensores sem fio, apresenta como premissa a existência de um modelo de dissipação de energia dos nós sensores.

Em (Zhao et al., 2002), durante um evento de sensoriamento, alguns pontos h , pré-definidos na rede, apresentam uma probabilidade p de iniciarem uma atividade de sensoriamento local, e todo nó localizado dentro de um círculo de raio r e de centro igual a um ponto h consomem uma quantidade fixa de energia. Uma desvantagem dessa proposta é que quando um evento ocorre, todos os nós localizados dentro de sua área de influência conseguem identifica-lo imediatamente. Essa situação é plausível apenas se todos os nós sensores estiverem com seus respectivos sensores ligados durante todo o tempo de simulação. Contudo, essa solução não é interessante porque a melhor forma de se economizar energia em RSSFs é, sempre que possível, manter inativos, os componentes que não tiverem sendo utilizados (Hill et al., 2000). Outra limitação do modelo proposto em (Zhao et al., 2002) é que ele não modela as duas primeiras formas de comunicação de dados apresentadas na figura 1.1: entre os nós sensores e o *sink*; e entre os nós sensores. Finalmente, esse modelo assume que todos os nós sensores dentro da região de sensoriamento apresentam o mesmo consumo. Isso não é necessariamente verdade, uma vez que para economizar energia, nem todos os nós devem sensoriar um mesmo evento.

O outro trabalho que aborda a modelagem do consumo de energia em nós sensores é o *State-based Energy Dissipation Model* (SEDM) proposto em (Mini, 2004; Mini et al., 2005, 2003a, 2002, 2003b). No SEDM, os nós possuem vários modos de operação com diferentes níveis de ativação e, conseqüentemente, diferentes níveis de consumo de energia. Nesse modelo, cada nó apresenta quatro modos de operação: Modo 1: sensor desligado, processador em *idle*, e radio desligado. Modo 2: sensor e processador ligados, e radio desligado. Modo 3: sensor e processador ligados, e radio recebendo; Modo 4: sensor e processador ligados, e radio transmitindo. As transições entre os modos ocorrem conforme descrito no diagrama da figura 2.7. No diagrama, os modos de operação são representados pelos estados 1, 2, 3 e 4. Observa-se também a presença de dois estados complementares $2'$ e $3'$. O estado i' apresenta um comportamento igual ao do estado i . A única diferença existente entre eles é que quando um nó é chaveado para o estado i , ele dispara um contador de tempo. Quando um nó é chaveado para o estado i' , ele verifica se existe algum evento para ele. Em termos de consumo de energia, os estado i e i' são idênticos, contudo, o comportamento deles é distinto.

O diagrama da figura 2.7 apresenta os comandos efetuados ao longo do caminho (transições) entre os estados. Isso significa que quando um nó altera seu estado corrente, ele realiza alguns testes e ações antes de alcançar o novo estado. Os testes são: “rotear” – verifica se existe alguma mensagem para ser roteada pelo nó; “dormir” – determina se o nó deve dormir ou não; “existe evento” – determina se existe um novo evento de sensoriamento; “ligar radio” – determina se o rádio deve ser ligado ou não; e “receber” – determina se o rádio deve ser ativado para receber ou transmitir. A saída de cada teste

depende de um parâmetro de probabilidade associado a cada um deles. Por sua vez, cada ação de “Tempo” tem como objetivo iniciar um contador de tempo. Finalmente, cada transmissão tenta capturar o comportamento do nó sensor em termos de seu consumo de energia.

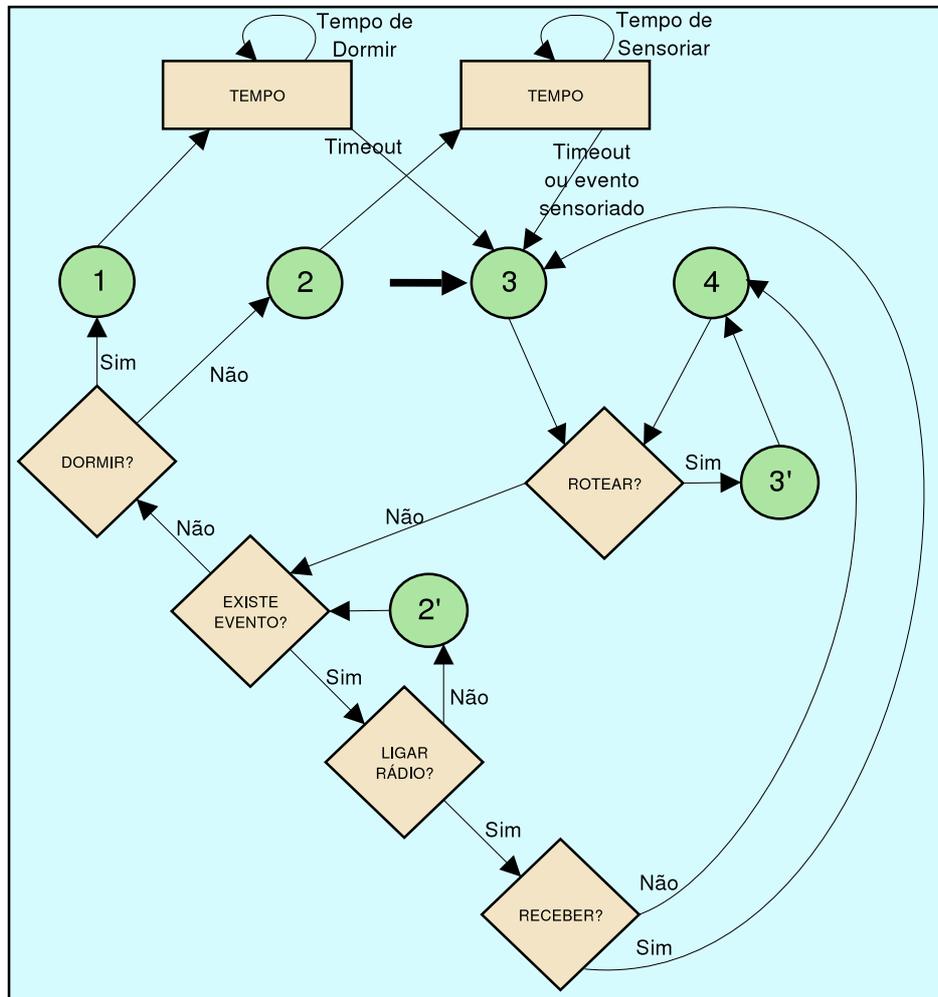


Figura 2.7: Diagrama do *State-based Energy Dissipation Model*.

2.5 Conclusão

Na área de redes de computadores, o roteamento é um tópico bastante pesquisado. Na computação móvel, devido à mobilidade dos dispositivos e/ou as restrições existentes, esse problema ganha uma atenção especial. Nas redes móveis *ad hoc*, o mesmo é considerado como o problema principal. Em redes de sensores sem fio, geralmente os dispositivos computacionais são estáticos, contudo, a rede também apresenta topologia dinâmica em função de nós que são chaveados/rechaveados para *sleeping mode*, nós que

morrem e nós que são adicionados na rede. Logo, o problema do roteamento continua merecendo uma atenção especial. Além disso, nas RSSFs, devido às restrições existentes (principalmente a de energia), o problema de roteamento torna-se mais desafiador. Dentre os principais protocolos para o roteamentos em RSSFs, destaca-se o *Directed Diffusion* e o LEACH.

É importante destacar que a maioria dos protocolos de roteamento utilizam em alguma etapa a difusão de dados para encontrar a rota desejada. Assim, o estudo e projeto de novas técnicas eficientes para a difusão de dados é de grande importância. Como pode ser observado nos trabalhos relacionados, vários protocolos de difusão em RSSF utilizam a técnica de *forwarding delay* para otimizar alguma métrica da rede, como o número de pacotes transmitidos ou número de nós cobertos pela difusão.

Uma área de estudo importante nas redes de sensores sem fio é o mapa de energia. Utilizando o mapa, é possível determinar se alguma parte da rede pode sofrer falhas devido à falta de energia (Zhao et al., 2002). Várias aplicações para RSSFs podem utilizar a informação fornecida pelo mapa, por exemplo, os algoritmos de difusão de dados. No estudo de RSSFs, mais especificamente no processo de avaliação das mesmas, um dos itens mais importantes é o modelo de dissipação de energia dos nós sensores. Um modelo interessante para isso é o *State-based Energy Dissipation Model* (SEDM).

A partir da importância do roteamento em RSSFs, da proposta de utilização da técnica de *forward delay* e das vantagens advindas do mapa de energia, o presente trabalho propõe um protocolo para difusão de dados que utiliza a técnica de *forward delay* juntamente com o mapa de energia para difundir pacotes na rede. O protocolo proposto será comparado com o *gossiping*, com o DDB2 e TEDD que utilizam informações de energia e a técnica de *forwarding delay*. A comparação não será feita com o BPS porque o mesmo não utiliza informações de energia na execução da difusão.

Capítulo 3

Delay and Energy Based Broadcasting (DEBB)

3.1 Introdução

Esta seção apresenta o DEBB (*Delay and Energy Based Broadcasting*), um protocolo de difusão de dados que utiliza o mapa de energia juntamente com o conceito de *forwarding delay* para disseminar informações de forma eficiente em redes de sensores sem fio. O DEBB reduz os custos de energia e processamento na difusão quando comparado com protocolos da literatura, e possui a capacidade de contornar obstáculos, como as regiões de baixa energia. Essa capacidade é escalável e independe do número de obstáculos. O DEBB tem como objetivos reduzir o número de transmissões, aumentar a quantidade de nós cobertos e eliminar o fluxo de dados dentro das regiões de baixa energia. Como a latência é o principal problema dos algoritmos que utilizam *forwarding delay*, é apresentada uma versão probabilística do DEBB, denominada DEBB-P, que deve ser utilizada em aplicações sensíveis ao atraso.

O DEBB e DEBB-P possuem dois módulos em seu funcionamento. O primeiro módulo (igual para os dois protocolos) é o de energia de corte e ela tem como objetivo evitar o fluxo de dados dentro de regiões de baixa energia. O segundo módulo do DEBB é o de *forwarding delay* e ela insere atrasos nas retransmissões do pacote de difusão com o objetivo de reduzir a quantidade de retransmissões e aumentar a cobertura da rede¹. O segundo módulo do DEBB-P é a de redução da latência. Essa função utiliza uma técnica probabilística na retransmissão do pacote de difusão com o objetivo de reduzir a latência da difusão mantendo as mesmas características de retransmissões e número de nós que receberam o pacote de difusão do DEBB. A seção 3.2 apresenta como a energia de corte é calculada. As seções 3.3 e 3.4 apresentam o funcionamento

¹Quantidade de nós que recebeu o pacote difundido.

básico do DEBB e do DEBB-P, respectivamente. A seção 3.5 apresenta a conclusão do presente capítulo.

3.2 Energia de Corte

A energia de corte é um limiar para identificar se um determinado nó encontra-se em uma região de baixa energia e, neste caso, o nó é proibido de retransmitir o pacote. Isso é feito para que os nós com baixa reserva de energia possam consumir sua energia com o processo de sensoriamento, e não de roteamento².

Para calcular o valor da energia de corte, é necessário definir o que é uma região de baixa energia. Este trabalho considera que uma região de baixa energia ocorre quando a média de energia dos nós que cobrem aquela região é menor que a média de energia da rede menos o desvio padrão da energia da rede. Para obter a energia de corte, o *sink* calcula a energia média da rede e o seu desvio padrão utilizando o mapa de energia. Em seguida, para cada nó sensor, o *sink* calcula um coeficiente de energia. Esse valor corresponde a soma da energia de um nó com a energia de todos os seus vizinhos dividido por $n + 1$, onde n é o número de vizinhos do nó e o “mais um” corresponde ao próprio nó. Dados a energia média, o desvio padrão e o coeficiente de energia de cada nó, a energia de corte será igual ao maior coeficiente de energia que seja menor que a energia média menos o desvio padrão. Caso não exista um coeficiente de energia que seja menor que a energia média menos o desvio padrão, a rede não possui regiões de baixa energia, logo, o valor da energia de corte é zero.

A figura 3.1 ilustra a escolha da energia de corte. Dados os coeficientes de energia de todos os nós da rede, a média de energia da rede e seu respectivo desvio padrão, tem-se que 4 nós estão abaixo da média menos o desvio padrão, caracterizando nós com baixa reserva de energia. Entre esses 4 nós, a energia de corte se caracteriza pelo nó que possui a maior reserva de energia, como ilustrado na figura.

Um benefício na utilização dessa técnica, é que ela consegue contornar de forma sensível as regiões de baixa energia da rede. Por exemplo, se um nó com alta reserva de energia está sobreposto (localizações geográficas) a algum nó com baixa reserva de energia, apenas o nó com alta reserva de energia participará do roteamento do pacote. Em outras palavras, a técnica proposta consegue determinar quais nós da rede podem participar do roteamento.

²Os termos roteamento e difusão são utilizados indistintamente no decorrer do texto.

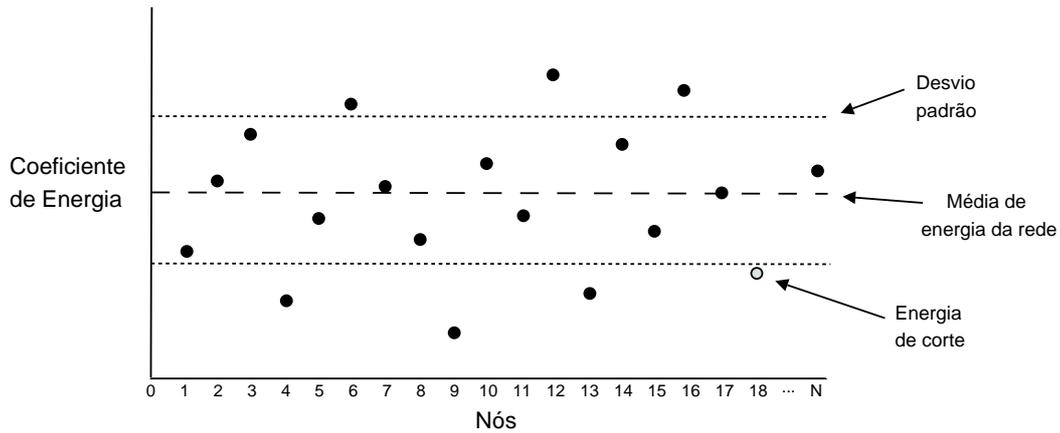


Figura 3.1: Escolha da energia de corte.

3.3 Funcionamento Básico do DEBB

O algoritmo de difusão DEBB inicia quando o nó *sink* deseja disseminar um dado para todos os nós da rede. Antes de fazer essa difusão, o *sink* calcula três parâmetros obtidos a partir do mapa de energia: *energia de corte*, *energia máxima* e *energia mínima* da rede. Esses valores de energia são inseridos nos pacotes a serem difundidos para todos os nós da rede.

O objetivo da energia de corte e a forma com que este valor é calculado foram mostrados na seção 3.2. Os outros dois parâmetros obtidos na avaliação do mapa de energia são a energia máxima e energia mínima que serão utilizadas pelos nós intermediários na definição de seus respectivos atrasos. Durante o processo de avaliação do mapa de energia, esses valores são trivialmente calculados e corresponde à energia do nó sensor com o maior valor de energia e a energia do nó sensor com o menor valor de energia. A utilização desses valores é importante para redução da latência, como será visto posteriormente.

No processo de difusão do DEBB, os valores da energia de corte, energia máxima e mínima e a coordenada geográfica do último nó que propagou o pacote são inseridos nos pacotes a serem transmitidos. Quando um nó intermediário recebe um pacote, ele utiliza uma abordagem do tipo *receiver-based* para decidir se deve retransmití-lo. Nesse caso, o nó toma a decisão de transmissão baseando-se nas informações contidas no pacote e em sua própria coordenada geográfica. A figura 3.2 ilustra o funcionamento do DEBB. Ao receber um pacote, o nó verifica se sua energia é menor que a energia de corte (figura 3.2, ponto A). Em caso positivo, o pacote é descartado (figura 3.2, ponto B) e em caso negativo, o nó calcula sua distância até o nó anterior (figura 3.2, ponto C) e, em seguida, ele calcula seu tempo de espera (figura 3.2, ponto D). O tempo de espera é inversamente proporcional à distância entre o nó corrente e o nó anterior

e inversamente proporcional à energia do nó. A equação (3.1) apresenta a fórmula utilizada para o cálculo do tempo de espera.

$$TempoEspera = TimeStep * \left(\frac{\frac{raio-dist}{DStep} + \frac{EMax - ((MinhaEnergia - EMin) / (EMax - EMin))}{EStep}}{2} \right) \quad (3.1)$$

onde o menor intervalo de tempo necessário para que um nó realize o processamento de um pacote e o transmita é denominado *TimeStep*, *raio* é o raio de comunicação do nó, *dist* é a distância entre o nó e o nó anterior, *EMax* e *EMin* são os parâmetros energia máxima e mínima contidos no pacote e *MinhaEnergia* corresponde a energia corrente do nó. Finalmente, os parâmetros *DStep*, *EStep* correspondem aos valores para normalizar as duas grandezas (distância e energia) utilizadas na fórmula. Numericamente, a normalização é feita pois, a distância entre dois nós pode ser no máximo o raio de comunicação (ex: 5 m) e a energia do nó em um determinado instante pode ser de 30 J. Além disso, a energia do nó varia com o tempo de operação da rede, o que não acontece com o raio de comunicação.

Após aguardar um intervalo de tempo igual ao seu tempo de espera (figura 3.2, ponto E), o nó verifica se algum vizinho já retransmitiu o pacote (figura 3.2, ponto F). Em caso positivo, o pacote é descartado (figura 3.2, ponto B). Caso contrário, o nó insere sua própria coordenada geográfica no pacote (figura 3.2, ponto G) e o propaga (figura 3.2, ponto H).

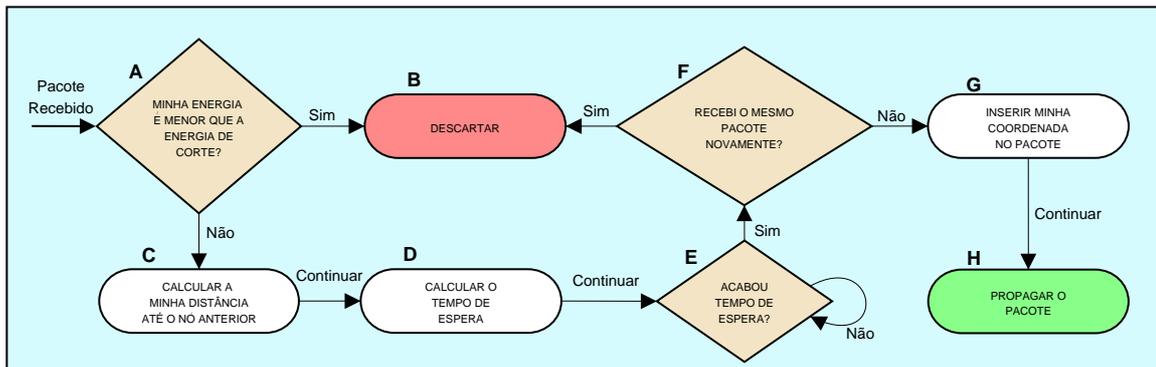


Figura 3.2: Funcionamento básico do DEBB.

É importante destacar que o DEBB poderia considerar a energia inicial dos nós em vez da energia máxima. Nesse caso, os menores tempos de atraso seriam reservados para os valores de energia entre a energia inicial e a energia máxima. Como não existe nó cujo valor da energia é maior que a energia máxima da rede, o DEBB estaria inserindo atrasos desnecessários e, conseqüentemente, sua latência seria maior. Além disso, nota-se que à medida que a energia dos nós acaba, a diferença entre os valores

da energia inicial e da energia máxima da rede aumenta. Logo, se a energia inicial fosse considerada, a latência iria aumentar durante o tempo de vida da rede, já que intervalos de tempos seriam inseridos desnecessariamente no cálculo do atraso.

O pseudo-código do algoritmo DEBB é apresentado no algoritmo 1. Considera-se “pacote” como o pacote de difusão recebido pelo nó. Além disso, é utilizado as seguintes variáveis de controle e auxiliares:

- **recebido**: corresponde se o pacote já foi recebido pelo nó;
- **pacote.id**: corresponde ao id do pacote;
- **pacote.x** e **pacote.y**: posição geográfica do nó que transmitiu o pacote;
- **pacote.energiaCorte**: corresponde a energia de corte calculada pelo sink;
- **x** e **y**: posição geográfica do nó que recebeu o pacote;
- **pacote.energiaMaximaRede** e **pacote.energiaMinimaRede**: correspondem ao nó com maior e ao nó com menor energia da rede e esses valores são calculados pelo sink;
- **distancia**: corresponde a distância euclidiana entre o nó que enviou e o nó que recebeu o pacote;
- **atrasoDistancia**: corresponde ao atraso do pacote que é inversamente proporcional a distância entre os nós;
- **atrasoEnergia**: corresponde ao atraso do pacote que é inversamente proporcional a energia do nó;
- **aux**: variável auxiliar para normalizar os valores de distância e energia;
- **minhaEnergia**: corresponde a energia corrente do nó sensor;
- **RAIO**: constante contendo o raio de comunicação do nó;
- **TIME_STEP**: tempo necessário para que um nó realize o processamento de um pacote e efetue a sua transmissão;

A seguir é feita uma descrição e análise do algoritmo 1:

- Linhas 2–6: inicialização das variáveis utilizadas no algoritmo.
- Linha 7: a função *pacoteRecebido(...)* recebe como parâmetro o id do pacote e verifica se esse pacote já foi recebido antes. Essa busca é feita em $O(\log n)$.

```

Entrada: pacote
1 inicio
2   recebido  $\leftarrow$  falso;
3   distancia  $\leftarrow$   $\emptyset$ ;
4   atrasoDistancia  $\leftarrow$   $\emptyset$ ;
5   atrasoEnergia  $\leftarrow$   $\emptyset$ ;
6   aux  $\leftarrow$   $\emptyset$ ;
7   recebido  $\leftarrow$  pacoteRecebido(pacote.id);
8   se recebido = verdadeiro ou minhaEnergia < pacote.energiaCorte então
9     | retorna ;
10  senão
11    | addPacoteRecebido(pacote.id);
12    | distancia  $\leftarrow$  distanciaEntreNos(x, y, pacote.x, pacote.y);
13    | atrasoDistancia  $\leftarrow$  RAI0 - distancia;
14    | se minhaEnergia = pacote.energiaMaximaRede então
15    |   | aux  $\leftarrow$  1;
16    |   senão
17    |     | aux  $\leftarrow$   $\frac{\text{minhaEnergia} - \text{pacote.energiaMinimaRede}}{\text{pacote.energiaMaximaRede} - \text{pacote.energiaMinimaRede}}$ ;
18    |     | atrasoEnergia  $\leftarrow$  RAI0 - ( aux * RAI0 );
19    |     | atraso  $\leftarrow$   $\lfloor \text{TIME\_STEP} * \left( \frac{\text{atrasoDistancia} + \text{atrasoEnergia}}{2} \right) \rfloor$  ;
20    |     | espera (atraso);
21    |     | recebido  $\leftarrow$  pacoteRecebido(pacote.id);
22    |     | se recebido = verdadeiro então
23    |     |   | retorna ;
24    |     |   senão
25    |     |     | pacote.x  $\leftarrow$  x;
26    |     |     | pacote.y  $\leftarrow$  y;
27    |     |     | send (pacote);
28 fim

```

Algoritmo 1: Algoritmo do DEBB.

- Linhas 8–9: verifica se o pacote já foi recebido ou se a energia que o nó possui é menor que a energia de corte. Se sim, o pacote é descartado. Essa verificação é feita em $O(1)$.
- Linhas 10–27: Se o pacote não foi recebido, o nó irá tentar executar a sua retransmissão.
- Linha 11: se o pacote não foi recebido, é feita a manutenção da lista de pacotes recebidos. A inserção nessa lista é feita de forma ordenada pela id do pacote. Essa manutenção é feita em $O(\log n)$.
- Linhas 12–13: cálculo da distância entre o nó que enviou e o nó que recebeu o

pacote. O atraso correspondente à distância dos nós é dado pela diferença entre o raio de comunicação do nó pela distância calculada. O atraso é inversamente proporcional a distância entre os nós. O cálculo do atraso é feito em $O(1)$.

- Linhas 14–17: se a energia corrente do nó é o valor da energia máxima da rede, a variável auxiliar recebe 1. Se for menor, essa variável recebe o resultado da divisão da energia corrente do nó menos a menor energia que um nó da rede possui pela energia máxima da rede. Isso é feito para obter a proporção de energia que o nó possui com relação a energia máxima e mínima da rede. Essa proporção varia de 0 a 1. A verificação da energia do nó é feita em $O(1)$.
- Linha 18: a proporção calculada no item anterior é utilizada para fazer uma normalização de valores. Ou seja, o mesmo peso que a distância possui no atraso do pacote (Linha 13) a energia deve possuir. O resultado contido em *atrasoDistancia* da linha 13 é no máximo de **RAIO** pois, a distancia entre dois nós pode ser de $0m$ e o resultado contido no *atrasoEnergia* da linha 18 também é no máximo **RAIO**.
- Linha 19: o atraso total do pacote é dado pela divisão entre a soma dos atrasos da distância e energia por 2 e multiplicado por *TIME_STEP*, que é o menor intervalo de tempo para diferenciar dois atrasos. O piso é utilizado para que os nós possuem atrasos em intervalos de *TIME_STEP*. Ou seja, os atrasos resultantes da fórmula são por exemplo 15 *TIME_STEPS* ou 70 *TIME_STEPS* e não 7.5 *TIME_STEPS*.
- Linha 20–27: depois de calculado, o nó espera o tempo necessário e verifica se nesse período ele recebeu uma cópia do mesmo pacote. Se sim, o nó cancela a sua transmissão. Caso contrário, ele atualiza a sua posição geográfica no pacote e efetua a sua transmissão. A complexidade dessa verificação é $O(1)$ e a complexidade de verificar se o pacote foi recebido é $O(\log n)$

Sendo assim, a complexidade de tempo total no pior caso é $O(\log n) + O(1) + O(\log n) + O(1) + O(1) + O(\log n) = O(\log n)$. A complexidade de trocas de mensagens no pior caso é $O(n)$, onde n é a quantidade de nós na rede. Nesse caso, o pior caso é quanto o nó possui apenas 1 vizinho, fazendo com que todos os nós participem da difusão.

3.4 DEBB Probabilístico

Como todos os protocolos baseados em *forwarding delay*, o DEBB apresenta alta latência devido ao tempo de espera associado a cada retransmissão. Portanto, soluções baseadas em *forwarding delay*, como o DEBB, não são indicadas para sistemas dependentes do tempo. Para atender às aplicações sensíveis ao atraso, uma variação probabilística do DEBB, denominada DEBB-P é proposta nesta seção. No DEBB-P, a probabilidade de transmissão é calculada ao invés do atraso. O valor dessa probabilidade será recalculado a cada *TIME_STEP* até que o nó ou algum de seus vizinhos tenha transmitido o pacote ou a quantidade de tentativas tenha alcançado um limite proporcional a sua probabilidade de transmissão, como será visto posteriormente.

A figura 3.3 ilustra o funcionamento do DEBB-P. Quando um nó recebe um pacote, ele verifica se sua energia é menor que a energia de corte (figura 3.3, ponto *A*). Em caso positivo, o pacote é descartado (figura 3.3, ponto *B*). Caso contrário, o nó calcula sua distância até o nó anterior e sua probabilidade (figura 3.3, pontos *C* e *D*). Conforme mostrado na equação (3.2), a probabilidade de transmissão é proporcional à distância:

$$ProbabilidadeTransmissao = PROB_STEP * distancia \quad (3.2)$$

onde, *PROB_STEP* é o coeficiente de crescimento da função no cálculo das probabilidades e *distancia* é a distância euclidiana entre o nó que enviou e recebeu o pacote. No passo seguinte, o nó sorteia um número aleatório, segundo uma distribuição uniforme (figura 3.3, ponto *E*). Se esse valor for menor ou igual a probabilidade (figura 3.3, ponto *F*), o nó insere sua própria coordenada geográfica no pacote e o propaga (figura 3.3, pontos *I* e *J*). Quando esse número for maior que a probabilidade, o nó aguarda um *TIME_STEP* (figura 3.3, ponto *G*). Após o *TIME_STEP*, o nó verifica se algum de seus vizinhos retransmitiu o pacote (figura 3.3, ponto *H*). Em caso positivo, o nó descarta o pacote (figura 3.3, ponto *B*). Caso contrário, o processo é reinicializado (figura 3.3, ponto *E*).

Uma consideração importante sobre o DEBB-P é que a probabilidade de transmissão não precisa levar em consideração os valores de energia. O comportamento não determinístico do sorteio de números aleatórios se encarrega de não utilizar sistematicamente os mesmos nós no roteamento. Além disso, o fluxo de dados dentro das regiões de baixa energia sempre será eliminado independente da probabilidade de transmissão através da utilização da energia de corte. Outro ponto importante do DEBB-P é o compromisso entre a latência e o número de transmissões. Quanto maior o valor da probabilidade, menor a latência e maior o número de transmissões. Destaca-se a possibilidade de se calibrar a probabilidade para que o protocolo obtenha bons desempenhos

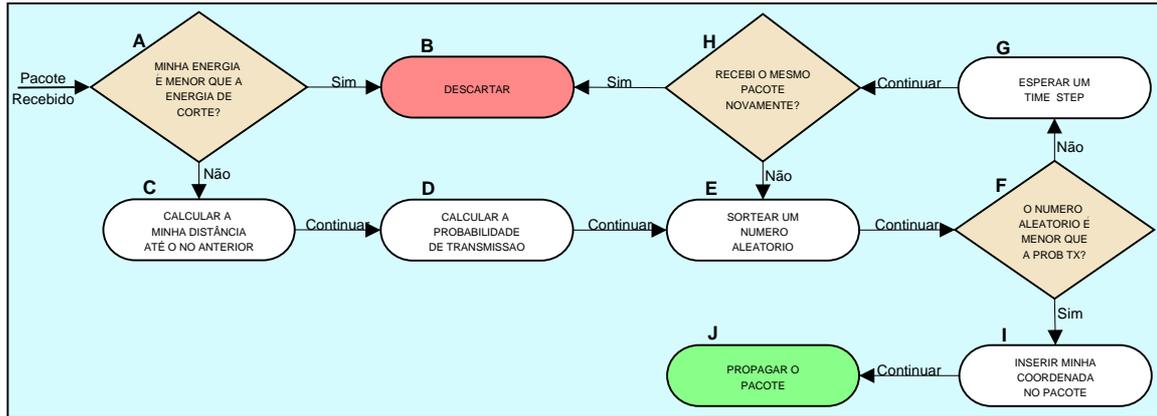


Figura 3.3: Funcionamento básico do DEBB-P.

de latência e número de transmissões.

O pseudo-código do algoritmo DEBB-P é apresentado no algoritmo 2. Considera-se “pacote” como o pacote de difusão recebido pelo nó. Além disso, é utilizado as seguintes variáveis de controle e auxiliares:

- **recebido**: corresponde se o pacote já foi recebido pelo nó;
- **pacote.id**: corresponde ao id do pacote;
- **pacote.x** e **pacote.y**: posição geográfica do nó que transmitiu o pacote;
- **pacote.energiaCorte**: corresponde a energia de corte calculada pelo sink;
- **x** e **y**: posição geográfica do nó que recebeu o pacote;
- **distancia**: corresponde a distância euclidiana entre o nó que enviou e o nó que recebeu o pacote;
- **aux**: variável auxiliar para o cálculo da probabilidade de transmissão;
- **p**: variável para conter um número aleatório entre 0 e 1;
- **probTransmissão**: variável contendo a probabilidade de transmissão do nó;
- **contador**: utilizado para verificar quantas tentativas de transmissão o nó realizou;
- **RAIO**: constante contendo o raio de comunicação do nó; **PROB_STEP**: coeficiente de crescimento da função no cálculo das probabilidades;
- **TIME_STEP**: tempo necessário para que um nó realize o processamento de um pacote e o transmita;

```

Entrada: pacote
1 inicio
2   recebido  $\leftarrow \emptyset$ ;
3   atraso  $\leftarrow \emptyset$ ;
4   distancia  $\leftarrow \emptyset$ ;
5   probTransmissao  $\leftarrow \emptyset$ ;
6   p  $\leftarrow \emptyset$ ;
7   recebido  $\leftarrow$  pacoteRecebido(pacote.id);
8   se recebido = verdadeiro ou minhaEnergia < pacote.energiaCorte ou (x =
   pacote.x e y = pacote.y) então
9     | retorna ;
10  senão
11    | addPacoteRecebido(pacote.id);
12    | distancia  $\leftarrow$  distancia(x, y, pacote.x, pacote.y);
13    | probTransmissao  $\leftarrow$  PROB_STEP*distancia;
14    | para contador  $\leftarrow \emptyset$  até contador  $\leq \frac{1}{probTransmissao}$  faça
15      |   p  $\leftarrow$  uniforme (0,1);
16      |   se p  $\leq$  probTransmissao então
17        |   | pacote.x  $\leftarrow$  x;
18        |   | pacote.y  $\leftarrow$  y;
19        |   | send (pacote);
20        |   | retorna ;
21      |   senão
22        |   | atraso  $\leftarrow$  TIME_STEP;
23        |   | espera (atraso);
24        |   | recebido  $\leftarrow$  pacoteRecebido (pacote.id);
25        |   | se recebido = true então
26          |   | | retorna ;
27 fim

```

Algoritmo 2: Algoritmo do DEBB-P.

A seguir é feita uma descrição e análise do algoritmo 2:

- Linhas 2–6: inicialização das variáveis de controle e auxiliares do algoritmo.
- Linha 7: a função *pacoteRecebido(...)* recebe como parâmetro o id do pacote e verifica se esse pacote já foi recebido. Essa busca é feita em $O(\log n)$.
- Linhas 8–9: verifica se o pacote já foi recebido ou se a energia que o nó possui é menor que a energia de corte. Em caso afirmativo, o pacote é descartado. Essa verificação é feita em $O(1)$. O pacote também é descartado se a posição geográfica do nó que enviou o pacote é a mesma do que recebeu. Isso é feito para o nó não entrar em *loop* infinito no laço da linha 14.

- Linhas 10–26: Se o pacote não foi recebido, o nó irá tentar executar a sua retransmissão.
- Linha 11: se o pacote não foi recebido, é feita a manutenção da lista de pacotes recebidos. A inserção nessa lista é feita de forma ordenada pela id do pacote. Essa manutenção é feita em $O(\log n)$.
- Linhas 12 e 13: cálculo da distância entre o nó que enviou e o nó que recebeu o pacote. A probabilidade de transmissão do nó é proporcional a distância calculada.
- Linhas 14–26: laço que contabiliza a quantidade de vezes que o nó tentou transmitir o pacote. A condição de parada do laço é proporcional a probabilidade de envio calculada no item anterior. Em média, depois de $\frac{1}{probTransmissao}$ vezes que o nó sortear um número aleatório, esse número será menor que $probTransmissao$. Então, esse laço é executado no máximo $\frac{1}{probTransmissao}$. Assim, seu custo é $O\left(\frac{1}{probTransmissao}\right)$. Como $\frac{1}{probTransmissao}$ é uma constante, a complexidade de tempo desse laço é $O(1)$. Além disso, o nó sairá desse laço se o mesmo executar a transmissão do pacote.
- 15–22: é sorteado um número aleatório e verifica se esse número é menor que a $probTransmissao$. Em caso afirmativo, o nó atualiza o pacote com as suas coordenadas geográficas e efetua a sua transmissão. Caso contrário, o tempo de atraso para uma nova tentativa de transmissão é de apenas um `TIME_STEP`.
- 23–26: o nó espera o tempo necessário para uma nova tentativa de retransmissão do pacote. Depois dessa espera, o nó verifica se o pacote recebido é uma cópia do pacote a ser transmitido por ele. Se sim, o nó descarta o pacote. Caso contrário, a execução do algoritmo volta para a linha 14 e esse processo se reinicia. A complexidade para verificar se o pacote foi recebido é $O(\log n)$

Sendo assim, a complexidade de tempo total no pior caso é $O(\log n) + O(\log n) + O(1) + O(1) + O(1) + O(\log n) = O(\log n)$. A complexidade de trocas de mensagens no pior caso é $O(n)$, onde n é a quantidade de nós na rede. Nesse caso, o pior caso é quanto todos os nós efetuam, na primeira tentativa, a transmissão do pacote e todos os nós da rede são alcançáveis pela difusão.

3.5 Conclusão

O *Delay and Energy Based Broadcasting (DEBB)* é um protocolo para difusão de dados baseado no mapa de energia e na técnica de *forwarding delay*. O DEBB e DEBB-P são

compostos por dois módulos: energia de corte (DEBB e DEBB-P), *forwarding delay* (DEBB) e probabilístico (DEBB-P).

A técnica de energia de corte tem como objetivo evitar o fluxo de dados dentro de regiões de baixa energia. Além disso, a técnica de energia de corte consegue determinar de maneira exata, quais nós podem participar do roteamento, independente da sua localização geográfica. A energia de corte é facilmente calculada e requer baixo custo computacional pelo *sink*. Assim, em cenários onde o mapa de energia não é criado no *sink* e sim nos nós da rede (mapas localizados), a técnica de energia de corte pode ser utilizada para difusões locais, onde deseja-se evitar o fluxos de dados dentro de pequenas regiões de baixa energia.

Finalmente, é importante destacar o baixo custo computacional do DEBB, como mostrado nos algoritmos 1 e 2. Além disso, a utilização da energia de corte preserva os nós com baixa energia. É importante observar que tanto o DEBB quanto o DEBB-P não precisam manter informações de estado em cada nó, tais como grafo de dependência ou tabela de vizinhos. Os nós utilizam somente informações retiradas do pacote de difusão e informações locais para tomar a decisão de transmitir ou não o pacote, o que tornam os protocolos robustos e eficazes no processo de difusão de dados na rede.

Capítulo 4

Resultados de Simulação

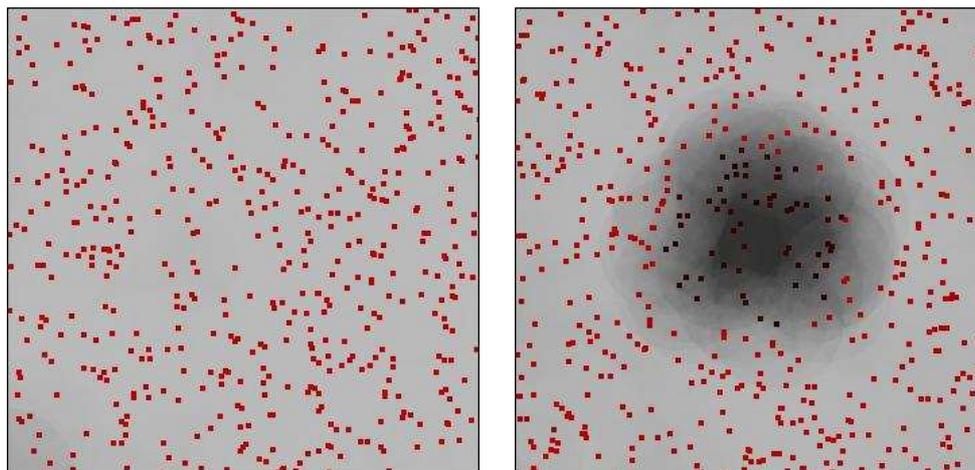
4.1 Introdução

Nesta seção, o DEBB é simulado e avaliado em três cenários de difusão de dados para RSSFs. Em todos eles, existe apenas um único *sink*, sem restrição de energia, localizado no canto inferior esquerdo da rede e que realiza difusões de dados periodicamente. O primeiro cenário é caracterizado pela distribuição homogênea da energia inicial dos nós sensores, como ilustrado na figura 4.1 (a). A figura 4.1 (b) mostra o segundo cenário de simulação no qual existe uma região de baixa energia no centro da rede. No último cenário, existem quatro regiões de baixa energia, como ilustrado na figura 4.1 (c). É importante destacar que as regiões de baixa energia são geradas de maneira aleatória na rede, ou seja, a posição (x,y) do centro da região de baixa energia varia em diferentes simulações.

Com o objetivo de encontrar a melhor probabilidade de transmissão, é feita uma avaliação da influência da probabilidade de transmissão no DEBB-P. Essa avaliação é feita para encontrar o melhor compromisso entre quantidade de pacotes transmitidos, número de nós que receberam o pacote de difusão (cobertura), consumo de energia, número de nós mortos e latência.

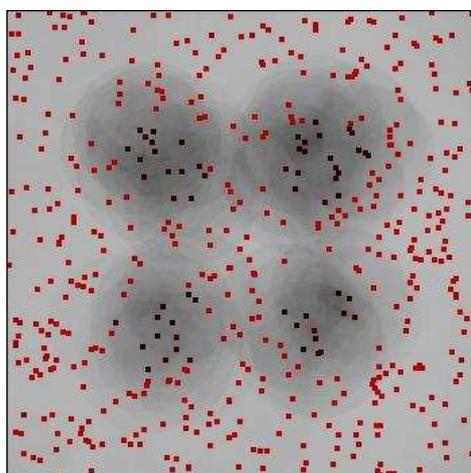
Em todas as simulações, os nós sensores são estáticos, entretanto, a topologia é dinâmica porque periodicamente os sensores dormem para economizar energia. Segundo Hill et al. (2000), uma RSSF deve ter como filosofia realizar seu trabalho o mais rápido possível e em seguida adormecer. A melhor forma de economizar energia em RSSFs é desligar as partes dos sensores que não estiverem sendo utilizadas. Esse comportamento é modelado pelo *State-based Energy Dissipation Model* (SEDM) (Mini et al., 2005). Neste trabalho, o SEDM é utilizado levando-se em consideração o consumo de energia do Mica2 (Mica2, 2004).

Em todos os cenários analisados, a rede utilizada é composta por 500 nós estáticos,



(a) Energia homogênea.

(b) Uma região de baixa energia.



(c) Quatro regiões de baixa energia.

Figura 4.1: Cenários de simulação.

homogêneos, dispostos aleatoriamente em uma área de $35 \times 35 m^2$ e cuja recarga da bateria é considerada impossível. Cada nó sensor apresenta em média 27 vizinhos, porém, durante a simulação esse valor é reduzido porque os nós adormecem para economizar energia. Conseqüentemente, durante a difusão de dados, nem todos os nós podem ser alcançados uma vez que alguns podem estar dormindo ou apenas sensoriando (rádio desligado). A energia inicial dos nós é igual a $40 J$ e o raio de comunicação é $5 m$. Durante cada simulação, o nó *sink* realiza a difusão de 200 mensagens que são disseminadas uniformemente durante o tempo de simulação. O tempo total de simulação é igual a 1000 segundos. Os nós dentro das regiões de baixa energia possuem $20 J$ de energia inicial.

Neste trabalho, o DEBB e DEBB-P foram implementados no simulador ns-2.26 (*Network Simulator 2.26*) (ns2, 2002) e o mesmo é comparado com o TEDD, DDB2 e o *gossiping* com probabilidade igual a 0.4. O valor dessa probabilidade foi definido

com base nos resultados apresentados em do Val Machado et al. (2005) e Goussevskaia et al. (2005b). O TEDD foi utilizado com a política de um fluxo. O DDB2 foi utilizado com um *threshold* de 60% da área que seria coberta pela transmissão do nó. Foi avaliado a utilização do threshold de 50% e 70%, mas esses não mostraram-se eficientes com relação as métricas avaliadas no roteamento. Para obter o mapa de energia, o DEBB e o TEDD utilizam a abordagem baseada em predição proposta por Mini et al. (2005). O custo da obtenção do mapa de energia não é considerado neste trabalho uma vez que o mesmo deve ser dividido entre as diversas aplicações da rede. Além disso, a energia máxima corrente da rede é utilizada pelo DDB2, e não a energia inicial dos nós, como proposto por Heissenbüttel et al. (2006). Isso é feito para que atrasos desnecessários não sejam inseridos em seu cálculo. Todos os valores apresentados nos resultados correspondem à média aritmética de n simulações, onde n é definido a partir do valor de confiança desejado na simulação (Jain, 1991). Em todos os experimentos, utilizou-se 95% de intervalo de confiança assintótico. Além disso, o Teste-T (Jain, 1991) foi utilizado para afirmar que um protocolo é superior, inferior ou igual a outro.

A tabela 4.1 apresenta os parâmetros utilizados em todas as simulações. A quantidade de simulações necessárias para obter-se um intervalo de confiança é $\left(\frac{100ts}{r\bar{x}}\right)^2$. Assim, foram feitas 10 simulações iniciais¹ e calculado a média (\bar{x}) e o desvio padrão (s) dessas simulações. O erro aceitável (r) é de 5%. Esses valores são parâmetros de entrada para obter-se a quantidade de simulações necessárias para se ter um intervalo de confiança de 95%.

O restante desta seção está organizado da seguinte forma. A seção 4.2 avalia a influência da variação da probabilidade de transmissão do DEBB-P. As seções 4.3, 4.4 e 4.5 avaliam o desempenho do DEBB, TEDD, DDB2 e *gossiping* nos três cenários de simulação analisadas neste trabalho.

4.2 Avaliação da Probabilidade de Transmissão do DEBB-P

Como descrito na seção 3.4, no DEBB-P, é possível calibrar a probabilidade de transmissão dos nós. O objetivo dessa seção é avaliar o desempenho do DEBB-P com diferentes probabilidades de transmissão. Essa avaliação é feita no cenário com energia homogênea. A probabilidade de transmissão do nó varia de 0 a uma probabilidade máxima, dependendo da sua distância em relação ao nó que enviou o pacote. Isso é feito para encontrar a probabilidade de transmissão que apresenta o melhor compro-

¹De acordo com Jain (1991), a quantidade de simulações iniciais não afeta o cálculo para encontrar a quantidade de simulações total para ter-se um intervalo de confiança de 95%.

Parâmetros	
Simulador	NS-2.26
Topologia da rede	plana
Nós (<i>hardware</i>)	Homogêneos
Dimensões da rede	35x35 <i>m</i>
Números de nós	500
Raio de comunicação	5 <i>m</i>
Energia Inicial	40 <i>J</i>
SEDM estado 1	0.00003 <i>W</i>
SEDM estado 2	0.0249 <i>W</i>
SEDM estado 3	0.0489 <i>W</i>
SEDM estado 4	0.1011 <i>W</i>
Tempo de Simulação	1000 <i>s</i>
Análise estatística	Teste-T
Intevalo de confiança	95% (distribuição <i>t</i>)
Quantidade de simulações	$n = \left(\frac{100ts}{r\bar{x}}\right)^2$
Protocolos comparados	DEBB, DEBB-P, TEDD, DDB2 e gossiping ($p = 0.4$)

Tabela 4.1: Parâmetros utilizados nas simulações.

misso entre vários parâmetros de rede avaliados, como a quantidade de transmissões, cobertura, nós mortos e latência.

A tabela 4.2 apresenta os resultados de simulação do DEBB-P variando a probabilidade máxima de transmissão. Observa-se que a medida que essa probabilidade aumenta, a quantidade de energia consumida na difusão cresce proporcionalmente com a quantidade de pacotes transmitidos. Em todas as probabilidades, o algoritmo se mantém constante com relação a cobertura dos nós da rede. Isso acontece pois, a probabilidade de transmissão não afeta o comportamento do algoritmo, e sim, na quantidade de vezes que um nó tenta executar a transmissão do pacote. Porém, a quantidade de pacotes transmitidos e conseqüentemente a energia consumida na difusão aumenta com o aumento da probabilidade de transmissão. Pode-se observar que energia consumida em todos os casos é diferente (Teste-T) pois, o intervalo de confiança das energias consumidas não se sobrepõem. Entretanto, observa-se que o DEBB-P com as probabilidades 0.03, 0.05 e 0.07 possuem a mesma quantidade de pacotes transmitidos (Teste-T) e diferente do DEBB-P com probabilidade 0.09, que é superior (Teste-T). Observa-se também que, a quantidade de nós mortos é desprezível para todas as probabilidades. Assim, pode-se verificar o que foi falado na seção 3.4, onde, o DEBB-P não precisa utilizar informações de energia para a escolha dos nós na transmissão. O sorteio de números aleatórios faz com que a cada difusão, nós diferentes sejam escolhidos para participar do roteamento.

Probabilidade do DEBB-P	Transmissões		Cobertura		Energia residual		Nós mortos	
	Tx	IC	Rx	IC	En (J)	IC	Nm	IC
0.03	72.22	4.983	305.33	8.64	4.67	0.045	1.0	0.816
0.05	74.11	5.65	306.44	12.3	4.55	0.07	0.66	0.66
0.07	77.0	5.13	306.77	11.10	4.44	0.098	1.22	1.22
0.09	92.11	5.85	307.5	12.21	4.07	0.071	1.33	0.66

Tabela 4.2: Média e intervalo de confiança (valores absolutos) da transmissão, cobertura, energia e nós mortos, variando a probabilidade de transmissão máxima do DEBB-P. Tx e Rx são a quantidade de nós que receberam pacotes transmitidos e recebidos respectivamente em uma difusão. En e Nm são energia média dos nós e a quantidade de nós mortos no final da simulação e IC é o intervalo de confiança.

Um bom indicativo da eficiência do protocolo é a razão entre pacotes *recebidos/transmitidos*. Assim, avalia-se a quantidade de pacotes transmitidos necessária para cobrir uma determinada região. O DEBB-P com probabilidade de 0.09 apresenta a menor razão entre pacotes *recebidos/transmitidos* que é 3.33. Já o DEBB-P com probabilidade de 0.07 possui uma razão entre pacotes *recebidos/transmitidos* de 3.98 superior ao DEBB-P com probabilidade de 0.09, mas ligeiramente inferior ao DEBB-P com as probabilidades de 0.05 e 0.03.

Após avaliar o número de transmissões, nós cobertos pela difusão, energia e nós mortos, será avaliada a latência do DEBB-P. A latência é o tempo decorrido entre a transmissão do pacote pelo *sink* e o instante em que o pacote alcançou o primeiro nó sensor localizado a uma distância maior ou igual a um determinado raio a partir do nó *sink*. A figura 4.2 apresenta a latência do DEBB-P variando as probabilidades de transmissão. Observa-se que o DEBB-P com probabilidade de 0.03 possui a maior latência, seguido das probabilidades 0.05, 0.07 e 0.09. O DEBB-P com probabilidade de 0.09 possui a menor latência pois, com menos tentativas, o nó executa a retransmissão do pacote. Entretanto, ele consome mais energia e possui uma menor razão entre pacotes *recebidos/transmitidos*. Já o DEBB-P com probabilidade de 0.07 possui bons resultados de transmissões, cobertura, consumo de energia e de latência, sendo assim, a melhor escolha para aplicações que exigem pouco recursos da rede mas são dependentes do tempo. Assim, nos demais resultados das seções seguintes, o DEBB-P será utilizado com a probabilidade de transmissão máxima de 0.07.

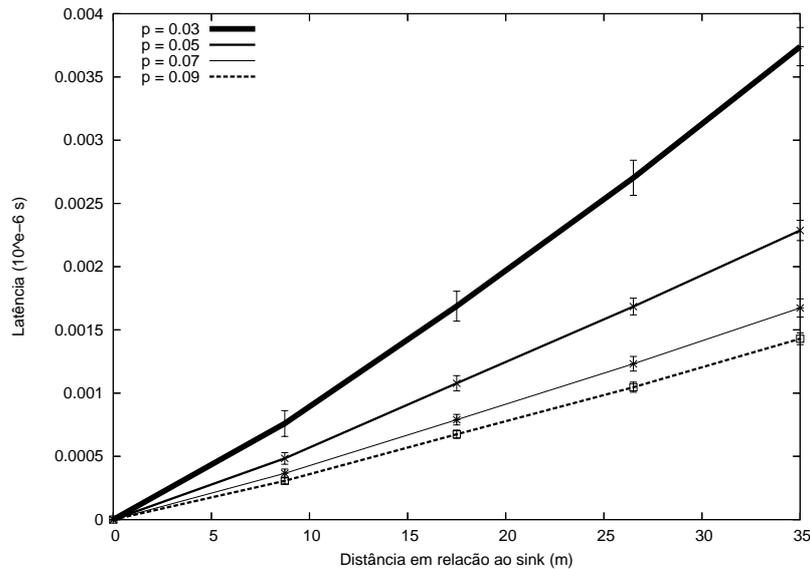


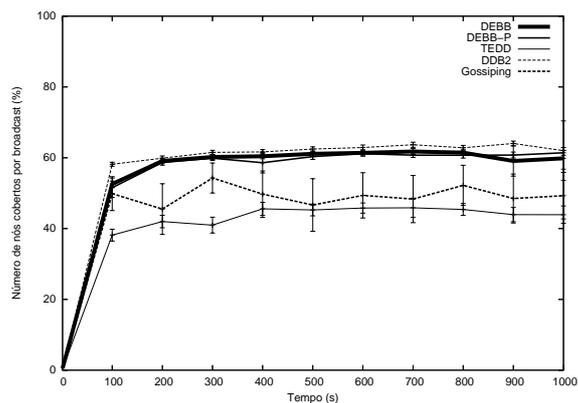
Figura 4.2: Latência nas diferentes probabilidades do DEBB-P.

4.3 Difusão de Dados em Rede com Energia Homogênea

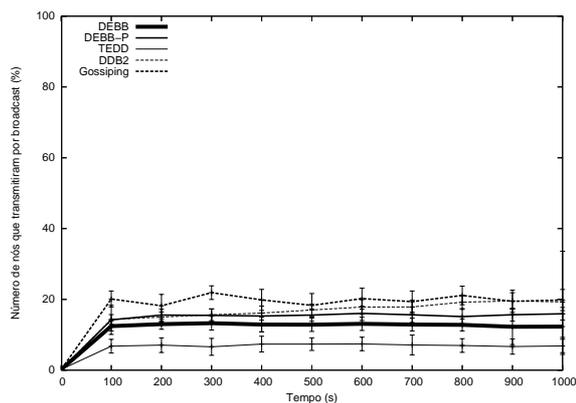
Esta seção analisa o desempenho do DEBB e DEBB-P em um cenário no qual o nó *sink* dissemina informações para todos os nós de uma rede com energia homogênea (figura 4.1 (a)). Nesse caso, o principal objetivo da difusão de dados é maximizar a cobertura da rede realizando o menor número de transmissões possível.

A figura 4.3 ilustra a evolução das métricas avaliadas durante o tempo de simulação. Na figura 4.3 (a), observa-se o número de nós cobertos por difusão. Nessa figura, o DEBB, DEBB-P e o DDB2 apresentam um desempenho similar. Aplicando-se o Teste-T, pode-se afirmar que os três protocolos possuem a mesma cobertura que é de aproximadamente 1.24 vezes maior que a do TEDD e *gossiping*. Além disso, o TEDD e *gossiping* apresentam um desempenho similar (Teste-T). É importante ressaltar que, devido ao fato dos nós adormecerem periodicamente, nem mesmo o *flooding* consegue cobrir toda a rede. De acordo com os resultados apresentados por Goussevskaia et al. (2005a,b), no cenário analisado nesta seção, a cobertura do *flooding* é de aproximadamente 80%.

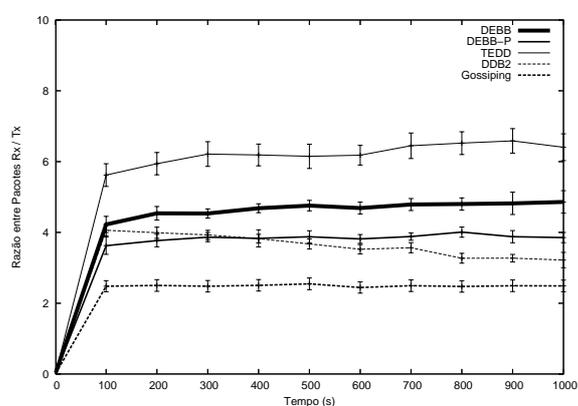
A figura 4.3 (b) apresenta a porcentagem de nós que transmitiu o pacote. Utilizando-se o Teste-T, pode-se afirmar que o DEBB, DEBB-P e o DDB2 transmitiram o mesmo número de pacotes. O *gossiping* transmitiu 1.57, 1.28 e 1.12 vezes mais pacote do que o DEBB, DEBB-P e DDB2 respectivamente. Além disso, o TEDD transmitiu menos pacotes que os demais protocolos (Teste-T). Na figura 4.3 (c), observa-se a evolução da razão entre pacotes *recebidos/transmitidos* em que o DEBB-P e o DDB2 apresentam



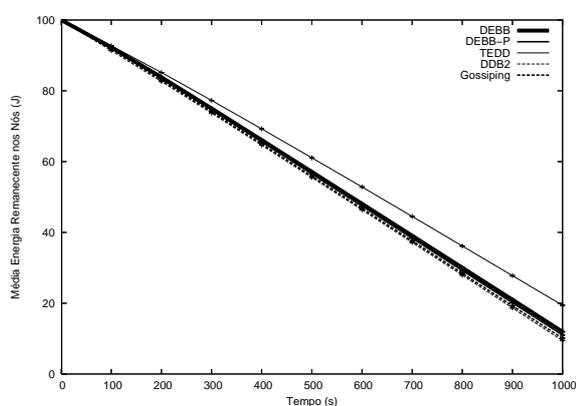
(a) Porcentagem de nós cobertos.



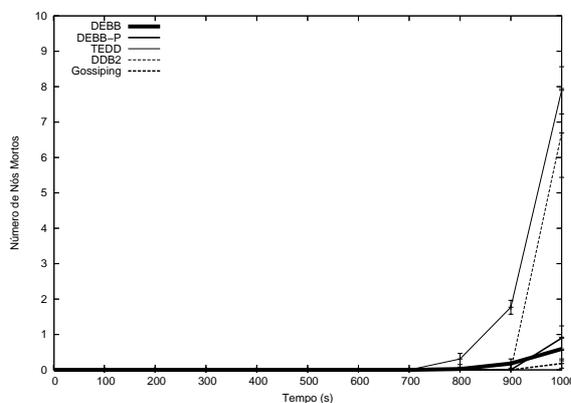
(b) Porcentagem de nós que transmitiu um pacote.



(c) Razão entre os pacotes recebidos/transmitidos.



(d) Energia Média.



(e) Nós mortos.

Figura 4.3: Resultados na rede com energia homogênea.

o mesmo resultado (Teste-T), que é inferior ao DEBB (Teste-T). O desempenho deles é superior apenas ao *gossiping*. O TEDD possui o melhor resultado que é 1.31 vezes melhor que o DEBB. Entretanto, o TEDD possui a pior cobertura na rede, devido ao baixo número de transmissões. Na figura 4.3 (d), observa-se a energia média disponível nos nós e esse resultado é consequência do número de pacotes transmitidos. O TEDD

apresentou o menor consumo e os demais protocolos apresentaram um mesmo valor de consumo (Teste-T). A quantidade de nós mortos é apresentada na figura 4.3 (e). Pode-se observar que o TEDD possui a maior quantidade de nós mortos no final da simulação que é 1.17 vezes maior que o DDB2, 8.86 vezes maior que o DEBB-P, 13.6 vezes maior que o DEBB e 46.41 vezes maior que o *gossiping*.

O objetivo da figura 4.4 é analisar a latência dos protocolos que, como descrito anteriormente é o tempo decorrido entre a transmissão do pacote pelo *sink* e o instante em que o pacote alcançou o primeiro nó sensor localizado a uma distância maior ou igual a um determinado raio a partir do nó *sink*. Nessa figura, nota-se que os maiores valores foram obtidos pelo TEDD, DEBB e DDB2. A latência do DEBB é 1.30 vezes maior que a do TEDD e 1.64 vezes maior que a do DDB2. Outro ponto observado na figura é a influência do mecanismo para a redução de latência dos protocolos baseados em *forwarding delay*. A latência do DEBB, TEDD e DDB2 são respectivamente 34.67, 26.59 e 21.03 vezes maior que a do DEBB-P que utiliza o mecanismo para redução da latência. Por sua vez, como o *gossiping* não insere atrasos na difusão, sua latência é próxima de zero.

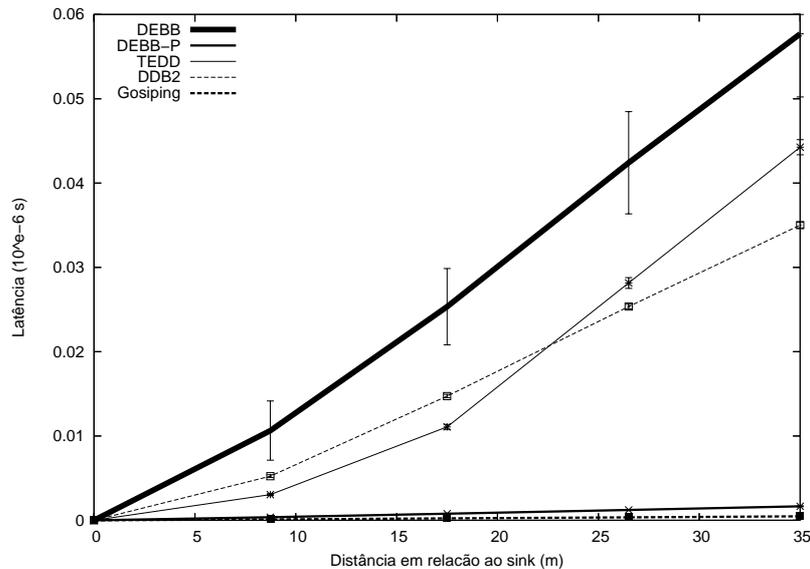


Figura 4.4: Latência dos protocolos.

A tabela 4.3 compara o número de operações realizadas pelos nós sensores para cada um dos protocolos simulados. O *gossiping* realizou o menor número de operações básicas e apresentou a menor cobertura. O TEDD apresentou um bom resultado de operações básicas, porém, como ilustrado na figura 4.3 (a) ele não possui cobertura satisfatória comparado ao DEBB, DEBB-P e DDB2. O DEBB-P apresentou uma cobertura satisfatória e um número de transmissões intermediário em relação aos demais protocolos. Com relação ao número de operações (considerando apenas as operações

de comparação e atribuição), o custo do DEBB-P é maior que o do DEBB porque, no DEBB-P, enquanto o nó ou algum de seus vizinhos não realizar a transmissão do pacote, o processo de avaliação da probabilidade é repetido a cada $TIME_STEP$. O DDB2 apresentou o pior resultado em número de operações básicas, atribuições e comparações e cerca de 3 ordens de grandeza superior aos demais protocolos.

Protocolos	Operações				Comparações	Atribuições	Tx (%)	Rx (%)
	+	-	× /	Sqrt				
<i>gossiping</i>	0.2	-	-	-	10	0	19.69	49.21
TEDD	1.01	2.9	0.3	-	8.2	2.2	7.1	43.97
DEBB	2.86	4.38	-	-	9.47	1.34	12.54	59.09
DEBB-P	0.15	1.17	-	-	16.49	6.47	15.36	59.26
DDB2	2440.34	1221.61	0.4	0.4	1230.22	1220.16	17.58	61.43

Tabela 4.3: Média do número de operações, transmissões e cobertura da rede em cada difusão de dados.

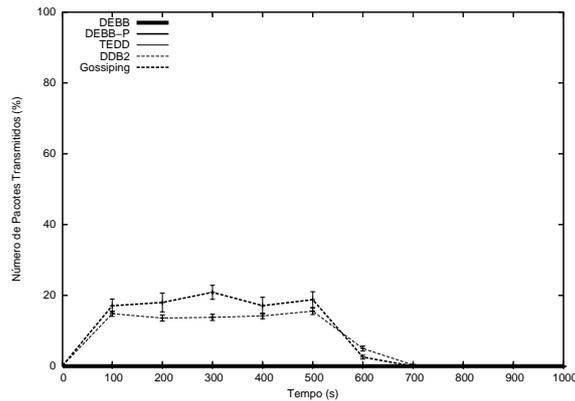
Os resultados apresentados na difusão de dados para uma rede com energia inicial homogênea mostraram que as duas versões do DEBB conseguem manter uma cobertura idêntica ao DDB2 e superior ao TEDD e *gossiping*. Uma vantagem dos protocolos propostos neste trabalho é o custo computacional que é reduzido devido à simplicidade do protocolo que minimiza a carga de trabalho realizada nos nós sensores. Destaca-se que o custo computacional do DEBB e DEBB-P é, em média, 3 ordens de grandeza menor que o DDB2. O custo computacional do DEBB é menor que o do DEBB-P porque neste, enquanto um nó ou um de seus vizinhos não transmitir o pacote, o processo de *forwarding delay* é repetido, ocasionando uma maior quantidade de comparações e atribuições. Finalmente, esta seção mostrou a eficiência do mecanismo de redução da latência utilizado pelo DEBB-P. É importante ressaltar que esse mecanismo pode ser estendido para qualquer protocolo baseado em *forwarding delay* para reduzir a latência dos algoritmos dessa classe. Finalmente, o DEBB e DEBB-P apresentaram o melhor desempenho em relação ao objetivo da difusão de dados que consiste em maximizar a cobertura, realizando o menor número de transmissões com um baixo custo operacional.

4.4 Difusão de Dados Evitando Uma Região de Baixa Energia

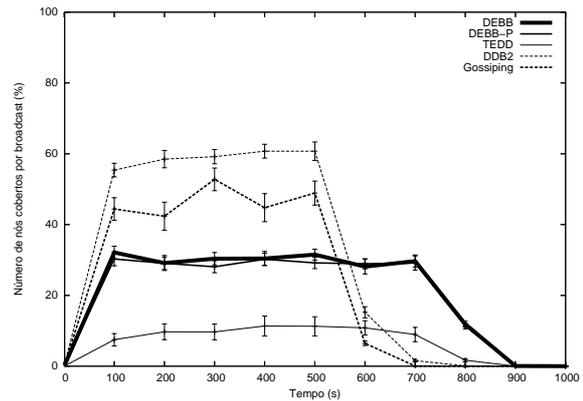
Esta seção estuda um cenário de difusão de dados que contém uma região de baixa energia que consiste de um círculo de raio 7 m localizado de maneira aleatória na área de sensoriamento. O número de nós existentes dentro dessa região é aproximadamente 63. Na difusão de dados em cenários contendo regiões de baixa energia, o principal objetivo é evitar o fluxo de dados dentro das regiões críticas e, conseqüentemente, prolongar o tempo de vida dos nós localizados dentro dessas regiões. Maximizar a cobertura da rede (dentro e fora da região crítica) e minimizar o número de transmissões também são objetivos nesse cenário.

A figura 4.5 apresenta os resultados de simulação obtidos para os nós localizados dentro da região de baixa energia. Na figura 4.5 (a), observa-se que o DEBB, o DEBB-P e o TEDD eliminaram completamente o fluxo de dados dentro da região crítica. O *gossiping* e DDB2 realizaram várias transmissões dentro dessa região e possuem os mesmos resultados (Teste-T). A figura 4.5 (b) mostra a cobertura dentro da região de baixa energia, Nesse caso, o DEBB e DEBB-P apresentam o mesmo resultado (Teste-T). O DDB2 apresenta o melhor resultado até o instante 500 s do tempo de vida da rede e cerca de 1.38 vezes melhor que o *gossiping*, 2.0 vezes melhor que o DEBB e DEBB-P e 6.02 vezes melhor que o TEDD. Após o instante 500 s de tempo de vida rede, a cobertura do DDB2 e *gossiping* decresce rapidamente porque nesses protocolos os nós começam a morrer por falta de energia. O DEBB e DEBB-P possuem uma cobertura de 3 vezes melhor que o TEDD e conseguem prolongar a cobertura dentro da região de baixa energia comparado aos demais protocolos. Neste ponto, destaca-se a capacidade do DEBB e do DEBB-P em contornar regiões de baixa energia de forma eficiente uma vez que eles conseguem eliminar completamente o fluxo de dados dentro das regiões. Basicamente, os protocolos propostos nesta dissertação conseguem contornar de forma precisa as fronteiras das regiões críticas e selecionar os nós que podem e os que não podem transmitir os pacotes de difusão. Outro ponto importante é que o DEBB e o DEBB-P, assim como o TEDD, são capazes de manter a cobertura dentro da região crítica por mais tempo que o *gossiping* e DDB2. As figuras 4.5 (c) e 4.5 (d) mostram a energia média dos nós e a porcentagem de nós mortos dentro da região de baixa energia. Como o DEBB, DEBB-P e o TEDD não transmitiram pacotes dentro dessa região, eles conseguem economizar energia e retardar a morte dos nós na região crítica. Destaca-se que na figura 4.5 (c), as curvas do DEBB e DEBB-P estão sobrepostas.

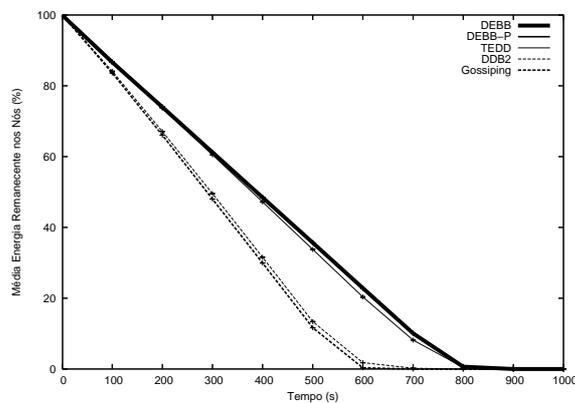
A figura 4.6 apresenta os resultados de simulação dos nós localizados fora da região de baixa energia. Nessa figura, observa-se a porcentagem de nós cobertos, a porcenta-



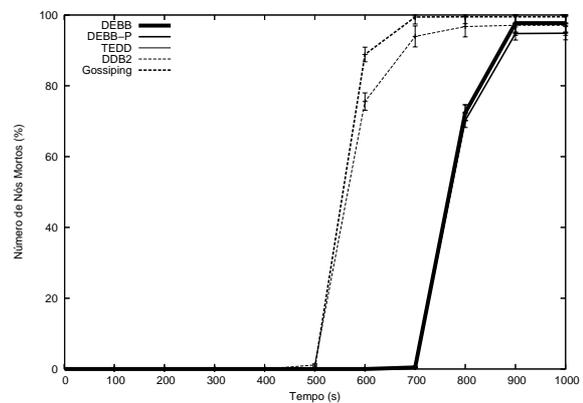
(a) Porcentagem de nós que transmitiu um pacote.



(b) Porcentagem de nós cobertos.



(c) Energia Média.



(d) Porcentagem de nós mortos.

Figura 4.5: Resultados dentro da região de baixa energia.

gem de nós que transmitiu o pacote e a energia média disponível nos nós sensores. A cobertura obtida pelo DEBB, DEBB-P e DDB2 apresentam o mesmo resultado (Teste-T) e, em média, 1,24 e 2,03 vezes maior que a do *gossiping* e TEDD respectivamente. Em relação ao número de transmissões, verifica-se que o DEBB, DEBB-P, DDB2 e o *gossiping* apresentaram o mesmo resultado (Teste-T) e transmitiram, em média, 3,91 vezes mais pacotes que o TEDD. É importante observar que no TEDD, apenas os nós localizados próximos às curvas transmitem e são cobertos por um pacote disseminado. Como a estratégia do TEDD neste tipo de cenário é evitar regiões de baixa energia, poucas curvas são geradas e isso implica em reduções de cobertura e transmissões. A figura 4.7 mostra um *snapshot* de cada um dos protocolos avaliados na qual é possível visualizar a menor cobertura do TEDD em relação ao DEBB, DEBB-P, DDB2 e o *gossiping*. Portanto, o DEBB, DEBB-P e o DDB2 apresentam o melhor compromisso entre a cobertura e a transmissão. Na figura 4.6 (c), observa-se que o consumo de energia foi proporcional ao número de transmissões realizadas pelos protocolos.

Os resultados apresentados na difusão de dados para rede contendo uma região

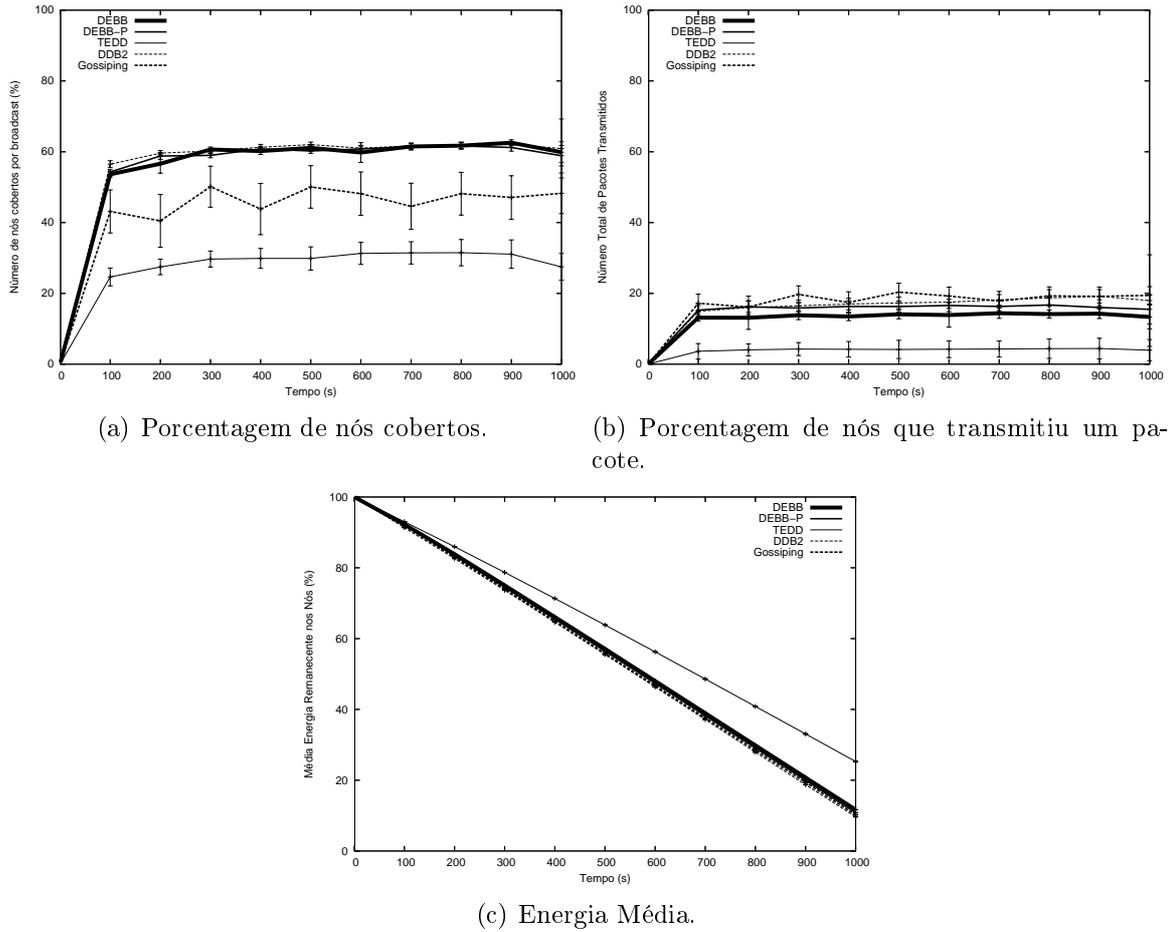


Figura 4.6: Resultados fora da região de baixa energia.

crítica mostram a capacidade do DEBB e do DEBB-P de contornarem uma região de baixa energia. Os dois protocolos conseguem eliminar completamente o fluxo de dados dentro dessa região e, também conseguem bons resultados em termos de cobertura comparados ao DDB2, TEDD e *gossiping*. Além disso, o DEBB, DEBB-P e o DDB2 possuem o melhor resultado de cobertura fora da região crítica. Um ponto importante observado nesta seção é que, apesar do TEDD eliminar o fluxo dentro da região crítica, ele não apresenta bons resultados em termos de cobertura quando comparado com os protocolos propostos por este trabalho. O resultado do TEDD é verificado porque sua difusão de dados é realizada ao longo das curvas de disseminação e os nós mais distantes das curvas não transmitem e nem são cobertos pelos pacotes difundidos. Além disso, foi verificado que o DDB2 não consegue eliminar o fluxo de dados dentro das regiões de baixa energia. Assim, em cenários onde a rede possui regiões de baixa energia, o DEBB e DEBB-P mostram-se mais eficientes no processo de difusão.

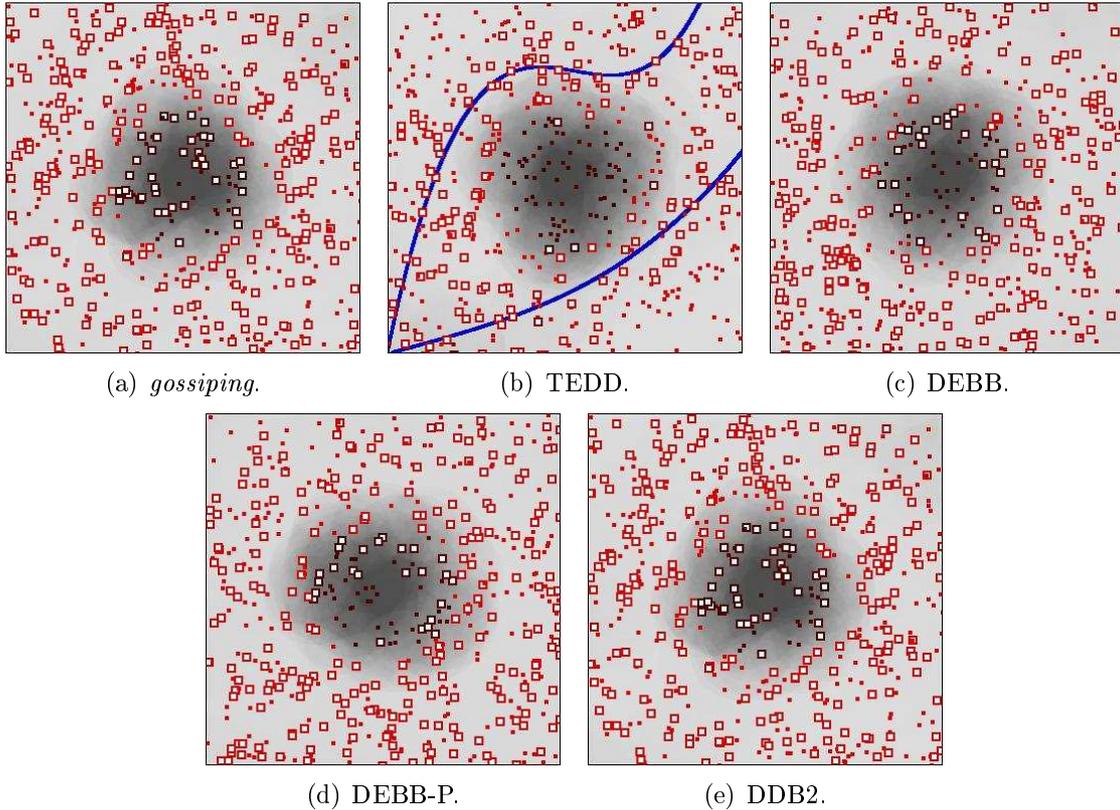


Figura 4.7: *Snapshot* dos nós cobertos pelos diferentes protocolos utilizando o cenário da figura 4.1 (b). Os quadrados brancos representam nós que receberam o pacote de difusão. Os quadrados pretos representam os nós que não receberam o pacote de difusão.

4.5 Difusão de Dados Evitando Quatro Regiões de Baixa Energia

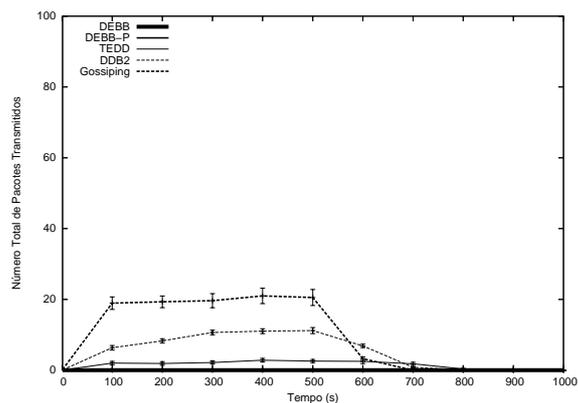
Esta seção estuda um cenário em que o nó *sink* dissemina dados para a rede e essa contém quatro regiões de baixa energia, como ilustrado na figura 4.1 (c). Cada região de baixa energia corresponde a um círculo de raio $3.5 m$ e possui em média 15.75 nós. Destaca-se que a área e a quantidade de nós de cada uma das regiões de baixa energia corresponde a um quarto dos valores utilizados na região de baixa energia da seção 4.4. Os objetivos da difusão de dados nesse cenário são os mesmos da seção contendo uma única região crítica.

A figura 4.8 apresenta os resultados de simulação obtidos para os nós localizados dentro das regiões de baixa energia. As figuras 4.8 (a) e 4.8 (b) mostram o número de transmissões e a quantidade de nós cobertos dentro das regiões críticas. Assim como na seção anterior, o DEBB e o DEBB-P conseguiram eliminar completamente o fluxo de dados dentro das regiões. Aplicando-se o Teste-T, observa-se que o desempenho dos dois protocolos é igual. Até o instate 530 s de simulação, o DDB2 e *gossiping*

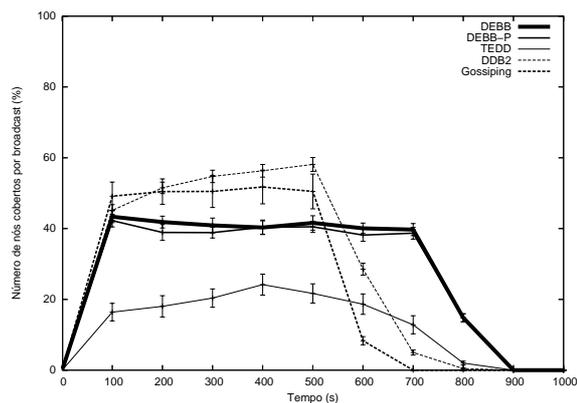
apresentaram o melhor desempenho com relação ao número de nós cobertos e possuem os mesmos resultados (Teste-T). Contudo, nota-se que o TEDD, DDB2 e *gossiping* não foram capazes de contornar as regiões de baixa energia. No TEDD, isso se deve ao fato de que à medida que o número de regiões de baixa energia aumenta, a geração de curva torna-se um problema mais complexo e difícil de ser tratado. Destaca-se a dificuldade de se gerar curvas para cenários que possuem muitas regiões de baixa energia. No DDB2, isso deve-se ao fato de que o protocolo tenta escolher os nós com mais reserva de energia para transmitir o pacote e não garantir que nós com baixa reserva de energia não participem do roteamento. Após o instante 530 s de tempo de vida rede, a cobertura do DDB2 e *gossiping* decresce rapidamente porque nesses protocolos os nós começam a morrer por falta de energia. Outro resultado observado é que o desempenho das duas versões do DEBB em termos de cobertura foi melhor que o da seção anterior quando comparados aos desempenhos do *gossiping*, DDB2 e do TEDD. Além disso, o DEBB e DEBB-P conseguem manter uma cobertura mínima por mais tempo dentro da região de baixa energia. Nas figuras 4.8 (c) e 4.8 (d), observa-se a energia média dos nós e a porcentagem de nós mortos durante a simulação. Os resultados obtidos nessas figuras são consequência do número de transmissões. Como as duas versões do DEBB conseguem eliminar o fluxo de dados dentro das regiões críticas, os dois protocolos consomem menos energia e prolongam o tempo de vida dos nós localizados nas regiões de baixa energia. Destaca-se que os resultados do DEBB e DEBB-P são os mesmos (Teste-T) para o consumo de energia e nós mortos.

A figura 4.9 mostra um *snapshot* de cada um dos protocolos avaliados. Nesse caso, é interessante observar que as curvas geradas pelo TEDD não conseguem desviar das regiões de baixa energia. Conseqüentemente, existem transmissões dentro das regiões críticas e os nós morrem antes quando comparado com a difusão de dados utilizando uma das versões do DEBB. Finalmente, destaca-se que o DEBB e DEBB-P possuem resultados semelhantes ao DDB2 e *gossiping*, que possuem as melhores coberturas, não efetuando nenhuma transmissão dentro das regiões.

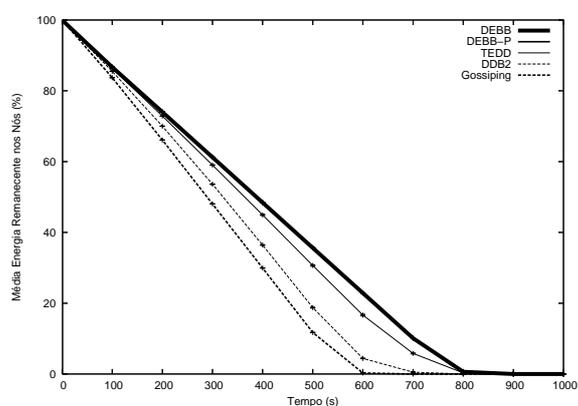
A figura 4.10 apresenta os resultados de simulação dos nós localizados fora das regiões de baixa energia. Nessa figura, observa-se a porcentagem de nós cobertos, a porcentagem de nós que transmitiu um pacote e a energia média, respectivamente. Em todas as figuras, observa-se que os resultados obtidos pelas duas versões do DEBB são iguais e isso é confirmado aplicando-se o Teste-T. Com relação à cobertura, verifica-se que o desempenho dos protocolos DEBB, DEBB-P e DDB2 são os mesmos (Teste-T) e em média 1.28 vezes melhor que o *gossiping* e 2.11 vezes melhor que o TEDD. Na figura 4.10 (b), nota-se que o TEDD transmitiu menos pacotes que o *gossiping*, DDB2 e que as duas versões do DEBB. Com relação à energia média dos nós, o resultado é proporcional ao número de pacotes transmitidos. Finalmente, destaca-se que o



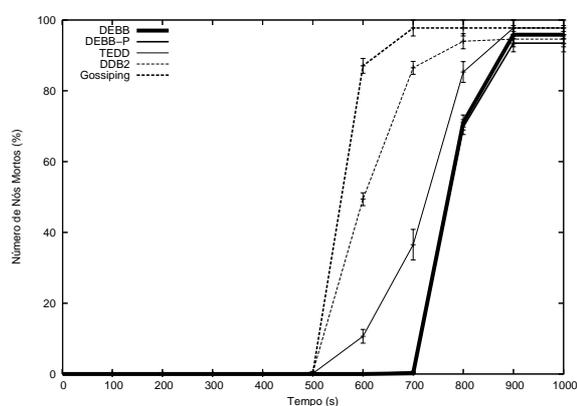
(a) Porcentagem de nós que transmitiu um pacote.



(b) Porcentagem de nós cobertos.



(c) Energia Média.



(d) Porcentagem de nós mortos.

Figura 4.8: Resultados dentro das regiões de baixa energia.

DEBB, DEBB-P e o DDB2 apresentam o melhor compromisso entre a maximização da cobertura e a minimização do número de transmissões.

No cenário com quatro regiões de baixa energia, os resultados apresentados para a difusão de dados mostram a escalabilidade do DEBB e do DEBB-P em relação ao número de regiões de baixa energia. Os dois protocolos conseguem eliminar completamente o fluxo de dados dentro dessas regiões. Outro ponto importante mostrado nesta seção foi que o TEDD e o DDB2 não são capazes de eliminar o fluxo de dados quando o número de regiões de baixa energia aumenta. No TEDD, isso acontece porque à medida que esse número cresce, o problema da geração dinâmica de curvas torna-se mais difícil, logo, alguns trechos das curvas geradas passam sobre as regiões críticas. No DDB2, como descrito anteriormente, o seu mecanismo tenta evitar que nós com baixa reserva de energia transmitam o pacote. Entretanto, ele não consegue eliminar o fluxo de dados dentro de regiões de baixa energia. Finalmente, em cenários com várias regiões de baixa energia, o DEBB e DEBB-P possuem o melhor compromisso segundo as métricas avaliadas, sendo assim a melhor escolha para esses cenários.

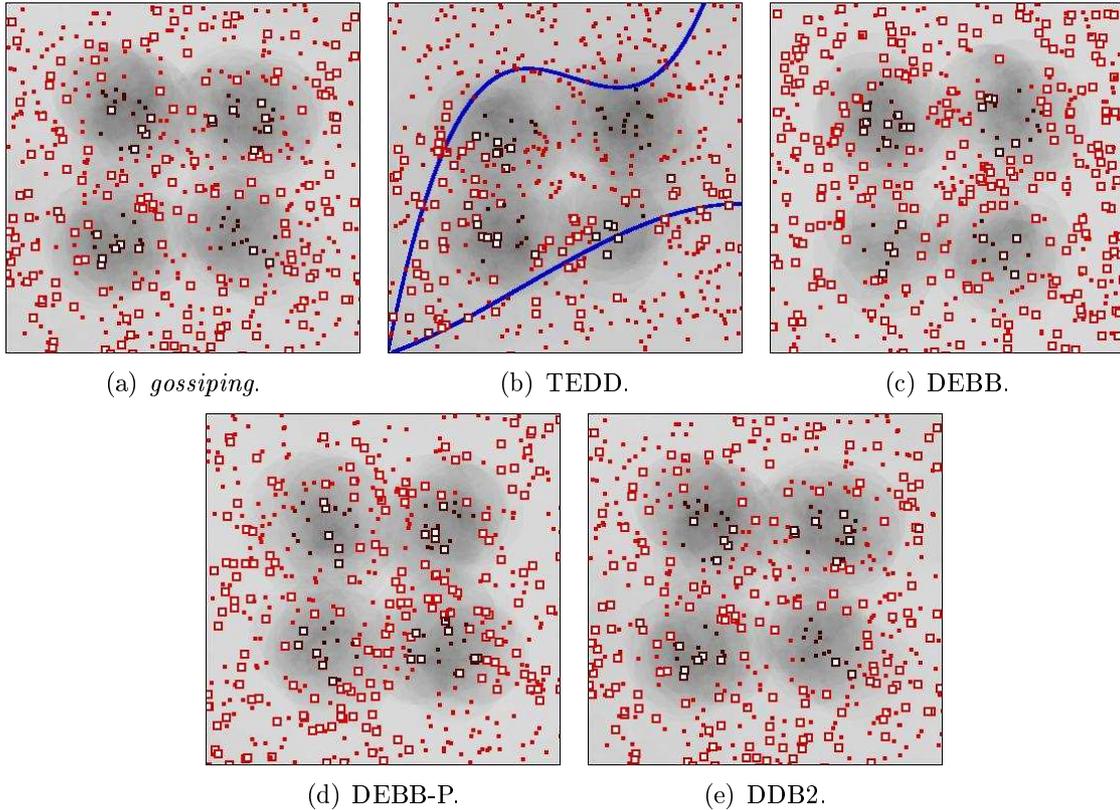
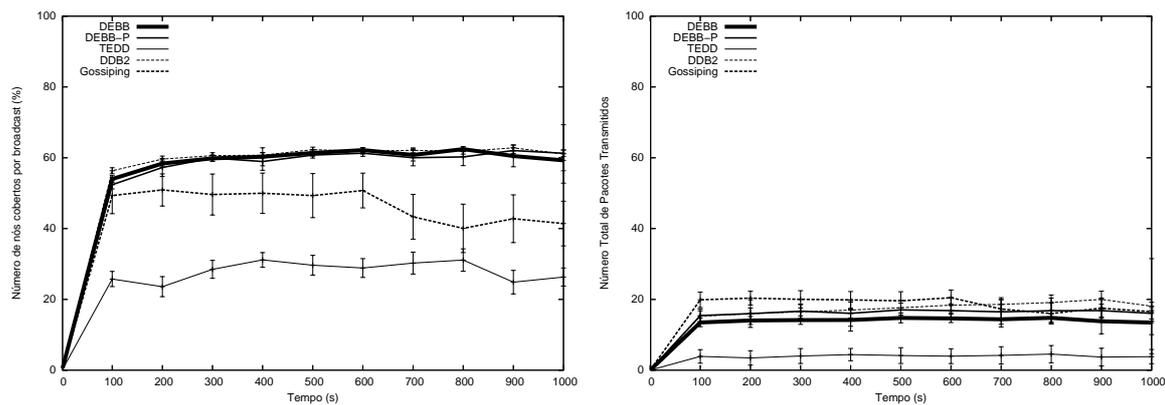


Figura 4.9: *Snapshot* dos nós cobertos pelos diferentes protocolos utilizando o cenário da figura 4.1 (c). Os quadrados brancos representam nós que receberam o pacote de difusão. Os quadrados pretos representam os nós que não receberam o pacote de difusão.

4.6 Conclusão

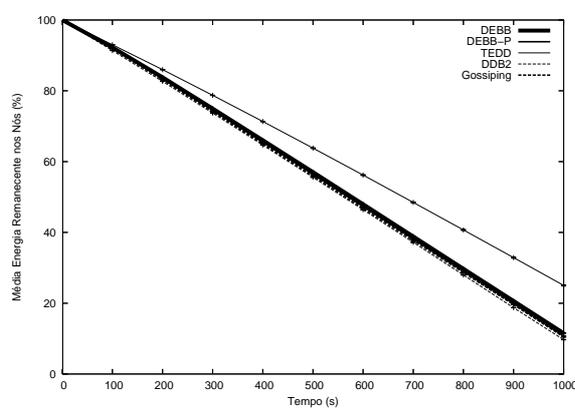
Nessa seção os protocolos de roteamento propostos nesta dissertação foram comparados com protocolos da literatura. Além disso, foi feita uma avaliação do impacto da probabilidade de transmissão do DEBB-P. Observa-se que, a medida que a probabilidade de transmissão aumenta, a latência no roteamento diminui. Porém, a quantidade de pacotes transmitidos aumenta e a cobertura mantém-se constante.

Em cenários que a energia dos nós é homogênea, o DEBB, DEBB-P e DDB2 apresentaram resultados similares. Entretanto, a quantidade de operações básicas realizadas pelo DDB2 é aproximadamente 3 ordens de grandeza superior aos demais protocolos. Nos cenários com regiões de baixa energia, o DEBB e DEBB-P conseguem eliminar o fluxo de dados dentro dessas regiões de maneira simples e eficiente. O TEDD consegue eliminar o fluxo de dados dentro de regiões de baixa energia quando a rede possui apenas 1 região. Isso acontece pois, a geração de curvas em cenários com várias regiões de baixa energia é uma tarefa difícil. O DDB2 não conseguiu eliminar o fluxo de dados em regiões de baixa energia, mesmo quando o tamanho dessas é reduzido. Como o *gossiping* não possui nenhum mecanismo para desviar o roteamento dessas regiões,



(a) Porcentagem de nós cobertos.

(b) Porcentagem de nós que transmitiu um pacote.



(c) Energia Média.

Figura 4.10: Resultados fora das regiões de baixa energia.

várias transmissões são feitas por nós com baixa reserva de energia. De maneira geral, o DEBB e DEBB-P são os protocolos com melhor desempenho quando avaliados nos cenários de energia homogênea e cenários com regiões de baixa energia.

Capítulo 5

Conclusões e Direções Futuras

Neste trabalho foram propostos dois novos protocolos de difusão de dados em redes de sensores sem fio: DEBB e DEBB-P. Esses protocolos utilizam o mapa de energia e o *forwarding delay* com a métrica de distância. O mapa de energia é utilizado a fim de extrair informações sobre regiões de baixa energia. Resultados de simulação mostram que quando o DEBB e o DEBB-P são utilizados, o processo de roteamento torna-se mais adaptativo com relação ao dinamismo da topologia da rede e à quantidade de regiões de baixa energia. Assim, mesmo em cenários com várias regiões de baixa energia, a energia gasta na atividade de roteamento pode ser concentrada nos nós que possuem maior reserva de energia. Dessa forma, nós com pouca energia podem dedicar-se a atividade de sensoreamento e ao recebimento de informações destinadas a eles. A principal vantagem do DEBB em relação ao DEBB-P é o menor número de operações básicas realizadas. Por outro lado, o DEBB-P apresenta um desempenho superior quando a métrica latência é avaliada. O mecanismo de redução de latência utilizado pelo DEBB-P pode ser utilizado para minimizar o problema de latência dos protocolos baseados nas técnicas de *forwarding delay*. O TEDD não se mostrou eficiente em cenários com várias regiões de baixa energia devido ao fato da complexidade da geração de curvas. O DDB2 mostrou-se eficiente em cenários com energia homogênea, mas a sua técnica de escolher os nós com maiores reservas de energia para participar do roteamento não mostrou-se eficiente em cenários que regiões de baixa energia devem ser desviadas. Assim, o TEDD e DDB2 não foram capazes de eliminar o fluxo de dados dentro das regiões críticas, não sendo, portanto, adequado para difusão de dados nesses cenários.

Como extensão deste trabalho, pretende-se desenvolver um modelo analítico para os protocolos propostos. Através do modelo analítico é possível definir os limites inferior e superior para a classe de problemas que o mesmo se propõe realizar. Dessa forma, será possível saber o quão próximo a solução apresentada estará do melhor e do pior caso

do problema em questão. O trabalho pode ser estendido no sentido de propor novas métricas para o *forwarding delay* e novas formas de se calcular o valor da probabilidade do DEBB-P como, por exemplo, levar em consideração o número de nós vizinhos.

Referências Bibliográficas

- Braginsky, D. e Estrin, D. (2002). Rumor routing algorithm for sensor networks. In *The First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA 2002)*, Westin Peachtree Plaza, Atlanta, GA, USA.
- Cartigny, J.; Ingelrest, F.; Simplot-Ryl, D. e Stojmenovic, I. (2005). Localized lmsr and rng based minimum energy broadcast protocols in ad hoc networks. *Ad Hoc Networks*, 3(1):1–16.
- Chang, C.-Y.; Shih, K.-P. e Lee, S.-C. (2005). Zbp: A zone-based broadcasting protocol for wireless sensor networks. *Wireless Personal Communication.*, 33(1):53–68.
- do V. Machado, M.; Goussevskaia, O.; Mini, R. A. F.; Rezende, C. G.; Loureiro, A. A. F.; Mateus, G. R. e Nogueira, J. M. (2005). Data dissemination using the energy map. In *WONS '05: Proceedings of the Second Annual Conference on Wireless On-demand Network Systems and Services (WONS'05)*, pp. 139–148, Washington, DC, USA. IEEE Computer Society.
- do Val Machado, M.; Goussevskaia, O.; Mini, R. A.; Loureiro, A. A.; Mateus, G. R. e Nogueira, J. M. (2005). Data dissemination in autonomic wireless sensor networks. *Special Issue of Autonomic Communication Systems in IEEE Journal on Selected Areas in Communications (JSAC)*, 23(12):2305 – 2319.
- Durresi, A.; Paruchuri, V. K.; Iyengar, S. e Kannan, R. (2005). Optimized broadcast protocol for sensor networks. *IEEE Transactions On Computers*, 54(8):1013–1024.
- Estrin, D.; Govindan, R.; Heidemann, J. e Kumar, S. (1999). Next century challenges: scalable coordination in sensor networks. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pp. 263–270, New York, NY, USA. ACM.
- Ganesan, D.; Govindan, R.; Shenker, S. e Estrin, D. (2001). Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *ACM Mobile Computing and Communications Review*, 5(4).

- Goussevskaia, O.; do Val Machado, M.; Mini, R. A.; Loureiro, A. A.; Mateus, G. R. e Nogueira, J. M. (2005a). Avoiding low-energy areas in wireless sensor network data dissemination. In *XXIII Simpósio Brasileiro de Redes de Computadores*.
- Goussevskaia, O.; do Val Machado, M.; Mini, R. A.; Loureiro, A. A.; Mateus, G. R. e Nogueira, J. M. (2005b). Data dissemination based on the energy map. In *IEEE Communications Magazine*.
- Guidoni, D. L.; do Val Machado, M.; Mini, R. A. F. e Loureiro, A. A. F. (2006). Difusão de dados baseada em atraso e energia para redes de sensores sem fio. In *XXVI Simpósio Brasileiro de Redes de Computadores (SBRC) (Trabalho recebeu menção Honrosa e foi classificado entre os 6 melhores da conferência)*, Curitiba, PR, Brasil.
- Heinzelman, W. B. (2000). *Application-Specific Protocol Architectures for Wireless Networks*. PhD thesis, Massachusetts Institute of Technology.
- Heinzelman, W. R.; Chandrakasan, A. e Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. In *In Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8*, p. 8020. IEEE Computer Society.
- Heinzelman, W. R.; Kulik, J. e Balakrishnan, H. (1999). Adaptive protocols for information dissemination in wireless sensor networks. In *MOBICOM 99*, pp. 174–185, Seattle, WA USA.
- Heissenbüttel, M. (Jul - 2005). *Routing and Broadcasting in Ad-Hoc Networks*. PhD thesis, University of Bern.
- Heissenbüttel, M.; Braun, T.; Wälchli, M. e Bernoulli, T. (2006). Optimized stateless broadcasting in wireless multi-hop networks. In *IEEE Infocom 2006*, Barcelona, Spain.
- Hill, J.; Szewczyk, R.; Woo, A.; Hollar, S.; Culler, D. e Pister, K. (2000). System architecture directions for networked sensors. In *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*.
- Ingelrest, F.; Simplot-Ryl, D. e Stojmenovic, I. (2006). Optimal transmission radius for energy efficient broadcasting protocols in ad hoc and sensor networks. *IEEE Trans. Parallel Distrib. Syst.*, 17(6):536–547.

- Intanagonwiwat, C.; Estrin, D.; Govindan, R. e Heidemann, J. (2002). Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *International Conference on Distributed Computing Systems (ICDCS)*, Vienna, Austria.
- Intanagonwiwat, C.; Govindan, R. e Estrin, D. (2000). Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of the sixth annual international conference on Mobile computing and networking*, pp. 56–67, Boston, MA USA.
- Intanagonwiwat, C.; Govindan, R.; Estrin, D.; Heidemann, J. e Silva, F. (2003). Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. Netw.*, 11(1):2–16.
- Jain, R. (1991). *The Art of Computer Systems Performance Analysis: techniques for experimental design, measurement, simulation and modeling*. John Wiley.
- Kulik, J.; Heinzelman, W. R. e Balakrishnan, H. (1999). Negotiation-based protocols for disseminating information in wireless sensor networks. In *ACM/IEEE Int. Conf. on Mobile Computing and Networking*, Seattle, WA.
- Mann, R. P.; Namuduri, K. R. e Pendse, R. (2005). Energy-aware routing protocol for ad hoc wireless sensor networks. *EURASIP J. Wirel. Commun. Netw.*, 5(5):635–644.
- Mica2 (2004). Mts/mda sensor and data acquisition boards user's manual. www.xbow.com.
- Mini, R. A. F. (Jan - 2004). *Mapa de Energia baseado em Predição para Redes de Sensores Sem Fio*. PhD thesis, Departamento de Ciência da Computação - Universidade Federal de Minas Gerais.
- Mini, R. A. F.; do Val Machado, M.; Loureiro, A. A. F. e Nath, B. (2005). Prediction-based energy map for wireless sensor networks. *Ad Hoc Networks Journal*, 3(2):235–253.
- Mini, R. A. F.; Loureiro, A. A. F. e Nath, B. (2003a). Prediction-based energy map for wireless sensor networks. In *Personal Wireless Communications - PWC 2003*, Venice - Italy.
- Mini, R. A. F.; Nath, B. e Loureiro, A. A. F. (2002). A probabilist approach to predict the energy consumption in wireless sensor networks. In *IV Workshop de Comunicação sem Fio e Computação Móvel*, São Paulo, Brazil.
- Mini, R. A. F.; Nath, B. e Loureiro, A. A. F. (2003b). Prediction-based approaches to construct the energy map for wireless sensor networks. In *XXI Simpósio Brasileiro de Redes de Computadores (SBRC)*, Natal, RN, Brasil.

- Niculescu, D. e Nath, B. (2003). Trajectory-Based Forwarding and its Applications. In *MOBICOM 03*, pp. 260–272, USA.
- ns2 (2002). The network simulator. www.isi.edu/nsnam/ns.
- Orecchia, L.; Panconesi, A.; Petrioli, C. e Vitaletti, A. (2004). Localized techniques for broadcasting in wireless sensor networks. In *DIALM-POMC '04: Proceedings of the 2004 joint workshop on Foundations of mobile computing*, pp. 41–51, New York, NY, USA. ACM Press.
- Ovalle-Martinez, F. J. (2005). Area based beaconless broadcasting in ad hoc and sensor networks. Master's thesis, University of Ottawa Canada.
- Pottie, G. e Kaiser, W. (2000a). Embedding the internet wireless integrated network sensors. In *Communications of the ACM*, volume 43, pp. 51–58.
- Pottie, G. e Kaiser, W. (2000b). Wireless integrated network sensors. In *Communications of the ACM*, volume 43, pp. 551–8.
- Rahnavard, N. e Fekri, F. (2006a). Crbcast: a collaborative rateless scheme for reliable and energy-efficient broadcasting in wireless sensor networks. In *IPSN '06: Proceedings of the fifth international conference on Information processing in sensor networks*, pp. 276–283, New York, NY, USA. ACM Press.
- Rahnavard, N. e Fekri, F. (2006b). Crbcast: a collaborative rateless scheme for reliable and energy-efficient broadcasting in wireless sensor networks. In *IPSN '06: Proceedings of the fifth international conference on Information processing in sensor networks*, pp. 276–283, New York, NY, USA. ACM Press.
- Servetto, S. e Barrenechea, G. (2002). Constrained random walks on random graphs: Routing algorithms for large scale wireless sensor networks. In *First ACM International Workshop on Wireless Sensor Networks and Applications - WSNA*, Atlanta, Georgia, USA.
- Su, W. e Akyildiz, I. F. (2005). Time-diffusion synchronization protocol for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 13(2):384–397.
- Wood, A. D.; Fang, L.; Stankovic, J. A. e He, T. (2006). Sigf: a family of configurable, secure routing protocols for wireless sensor networks. In *SASN '06: Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks*, pp. 35–48, New York, NY, USA. ACM Press.

- Wu, J. e Dai, F. (2004). A generic distributed broadcast scheme in ad hoc wireless networks. *IEEE Trans. Computers*, 53(10):1343–1354.
- Wu, J. e Dai, F. (2005). Efficient broadcasting with guaranteed coverage in mobile ad hoc networks. *IEEE Trans. Mob. Comput.*, 4(3):259–270.
- Yu, Y.; Govindan, R. e Estrin, D. (2001). Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. Technical Report UCLA/CSD-TR-01-0023, UCLA Computer Science Department Technical Report.
- Zhao, Y. J.; Govindan, R. e Estrin, D. (2002). Residual energy scans for monitoring wireless sensor networks. In *IEEE Wireless Communications and Networking Conference (WCNC'02)*, Orlando, FL, USA.