

# **Geração de Curvas de Roteamento para Redes de Sensores Sem Fio**

**Olga N. Goussevskaia**

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

Orientador: Geraldo Robson Mateus

Co-Orientador: Antonio A. F. Loureiro

Belo Horizonte  
Abril de 2005



# Agradecimentos

---

---

*I*nicialmente, eu gostaria de agradecer ao Departamento de Ciência da Computação da UFMG, pelas oportunidades concedidas durante o mestrado e a graduação. Ao Geraldo Robson e Antônio Loureiro, por terem sido meus tutores desde o meu primeiro ano de graduação e por terem me ensinado a gostar do trabalho de pesquisa. À Raquel e ao Max, por terem sido grandes parceiros e amigos durante a realização deste trabalho. Gostaria de agradecer especialmente aos meus pais, pelo apoio, amor e incentivo. E aos amigos, familiares e demais pessoas que amo, por existirem e tornarem minha vida melhor.



# Abstract

---

---

Beyond today's embedded technology, next generation micro sensors incorporating wireless technology will be able to cost-effectively connect large numbers of tiny, inexpensive devices, originating the *Wireless Sensor Networks* (WSNs), sometimes called "smart dust". This technology will enable new kinds of pervasive computing and smart environments

The routing problem in WSNs is often called "data dissemination". An interesting data dissemination technique is trajectory-based forwarding. The innovation of this approach consists in representing the route as a continuous function, instead of a set of discrete points. The main idea is to embed a curve equation into the packet header and let the intermediate nodes forward it to nodes located close to the curve. This technique is immune to changes in network connectivity and is scalable to the size of the network and the number of nodes comprising the route.

Trajectory-based forwarding assumes there is a mechanism to generate the trajectories. However, as far as we know, the problem of trajectory generation has not been addressed in the literature. In this work, a method to dynamically generate routing curves based on the energy map of the network is proposed.

Simulation results reveal that the energy spent with data dissemination activity can be concentrated on node with high energy reserves, whereas low-energy nodes can use their energy only to perform sensing activity. In this manner, partitions of the network due to nodes that run out of energy can be significantly delayed and the network lifetime extended.

The contributions of this thesis were published in *IEEE Communications Magazine (Ad Hoc and Sensor Networks Series)* [11] and *Journal on Selected Areas in Communication (Special Issue on Autonomics)* [8].



# Resumo

---

Ultrapassando a tecnologia embutida de hoje, a próxima geração de micro-sensores irá incorporar a tecnologia sem fio, o que irá permitir que um grande número de pequenos dispositivos de baixo preço sejam conectados, formando as *Redes de Sensores Sem Fio* (RSSFs), também chamadas de “poeira inteligente”. Essa tecnologia irá possibilitar novos tipos de computação ubíqua e ambientes inteligentes.

O problema de roteamento em RSSFs é freqüentemente denominado “disseminação de dados”. Uma técnica interessante de disseminação de dados é a baseada em trajetórias, ou disseminação sobre curvas. A inovação dessa abordagem está na representação de rotas através de funções contínuas, ao invés de conjuntos de pontos discretos. A idéia principal é embutir uma equação de curva no cabeçalho do pacote e deixar que nós intermediários o transmitam para nós próximos à curva. Essa técnica é imune a mudanças na conectividade da rede e é escalável em relação ao tamanho da rede e ao número de nós compondo a rota.

A técnica de roteamento sobre curva pressupõe que exista um mecanismo de geração de trajetórias. No entanto, pelo nosso conhecimento, o problema de geração de curvas não possui nenhuma proposta de solução na literatura. Neste trabalho é proposto um método de geração dinâmica de curvas de roteamento baseado no mapa de energia da rede.

Os resultados de simulação revelaram que, através da política de geração de curvas proposta, é possível evitar que nós-sensores localizados em regiões de baixa energia gastem suas reservas com atividades de retransmissão de pacotes e se concentrem exclusivamente em tarefas de sensoriamento. Dessa forma, particionamentos da rede podem ser significativamente adiados, e o seu tempo de vida, estendido.

As contribuições desse trabalho foram publicadas nos seguintes periódicos: *IEEE Communications Magazine (Ad Hoc and Sensor Networks Series)* [11] e *Journal on Selected Areas in Communication (Special Issue on Autonomics)* [8].





# Sumário

---

---

Agradecimentos . . . . .	iii
Abstract . . . . .	v
Resumo . . . . .	vii
Sumário . . . . .	x
Lista de Figuras . . . . .	xii
Lista de Tabelas . . . . .	1
<b>1 Introdução</b>	<b>3</b>
1.1 Redes de Sensores Sem Fio . . . . .	3
1.1.1 Características . . . . .	4
1.1.2 Componentes . . . . .	5
1.1.3 Aplicações . . . . .	7
1.1.4 Problemas . . . . .	9
1.2 Motivação e Contribuição . . . . .	11
1.3 Organização do Trabalho . . . . .	13
<b>2 Trabalhos Relacionados</b>	<b>15</b>
2.1 Roteamento em Redes de Sensores Sem Fio . . . . .	15
2.1.1 Tipos de roteamento . . . . .	17
2.1.2 Algoritmos baseados em inundação . . . . .	18
2.1.3 Algoritmos baseados em topologia plana . . . . .	19
2.1.4 Algoritmos baseados em topologia hierárquica . . . . .	21
2.1.5 Algoritmos baseados em curva . . . . .	22
2.2 Mapa de Energia . . . . .	27
<b>3 Definição e Análise do Problema</b>	<b>31</b>
3.1 Problema de Geração de Curvas de Roteamento em Redes de Sensores Sem Fio . . . . .	31
3.2 Aplicações . . . . .	32
3.3 Estado da Arte . . . . .	33

3.4	Levantamento de Subproblemas . . . . .	34
3.4.1	Dados de Entrada . . . . .	34
3.4.2	Representação de Trajetórias . . . . .	35
3.4.3	Codificação de Trajetórias . . . . .	39
3.4.4	Ajuste de Curvas . . . . .	39
3.4.5	Arquitetura do Processo . . . . .	42
3.5	Parâmetros de Qualidade . . . . .	42
<b>4</b>	<b>Solução Proposta</b>	<b>45</b>
4.1	Entrada: Mapa de Energia . . . . .	45
4.1.1	Seleção de Pontos . . . . .	46
4.2	Representação e Ajuste de Curvas . . . . .	47
4.3	Arquitetura . . . . .	49
4.3.1	Setorização da Rede . . . . .	50
4.3.2	Seleção do Melhor Conjunto de Curvas . . . . .	51
4.3.3	Broadcast . . . . .	52
4.3.4	Unicast . . . . .	55
4.3.5	Multicast . . . . .	55
4.4	Considerações Finais . . . . .	56
<b>5</b>	<b>Resultados de Simulação</b>	<b>61</b>
5.1	Ambiente de Simulação . . . . .	62
5.2	Métricas . . . . .	64
5.3	<i>Broadcast</i> : Distribuição Uniforme de Energia . . . . .	64
5.4	<i>Broadcast</i> : Evitando Regiões de Baixa Energia . . . . .	68
5.5	<i>Multicast</i> : Disseminação para uma Área-Alvo . . . . .	72
5.6	Conclusões . . . . .	74
<b>6</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>77</b>
	<b>Referências</b>	<b>85</b>

# Lista de Figuras

---

---

1.1	Tamanho de um nó-sensor. . . . .	6
1.2	Mapa de energia de uma rede de sensores sem fio. . . . .	11
2.1	Tipos de roteamento em redes de sensores sem fio. . . . .	17
2.2	Políticas de escolha do próximo nó, sendo $N_0$ o atual. . . . .	25
2.3	Funcionamento básico do TBF. . . . .	26
3.1	Curva de Koch, da esquerda para a direita, quatro iterações e abaixo várias iterações. . . . .	38
3.2	Vizinho mais próximo. . . . .	40
3.3	Interpolação linear. . . . .	40
3.4	Média móvel. . . . .	41
4.1	Processo de geração de curvas. . . . .	49
4.2	Seções cônicas e polinômios de quarto grau sobre um mapa de energia. . . . .	53
4.3	Número máximo de setores depende da posição do nó-monitor. . . . .	53
4.4	Conjuntos de curvas para realizar <i>broadcast</i> . Mapa de energia (35X35) com uma e duas regiões de baixa de energia. . . . .	54
4.5	Conjuntos de curvas para realizar <i>unicast</i> . Mapa de energia (35X35) com uma e duas regiões de baixa de energia. . . . .	55
4.6	Conjuntos de curvas para realizar <i>multicast</i> . Mapa de energia (35X35) com uma e duas regiões de baixa de energia. Área-alvo = (20,20)-(35,35). . . . .	57
5.1	<i>Broadcast</i> : Mapa de energia e cobertura da rede. (T=tempo, C=cobertura, E=energia média) . . . . .	66
5.2	<i>Broadcast</i> : Resultados numéricos. . . . .	67
5.3	<i>Broadcast com região de baixa energia</i> : Mapa de energia e co- bertura da rede (T=tempo, Cd/Cf, Ed/Ef = Cobertura, Energia dentro/fora da região de baixa energia). . . . .	69

5.4	<i>Broadcast</i> com região de baixa energia: Resultados dentro da região de baixa energia. (As curvas correspondentes aos desempenhos do TEDD e do TBF coincidem nas Figuras (a), (c) e (d).) . .	70
5.5	<i>Broadcast</i> com região de baixa energia: Resultados fora da região de baixa energia. . . . .	71
5.6	<i>Multicast</i> : Mapa de energia e cobertura da rede (T=tempo, Cd = Cobertura dentro da área-alvo, Ed= Energia dentro da região de baixa energia, Et = Energia na rede toda). . . . .	73
5.7	<i>Multicast</i> : Resultados numéricos. . . . .	74

# Lista de Tabelas

---

---

4.1	Requisitos computacionais do algoritmo LSQR. . . . .	49
5.1	Consumo de cada componente básico de um nó-sensor. . . . .	62
5.2	Parâmetros utilizados nas simulações. . . . .	64



---

# Introdução

---

**E**ste capítulo tem por objetivo apresentar a área de redes de sensores sem fio. Primeiramente, será feita uma descrição das características e dos principais componentes dessas redes. Em seguida, serão apresentadas algumas áreas de aplicação em que essas redes podem vir a atuar. O assunto será concluído com uma análise dos principais problemas encontrados nessa emergente área de pesquisa. Ainda neste capítulo será apresentado brevemente o problema tratado nesta dissertação, juntamente com a sua motivação. Para concluir, a organização do texto, como um todo, será resumida no final deste capítulo.

## 1.1 *Redes de Sensores Sem Fio*

Na história da computação - *mainframes*, estações de trabalho, computadores pessoais, *PDA*s (*Personal Digital Assistants*), e telefones celulares - os maiores impactos sociais e econômicos se deram por classes novas de computadores, tipicamente menores e mais baratos do que os anteriores, especialmente quando os dispositivos até então independentes se tornaram conectados através de uma rede. Qual será então o próximo passo?

O número de computadores tradicionais, como os citados acima, é vastamente superado pelo número de computadores embutidos que controlam muitos produtos, tais como automóveis, fornos de micro-ondas, câmeras, dispositivos de chão de fábricas e sistemas de monitoramento de saúde. Ultrapassando a tecnologia embutida de hoje, a próxima geração de micro-sensores irá incorporar a tecnologia sem fio, o que irá permitir que um grande número de pequenos dispositivos de baixo preço sejam conectados, formando o que

podemos chamar de “poeira inteligente” (*smart dust*) [22].

Essa tecnologia irá possibilitar novos tipos de computação ubíqua e ambientes inteligentes, tais como sistemas integrados de segurança física, sistemas de monitoramento ambiental e acompanhamento médico contínuo. Apesar dessas aplicações ainda serem mais próximas da ficção científica do que da realidade, essa área representa uma oportunidade única e promissora de pesquisa.

### 1.1.1 Características

Redes de sensores são uma nova família de redes sem fio e apresentam diferenças significativas em relação às redes tradicionais, tais como redes celulares e MANETs (redes ad-hoc). Em redes sem fio tradicionais, as tarefas de organização, roteamento e gerência de mobilidade são feitas com objetivo de otimizar QoS (Qualidade de Serviço) e aumentar a eficiência no uso da banda. Essas redes são projetadas para prover melhor vazão e atraso, dadas as condições de mobilidade da rede. Consumo de energia é de importância secundária, já que as baterias podem ser recarregadas quando necessário. As redes de sensores, no entanto, consistem de centenas de milhares de nós, projetados para operação autônoma. O objetivo principal é prolongar o tempo de vida da rede, prevenindo a degradação de conectividade através de gerência agressiva de energia, já que, tipicamente, as baterias não poderão ser recarregadas devido à operação em ambientes hostis ou remotos.

Redes de sensores sem fio apresentam várias características particulares, que impõem inúmeros desafios operacionais. Algumas dessas características são discutidas a seguir:

- *Endereçamento dos nós.* Em grande parte das aplicações, os nós-sensores não precisam ser endereçados unicamente. Por exemplo, numa aplicação de monitoramento de uma região, o importante não é saber por qual nó foi coletado um determinado dado, mas de que coordenada geográfica ele o foi.
- *Distribuição ad-hoc.* O sistema deve ser capaz de lidar com a distribuição aleatória de nós e formar conexões entre os mesmos.
- *Condições dinâmicas do ambiente.* O sistema deve ser adaptativo a mudanças na conectividade e falhas de nós.
- *Restrições de energia.* Em muitas aplicações, os sensores serão lançados em uma área remota, o que impossibilitará o acesso a esses dispositivos para manutenção. Nesse cenário, o tempo de vida de um sensor dependerá da quantidade de energia que o mesmo possui. Portanto, todas as



operações do sistema, tais como comunicação e processamento de dados, devem levar em consideração o consumo, o modelo de energia e o mapa de energia da rede.

- *Auto-organização da rede.* Devido à dinâmica das condições ambientais, ao grande número de componentes e à possível dificuldade de acesso para manutenção, a configuração de uma rede de sensores sem fio deve ser feita de forma periódica e automática.

Esse cenário impõe várias decisões primárias de arquitetura e resulta em vários compromissos: tamanho pequeno dos nós, baixo custo e operação orientada à economia de energia. O pequeno tamanho limita a capacidade de baterias, tornando necessário que cada operação seja feita de forma eficiente. Além disso, as restrições de energia impõem um limite ao alcance das transmissões de rádio, o que sugere esquemas de transmissões curtas e *multi-hop*. O processamento local de dados também se torna interessante para reduzir a quantidade de dados a serem transmitidos via rádio.

### 1.1.2 Componentes

Os principais componentes das redes de sensores sem fio são: nós-sensores, atuadores, interfaces de comunicação sem fio e um *gateway* para comunicação com entidades externas.

#### *Nós-sensores*

Nós sensores são dispositivos autônomos, equipados com capacidades de sensoriamento, processamento e comunicação. Quando esses nós são dispostos de forma ad-hoc, formam as redes de sensores sem fio. Os nós coletam dados via sensores, processam esses dados localmente, ou através de uma colaboração com alguns de seus vizinhos, e enviam esses dados para uma entidade externa. Assim, cada nó deve desempenhar diversas tarefas, tais como, sensoriamento do ambiente, processamento da informação e atividades ligadas ao tráfego em um esquema de retransmissão *multi-hop*.

Os componentes básicos de um nó-sensor são: transceptor, memória, processador, sensor e bateria. A redução do tamanho do nó (já existem nós medindo apenas poucos centímetros, como na Figura 1.1) tem como consequência a redução do tamanho e capacidade dos seus componentes. Para se ter uma noção dos valores envolvidos, os processadores são geralmente de 8 bits com frequência de 10 MHz, os transceptores têm largura de banda de 1Kbit/s a 1Mbit/s, e a capacidade de memória pode ser de 128 KBytes a 1MByte. Há uma grande diferença entre as tecnologias de fabricação de baterias e, conseqüentemente, do consumo de energia.

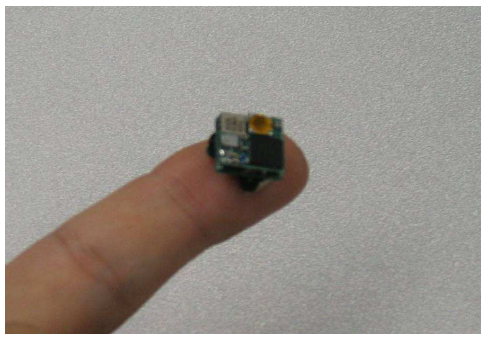


Figura 1.1: Tamanho de um nó-sensor.

### *Atuadores*

Os atuadores têm a função de modificar valores ou características do meio, a fim de corrigir falhas ou controlar o objeto monitorado. As redes que contêm atuadores são de grande interesse em várias áreas, como a Medicina, por exemplo, onde sistemas embutidos podem liberar medicamentos de acordo com as necessidades do paciente.

### *Gateway*

A comunicação da rede de sensores sem fio com outras redes ocorre através de um *gateway*. Os dados coletados percorrem a rede de sensores até chegar a um *gateway* que irá encaminhar os mesmos até o computador do operador da rede através de um elo de satélite ou uma rede como a Internet.

### *Tecnologias de acesso ao meio*

O principal objetivo de um protocolo da camada de acesso ao meio (MAC) é alocar canais de rádio compartilhados entre os nós-sensores de forma mais justa possível e garantir que não ocorram colisões.

Os protocolos de acesso ao meio para redes de sensores sem fio atuais podem ser classificados em dois grupos principais: protocolos baseados em contenção e protocolos baseados em TDMA (*Time Division Multiple Access*).

No grupo baseado em contenção, os mais importantes são: IEEE 802.11 [37], PAMAS [44] e S-MAC [48]. O IEEE 802.11, apesar de ser muito usado devido a sua simplicidade e robustez contra o problema de terminal escondido, não apresenta um consumo eficiente de energia, já que não trata os problemas de escuta redundante e escuta ociosa. PAMAS tenta evitar escuta redundante. S-MAC, uma versão melhorada de PAMAS, reduz ainda mais o consumo de energia tratando o problema de escuta ociosa através do desligamento do rádio de nós inativos. No entanto, a duração do “sono” é igual para todos os nós, o que pode ser injusto para com os nós de menor energia.

Os protocolos baseados em TDMA têm a vantagem natural de não apresentar colisões e sobrecarga de controle de pacotes. No entanto, esse grupo de protocolos não apresenta a mesma escalabilidade que o grupo baseado em contenção. Um exemplo de protocolo TDMA é o proposto em [42], que utiliza FDMA e CDMA para evitar interferência entre elos adjacentes. A desvantagem desse algoritmo é a má utilização de banda, já que cada nó só pode se comunicar com um vizinho de cada vez.

### 1.1.3 Aplicações

Apesar de o enfoque inicial de pesquisas em redes de sensores sem fio ter sido em aplicações militares, tais como vigilância de campos de batalha e localização de tropas inimigas, hoje existe um vasto potencial de aplicações dessa tecnologia em áreas civis. As aplicações nessas áreas podem ser classificadas em quatro grupos principais: monitoramento de *habitat* de animais e plantas, sistemas de previsão e observação do meio ambiente, área de saúde e aplicações comerciais.

#### *Monitoramento de habitat de animais e plantas*

O objetivo dos sistemas de monitoramento de *habitat* de animais e plantas é auxiliar no processo de mapeamento da *biocomplexidade* da Terra. Um exemplo de tal aplicação é o projeto *Great Duck Island*, proposto por pesquisadores de *UC Berkeley/Intel* em 2002 [26]. O projeto visa estudar os padrões de comportamento das gaivotas que habitam a ilha. Trinta e dois nós do tipo *Mica Mote* [17] foram distribuídos na região de interesse, postos dentro de tocas onde as aves habitam. Os dados coletados eram enviados via um *gateway* para uma estação remota que armazenava os dados e os reenviava para uma base de dados em Berkeley por um enlace de satélite a cada 15 minutos. Em [26, 45, 46] são discutidos diversos problemas e soluções para aplicações de monitoramento de *habitat*.

#### *Sistemas de previsão e observação do meio ambiente*

Um sistema de previsão e observação do meio ambiente é um sistema distribuído de grande porte que cobre vastas áreas geográficas, realiza monitoramento, modelagem e previsão de processos físicos, tais como poluição ambiental, inundações, incêndios, etc. CORIE [2] é um protótipo de tal sistema. Ele é composto por treze sensores estacionários, distribuídos pela superfície do rio, e um sensor móvel na beira do rio. Os sensores fixos foram postos em cima de pequenas plataformas ou bóias. Os dados capturados são transmitidos através de um elo sem fio para um servidor centralizado, onde servem de

entrada para um modelo de ambiente físico computacionalmente intenso. A saída do modelo é usada para realizar previsão e guiar (orientar) o transporte de navios. ALERT (*Automated Local Evaluation in Real-Time*) [1] é talvez a mais conhecida rede de sensores sem fio sendo lançada no mundo real. Ela foi desenvolvida nos anos 70 pela agência americana de previsão de tempo para captar informações sobre precipitações (chuvas) e níveis de água de forma a avaliar a possibilidade de inundações. O sistema é composto por sensores meteorológicos e hidrológicos, tais como sensores de nível d'água, de temperatura e de ventos. Os dados são transmitidos via um canal de rádio para uma estação-base, onde são processados pelo modelo de previsão de inundações, que produz alarmes automáticos e possibilita consultas via *Web*.

### *Área de saúde*

Aplicações na área de saúde incluem *tele-monitoramento* de sinais fisiológicos humanos, localização e monitoramento de médicos e pacientes dentro de hospitais, administração de remédios em hospitais, etc. [4]. Em [40] é apresentada uma aplicação biomédica que visa desenvolver uma retina artificial. A prótese é composta por 100 microsensores construídos e implantados dentro do olho humano. Esse mecanismo permite pacientes completamente cegos ou com visão comprometida enxergarem em níveis aceitáveis. A comunicação sem fio é necessária para realizar o controle de ruído, identificação e validação da imagem. Outras aplicações similares envolvem monitoramento de órgãos, nível de glicose e detecção de câncer. A idéia de embutir sensores biomédicos sem fio no corpo humano é bastante promissora, porém ainda enfrenta vários desafios: o sistema deve ser extremamente seguro e confiável, requerer manutenção mínima e se alimentar do calor do corpo humano. O progresso nessa área vai melhorar a qualidade de vida de muitas pessoas que hoje são presas a aparelhos fixos de monitoramento, além de reduzir custos médicos.

### *Aplicações comerciais*

Grandes oportunidades comerciais existem na área de automação de residências e ambientes de trabalho. Um exemplo de aplicação nessa categoria, descrito em [43], é o "*Smart Kindergarden*", que constrói uma rede de sensores sem fio para educação de crianças recém nascidas. Espera-se que o método interativo proposto logo tomará o lugar de técnicas tradicionais baseadas em estímulo e resposta.

Apesar do grande número de aplicações potenciais que as redes de sensores sem fio apresentam, ainda não há aplicações reais dessa tecnologia. A situação dessa área é análoga à Internet há 30 anos atrás. Dado que essa área é altamente orientada à aplicação, os requisitos e as restrições de várias apli-

cações ainda não são completamente compreendidas e, como consequência, a maioria delas ainda não está pronta para o mundo real. As aplicações-protótipo lançadas até agora apresentam algumas características em comum: transmissão de dados não processados através de conexões sem fio, processamento centralizado de dados, esquemas simples de roteamento. Esses protótipos, no entanto, podem servir para identificar os potenciais problemas e verificar os métodos propostos. Com o progresso na fabricação de sensores, pesquisa na área de redes de sensores sem fio, juntamente com intensa cooperação interdisciplinar, pode-se esperar que aplicações reais venham a surgir num futuro próximo. É só uma questão de tempo.

#### *1.1.4 Problemas*

As redes de sensores sem fio têm recebido bastante atenção da comunidade de pesquisa. Os desafios estão em absolutamente todas as áreas envolvidas no processo de construção e operação dessas redes. A seguir, será feita uma lista com os principais problemas sendo estudados.

##### *Construção de nós*

Antes de tudo, é necessário desenvolver e aperfeiçoar as tecnologias de sensoriamento de diversos estímulos, tais temperatura, luz, pressão e suas combinações. Todos os componentes de um nó-sensor, tais como o processador, a memória e o rádio devem apresentar um consumo leve de energia. A fonte de energia é outra parte essencial do processo de construção de um nó-sensor. Por fim, a questão de resistência a meios austeros deve ser pensada com cuidado, dado que muitas aplicações necessitam dos nós operando em ambientes submersos ou com condições climáticas variadas. Exemplos de plataformas reais em fase de desenvolvimento são: *RSC WINS & Hidra*, *UCLA Medusa MK-II*, *Berkeley Mote*, *MIT uAMPS*, entre outros.

##### *Acesso ao meio*

O principal objetivo de um protocolo da camada de acesso ao meio (MAC) é alocar canais de rádio compartilhados entre os nós-sensores de forma mais justa possível e garantir que não ocorram colisões. Em redes de sensores sem fio, atenção especial deve ser dada ao consumo de energia e ao problema de escalabilidade.

##### *Sistemas operacionais*

Projeto de sistemas operacionais simples e leves, que gerenciem o acesso aos componentes dos nós-sensores de forma a gastar o mínimo de energia

possível, é outra área importante de pesquisa. Um exemplo de tal sistema é o *TinyOS* [3], desenvolvido pela *UC Berkeley*.

### *Problemas de tempo e espaço*

Os problemas de sincronização de tempo e localização dos nós também são de fundamental importância para possibilitar o funcionamento da rede. Questões como operação distribuída, custo dos equipamentos e restrições no consumo de energia devem ser levados em consideração para resolver esses problemas.

### *Análise colaborativa de dados*

Os nós devem operar de forma colaborativa e processar os dados coletados do meio usufruindo a soma de suas capacidades individuais. Fusão de dados pode ser realizada dentro da rede para diminuir a quantidade de dados a serem transmitidos para fora da mesma.

### *Auto-configuração*

A rede deve ser capaz de detectar e responder de forma autônoma a mudanças no meio em que se encontra e nos estados de seus componentes.

### *Gerência de topologia*

Deve existir um mecanismo de controle de topologia. É desejável que se possa explorar sobreposições de cobertura de sensoriamento para manter somente um subconjunto de nós ativos em cada período de tempo. Os nós devem ser capazes de ligar e desligar seus componentes, tais como rádio e processador, visando economizar energia.

### *Construção do mapa de energia*

A informação sobre a quantidade de energia restante em cada parte da rede é denominada *mapa de energia* e pode ser utilizada para prolongar o tempo de vida da rede. O mapa de energia de uma rede de sensores sem fio pode ser representado através de uma imagem em tons de cinza, como ilustrado na Figura 1.2. As áreas mais claras representam as regiões com maiores reservas de energia, e as mais escuras, as regiões com pouca energia. Utilizando o mapa de energia, o usuário se torna capaz de determinar se uma área da rede está prestes a sofrer uma falha de sistema devido à falta de energia, por exemplo. Existem diversas maneiras de obter o mapa de energia da rede. Esse assunto é abordado em mais detalhes na Seção 2.2.

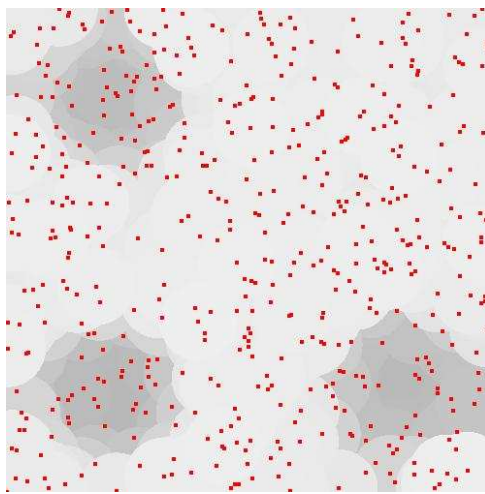


Figura 1.2: Mapa de energia de uma rede de sensores sem fio.

### *Roteamento*

O problema de roteamento em redes de sensores sem fio é freqüentemente denominado “disseminação de dados” e pode ser dividido em vários subproblemas: Como disseminar dados do nó-monitor para parte ou toda a rede? Como rotear as informações coletadas pelos nós-sensores para o nó-coletor? Como realizar a comunicação entre os nós-sensores de forma a possibilitar o processamento colaborativo de dados? Uma análise mais detalhada desses problemas será realizada na Seção 2.1, onde serão apresentados diversos algoritmos propostos na literatura.

Uma técnica de disseminação de dados importante no contexto deste trabalho é a disseminação baseada em trajetórias, ou disseminação sobre curvas. Proposta pela primeira vez em [35], essa abordagem representa um novo paradigma de disseminação de dados em redes ad-hoc compostas por grandes números de nós densamente distribuídos. A inovação está na representação de rotas através de funções contínuas, ao invés de conjuntos de pontos discretos. A idéia principal é embutir uma trajetória no cabeçalho do pacote e deixar que nós intermediários transmitam esse pacotes para nós localizados próximo à trajetória. Devido ao fato da trajetória não conter uma enumeração explícita de identificadores de nós que compõem a rota, essa técnica é imune a mudanças nos estados de nós específicos que compõem a topologia. Além disso, essa abordagem é altamente escalável em relação ao tamanho da rede e ao número de nós compondo a rota.

## *1.2 Motivação e Contribuição*

O objetivo deste trabalho é explorar o problema de roteamento baseado em trajetórias para redes de sensores sem fio. A técnica de disseminação de dados

sobre curvas apresenta várias vantagens no contexto de redes de sensores sem fio, entre as quais se destacam:

- *Comunicação trocada pela computação.* Os caminhos, ao invés de serem “descobertos”, são computados. Essa troca é vantajosa, considerando uma diferença de quase quatro ordens de grandeza entre os custos de envio de um pacote via um canal sem fio e de execução de uma instrução [47].
- *Separação do nome da rota da rota em si.* Essa característica é de grande importância no contexto de uma rede densa, onde nós intermediários podem se mover ou trocar de estado, ao desligar o rádio ou falhar, por exemplo, tornando um caminho discreto completamente inutilizável nesse caso.

Apesar de ser inovadora, a técnica de roteamento sobre curva possui várias questões em aberto. Além de deixar muito a desejar na parte de políticas de propagação (algoritmos executados localmente em cada nó para realizar as decisões de retransmitir ou não um pacote), o problema de geração de curvas não possui nenhuma proposta de solução na literatura.

Neste trabalho, é proposto um método de disseminação de dados sobre curvas, chamado *Trajectory and Energy-based Data Dissemination*, que apresenta duas contribuições principais. A primeira, descrita na dissertação de mestrado [9], é composta por uma nova política de propagação de pacotes. Essa parte do método é descrita em mais detalhes na Seção 2.1.5. A segunda parte do método, o assunto e a contribuição principal desta dissertação, é um algoritmo de geração dinâmica de curvas de roteamento baseado no mapa de energia da rede. Ambas as partes foram propostas em paralelo e de forma complementar.

A idéia principal da técnica de geração de curvas proposta aqui é selecionar um subconjunto de nós da rede que sejam mais indicados para desempenhar o papel de propagadores de pacotes enviados pelo nó-monitor. Em seguida, procura-se determinar o melhor conjunto de curvas que passem por ou perto dos nós selecionados. A escolha do melhor conjunto de curvas pode ser baseada em vários critérios, tais como, quantidade de energia remanescente nos nós-propagadores, a quantidade de nós que os dados transmitidos devem alcançar, ou a localização da área para a qual a disseminação é orientada. Após a geração da(s) curva(s) de roteamento, a(s) mesma(s) pode(m) ser embutida(s) no cabeçalho de um pacote de dados para indicar aos nós intermediários qual rota o pacote deve seguir.

Os resultados de simulação mostram que, através da política de geração de curvas proposta nesse trabalho, é possível evitar que nós de baixa energia



gastem suas reservas com atividades de retransmissão de pacotes. Assim, o papel de roteamento é desempenhado somente pelos nós de maior energia, permitindo que nós de baixa energia se concentrem exclusivamente em tarefas de sensoriamento. Dessa maneira, particionamentos da rede devido a nós que saem de operação pois consumiram totalmente suas baterias podem ser significativamente adiadas, e o tempo de vida dos nós de baixa energia pode ser estendido em até 60%.

### 1.3 Organização do Trabalho

O texto desta dissertação está organizado da seguinte maneira. No Capítulo 2 são apresentados os trabalhos relacionados ao problema abordado aqui. A literatura discutida aborda duas áreas: roteamento em redes de sensores sem fio e obtenção do mapa de energia. No Capítulo 3 é definido o problema abordado neste trabalho - o *Problema de Geração de Curvas de Roteamento para Redes de Sensores Sem Fio* e é feito um levantamento dos subproblemas envolvidos. No Capítulo 4 é apresentada a solução proposta para o problema levantado. No Capítulo 5 são apresentados os resultados experimentais obtidos. E, finalmente, no Capítulo 6, é feita uma análise do trabalho realizado, apresentadas as conclusões e os trabalhos futuros planejados.



---

## Trabalhos Relacionados

---

**N**este capítulo serão discutidos os trabalhos relacionados ao *Problema de Geração de Curvas de Roteamento em Redes de Sensores Sem Fio*. Primeiramente, será feita uma análise dos tipos de roteamento existentes e uma classificação dos algoritmos de roteamento. Em seguida, serão brevemente descritos os mecanismos dos algoritmos de roteamento propostos na literatura. Para concluir, será feita uma análise do problema de obtenção do mapa de energia da rede, seus custos e alternativas de solução.

### 2.1 Roteamento em Redes de Sensores Sem Fio

Algumas características das redes de sensores sem fio tornam o problema de roteamento especialmente desafiador e diferente de redes fixas convencionais, tais como a Internet, ou redes sem fio, tais como redes celulares ou MANETS (redes ad-hoc). Os três principais desafios são:

- *Escala e densidade da rede*. Uma rede de sensores pode ser composta por milhares, ou até mesmo centenas de milhares de nós aleatoriamente distribuídos de forma a cobrir uma vasta área geográfica. Cada nó-sensor é uma potencial fonte de dados, e múltiplos nós-coletores (ou nós-monitores) podem existir. Para aumentar a confiabilidade e o tempo de vida do sistema, a distribuição inicial dos nós-sensores deve ser densa. Apesar de cada sensor possuir recursos limitados, coletivamente vários sensores podem alcançar mais robustez e eficiência. O protocolo pode prolongar o tempo de vida do sistema mantendo um subconjunto pequeno de nós funcionando, enquanto desativando o resto.

- *Recursos limitados.* Para reduzir custos e aumentar a cobertura espacial da rede, nós-sensores são dispositivos tipicamente simples e baratos. Se comparados a computadores pessoais convencionais que compõem as redes ad-hoc ou a Internet, nós-sensores têm energia escassa, processadores lentos, e memórias limitadas. Uma vez esgotada a energia da bateria, o nó pode ser considerado morto, já que a recarga de bateria pode ser difícil, devido ao grande número, à distribuição ad-hoc e localização remota dos nós.
- *Topologia dinâmica.* Mesmo não apresentando, em geral, nós móveis, as redes de sensores sem fio apresentam uma topologia dinâmica. Isso se deve à possibilidade dos nós desligarem periodicamente seus rádios, visando economia de energia. Nós podem sair da topologia devido ao consumo total de suas baterias ou destruição por fatores externos.
- *Falhas imprevistas e canal sem fio não-confiável.* A quantidade limitada de energia e o pequeno tamanho tornam inviável a instalação de rádios e antenas potentes para prover um canal de comunicação de qualidade. Potenciais obstáculos ambientais e interferências comprometem ainda mais a situação, tornando o canal sem fio utilizado pelos nós-sensores sujeito a erros. Em um ambiente inseguro, os próprios nós podem falhar ou serem destruídos inesperadamente.

De uma forma geral, existem três propriedades desejáveis para os protocolos de roteamento:

- *Escalabilidade.* O protocolo de roteamento deve ser escalável para um grande número de nós, aleatória e densamente distribuídos.
- *Uso eficiente de energia.* Já que energia é um recurso crítico, o protocolo deve minimizar a carga de computação e, especialmente, as operações de comunicação.
- *Robustez.* O protocolo deve funcionar em canais de comunicação altamente dinâmicos e sujeitos a erros, assim como em nós que podem falhar devido a interferências externas ou término de energia.
- *Orientação a dados.* Em muitas aplicações não é importante a identificação dos nós participantes do processo de roteamento. O importante são os dados que os mesmos coletam.

As aplicações de sensoriamento geralmente necessitam que cada nó conheça, ao menos aproximadamente, a sua localização. Isso pode ser feito

através de GPS ou alguma outra técnica de localização. O protocolo de roteamento pode explorar essa informação adicional para alcançar uma melhor escalabilidade e eficiência.

### 2.1.1 Tipos de roteamento

Sob o ponto de vista das entidades de comunicação envolvidas, o roteamento em redes de sensores sem fio pode ser classificado em três tipos: transmissão de dados pelos nós-sensores em direção ao nó-monitor, transmissão de dados entre nós-sensores e transmissão de dados pelo nó-monitor em direção a parte ou toda a rede. A Figura 2.1 ilustra os três tipos de roteamento. A transmissão de dados dos nós-sensores para o nó-monitor é usada para enviar os dados sensorizados pela rede para uma aplicação que os requisitou. Neste trabalho, denominamos esse tipo de roteamento de “coleta de dados”. Transmissão de dados entre nós-sensores geralmente ocorre quando algum tipo de cooperação, como, por exemplo, fusão de dados, é necessária. Transmissão de dados partindo do nó-monitor em direção a parte ou toda a rede geralmente é utilizada para enviar uma informação importante para esses nós. Neste trabalho, denominamos esse tipo de roteamento de “disseminação de dados”.

Disseminação eficiente de dados é crucial para o funcionamento de uma rede de sensores sem fio. Através de algoritmos de disseminação, o nó-monitor desempenha várias atividades, tais como, mudar o modo operacional de parte ou toda a rede, disseminar um novo interesse (requisição) para os nós-sensores, ativar ou desativar um ou mais sensores e enviar consultas para a rede.

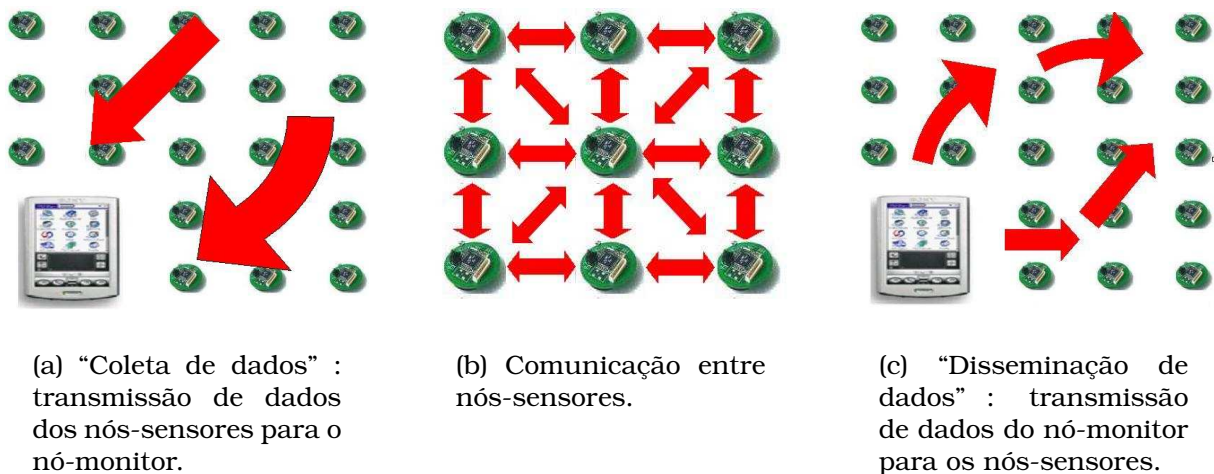


Figura 2.1: Tipos de roteamento em redes de sensores sem fio.

Os protocolos de roteamento propostos na literatura podem ser classificados em quatro grupos principais:

- *Baseados em inundação*. Neste grupo encontramos o *Flooding*, o *Gossiping* [13] e o *SPIN* [15, 23].
- *Baseados em topologia plana*. São protocolos que operam sem construir nenhum tipo de hierarquia entre os nós. Nessa categoria se encaixam *Directed Difusion* [19, 20], *DRP* [6], *GRAB* [51], entre outros [50].
- *Baseados em topologia hierárquica*. Constróem e mantêm uma hierarquia de comunicação entre os nós, visando melhor escalabilidade. Nessa categoria se encaixam *LEACH* [14] e *TTDD* [49].
- *Baseados em curvas*. Utilizam equações de curvas contínuas para especificar as rotas de roteamento. Neste grupo encontramos *TBF* [35] e *TEDD* [8, 10, 11, 12].

Existem vários outros algoritmos não mencionados aqui. Uma lista mais completa é apresentada em [50]. A seguir será feita uma breve descrição de alguns dos algoritmos listados acima.

### 2.1.2 Algoritmos baseados em inundação

Nesta seção serão apresentados alguns protocolos de roteamento que são baseados em inundação (*Flooding*). Inundação é a técnica mais trivial de retransmissão de mensagens, em que cada nó retransmite todas as mensagens que recebe para todos os seus vizinhos. Apesar de serem muito robustos, os protocolos baseados em inundação apresentam vários problemas:

- *Implosão*. Como o nó sempre envia os dados que recebe para os seus vizinhos, sem considerar se o vizinho já recebeu esse dado de outra fonte, o protocolo gasta recursos da rede para enviar várias cópias do mesmo dado para cada nó.
- *Sobreposição*. Nós-sensores freqüentemente cobrem (sensoriam) áreas geográficas sobrepostas e, conseqüentemente, muitas vezes coletam dados sobrepostos. Mais uma vez, os algoritmos baseados em inundação desperdiçam energia e banda ao enviar duas cópias de um mesmo dado para o mesmo nó.
- *Desconsideração de recursos*: Em abordagens de inundação, os nós não modificam suas atividades com base na quantidade de energia disponível em um determinado momento.

A seguir, será descrito o funcionamento dos principais algoritmos desse tipo.

*Gossiping* foi proposto em [13] como um protocolo baseado em “fofoca” para redes *ad-hoc*. O algoritmo é baseado em inundação probabilística, ou seja, cada nó retransmite uma mensagem que recebeu com uma probabilidade  $p$ . *Gossiping* apresenta comportamentos distintos 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 atingidos (cobertos). Por outro lado, em redes densas ou quando a probabilidade de retransmissão for suficientemente alta, o algoritmo apresenta um desempenho bastante satisfatório quanto ao número de nós cobertos. Além disso, para o mesmo número de nós cobertos, o número de transmissões de rádio costuma ser menor que a inundação completa, permitindo uma economia de energia. Resultados de simulação mostram que para configurações de rede consideradas em [13], a probabilidade  $p$  entre 0.6 e 0.8 é suficiente para que praticamente todos os nós sejam cobertos em praticamente todas as disseminações realizadas.

## *SPIN*

*SPIN (Sensor Protocols for Information via Negotiation)* é uma família de protocolos de disseminação de dados para redes de sensores sem fio. Esse é um dos primeiros trabalhos na área e aborda o problema de disseminação eficiente dos dados coletados por um nó-sensor para todos os nós da rede. Dessa maneira, todos os nós são tratados como potenciais nós-coletores. Essa família de protocolos foi projetada com base na inundação clássica, porém melhorando algumas de suas limitações, tais como implosão, sobreposição e descon sideração de recursos. As melhorias são atingidas através de duas inovações básicas: negociação e adaptação direcionada por recursos. Para evitar problemas de implosão e sobreposição, os nós negociam uns com os outros antes de retransmitir os dados. A negociação permite que somente dados úteis sejam retransmitidos. Para possibilitar esse processo, os dados são descritos através de meta-dados. Além disso, cada nó mantém um gerenciador de recursos que permite ao nó recusar a desempenhar certas tarefas, tais como retransmissão de dados de terceiros, quando suas reservas de energia são baixas.

### *2.1.3 Algoritmos baseados em topologia plana*

Nesta seção serão apresentados dois algoritmos que assumem que cada nó da rede possui a mesma funcionalidade, ou seja, que a rede é organizada em uma topologia plana. Os dois algoritmos aqui expostos possuem mais uma característica em comum - eles adotam a abordagem de propagação reversa.

Nessa abordagem, as mensagens de dados seguem o caminho inverso ao percorrido pela mensagem de consulta. O nó-monitor emite uma mensagem de consulta, expressando o seu interesse, possivelmente através de simples inundação. Toda vez que um nó recebe uma mensagem de consulta vinda de um de seus vizinhos, ele estabelece um estado de retransmissão em forma de um vetor, apontando dele para o vizinho, ou seja, indicando a direção inversa. Quando o dado é gerado por uma fonte, o mesmo é transmitido de nó em nó ao longo do vetor até atingir o nó-monitor que emitiu a consulta.

Vale ressaltar que algoritmos que utilizam propagação inversa podem naturalmente integrar em seu funcionamento o processamento de dados dentro da rede. Os dados podem ser agregados em qualquer nó que receba mensagens sobrepostas de diferentes fontes. Tanto o DRP, como a Difusão Direcionada, exploram esse recurso para melhorar sua eficiência. Uma das desvantagens dos algoritmos baseados em propagação inversa é o custo adicional necessário para manter a consistência do estado de roteamento em cada nó num contexto de topologia dinâmica.

## *DRP*

*DRP (Declarative Path Routing)* visa eficiência no uso de energia através de agregações dentro da rede. A abordagem de propagação reversa é utilizada para estabelecer uma árvore de roteamento para cada nó-monitor, de forma que esse possa receber dados sensorizados por toda a rede. Os nós localizados dentro do alcance do raio de comunicação um do outro são organizados em vizinhanças. Quando um nó-monitor emite uma mensagem de consulta, essa mensagem é repassada de vizinhança em vizinhança. A decisão de qual será o próximo nó para o qual a mensagem de consulta deve ser transmitida é feita através da soma ponderada de vários fatores, tais como qualidade do enlace em termos de RSN (Razão Sinal/Ruído), energia remanescente, entre outros. A construção da árvore se completa quando a mensagem de consulta percorre todas as vizinhanças. Quando um nó-sensor começa a gerar dados dentro da sua vizinhança, qualquer vizinho que encontre um casamento entre o dado e o interesse disseminado por algum nó-monitor deve repassar essa mensagem seguindo o caminho inverso da mensagem de consulta. A consistência dos estados de roteamento em cada nó é mantida através de números de seqüência, que são incrementados toda vez que o estado é atualizado. Inconsistências nos estados entre vizinhos são detectadas comparando seus números de seqüência, podendo requisitar a última versão.



Similar ao DRP, a Difusão Direcionada também utiliza a abordagem de propagação reversa e segue o paradigma de orientação a dados. No entanto, a Difusão é mais voltada para problemas de escalabilidade para vários nós-monitores e robustez perante a dinâmica da rede. Ao invés de construir uma árvore de roteamento para cada nó-monitor, a Difusão constrói um grafo compartilhado para entrega de dados de múltiplos sensores para múltiplos monitores. Para melhorar a escalabilidade para vários monitores, os nós não mantêm um estado (vetor) para cada monitor, ou seja, quando um nó recebe uma mensagem de consulta de um vizinho, ele não associa essa mensagem a qualquer nó-monitor, ele apenas armazena um vetor (também chamado de *gradiente*) apontando para o vizinho. Inicialmente, o nó-monitor dissemina a sua consulta através de simples inundação para toda a rede. Ao receber as primeiras respostas a sua consulta, o nó-monitor começa a enviar mensagens de reforço para selecionar o caminho de melhor qualidade. Mensagens de revigoramento positivo são enviadas para o vizinho do qual o dado foi recebido com atraso mínimo (ou outra métrica, dependendo da aplicação). Mensagens de revigoramento negativo podem ser utilizadas para desativar determinados caminhos ineficientes. Gradualmente, os dados começam a fluir somente pelo melhor caminho. Para garantir robustez em caso de falha de nós, os nós-sensores periodicamente enviam mensagens com seus dados para que rotas alternativas possam ser formadas. Vale ressaltar que essa robustez é alcançada a custo de consumo adicional de energia e banda.

#### *2.1.4 Algoritmos baseados em topologia hierárquica*

Até agora nós apresentamos esquemas de roteamento para topologias planas, em que todos os nós possuem a mesma funcionalidade e nenhuma hierarquia de rede é formada durante a disseminação. Nesta seção, serão expostas duas técnicas que constroem hierarquias virtuais visando aumentar a escalabilidade.

##### *LEACH*

*LEACH (Low-Energy Adaptive Clustering Hierarchy)* visa otimizar a organização da rede em termos da energia gasta nas transmissões de rádio. O algoritmo assume que cada nó-sensor seja capaz de ajustar a potência de transmissão do seu rádio, controlando assim a topologia da rede. Se o nó utiliza a potência máxima, o mesmo é capaz de se comunicar diretamente com o nó-monitor. Utilizando uma análise baseada em um dado tamanho de rede e um modelo de rádio, o LEACH dissemina dados através de uma estru-

tura de dois enlaces, visando otimizar o compromisso entre os consumos de energia durante transmissões e recebimentos. Mensagens de dados de cada nó-sensor são primeiramente transmitidas para um líder local e, em seguida, repassadas para o nó-monitor utilizando uma transmissão de rádio de (potencialmente) longo alcance. Periodicamente, o sistema executa um algoritmo distribuído aleatório para eleger um número pré-determinado de líderes. Os líderes escolhidos se anunciam para a rede, de forma a guiar a sua reorganização. Os nós-sensores se subordinam ao líder que possui o sinal de rádio mais forte. O líder então cria um escalonamento TDMA e informa a cada membro do seu grupo em que momento o mesmo pode transmitir seus dados. LEACH também propõe o uso de um esquema CDMA para diminuir a interferência entre os grupos.

### *TTDD*

*TTDD (Two-Tier Data Dissemination)* é uma abordagem baseada em uma hierarquia de dois níveis e visa resolver os problemas de nós-monitores múltiplos ou móveis. Assume-se que cada nó-sensor conhece a própria localização. No entanto, é permitido que os nós-monitores móveis possam não conhecer as suas próprias coordenadas. Uma vez detectado um estímulo, os nós-sensores processam o sinal coletivamente e um deles se torna a fonte que enviará as mensagens com os dados em direção a um nó-monitor. Quando uma fonte começa a gerar dados, ela se prepara para a disseminação construindo uma estrutura de *grid*. Os nós localizados nos pontos de cruzamento do *grid* são os responsáveis por retransmitir os dados. Os nós-monitores disseminam as suas consultas dentro de uma célula local. Ao atingir o nó-retransmissor mais próximo, a consulta é propagada até a fonte através de outros nós-propagadores localizados nos pontos de cruzamento do *grid*. Os dados requisitados seguem o mesmo caminho de volta ao nó-monitor, porém na direção inversa.

### *2.1.5 Algoritmos baseados em curva*

Roteamento baseado em curva é uma técnica que combina roteamento baseado na origem (*source based routing*) [21] e roteamento Cartesiano (*Cartesian routing*) [18], porém adota representação contínua da rota.

Roteamento baseado na origem é uma técnica de roteamento em que o nó que envia o pacote determina a seqüência completa de nós através dos quais o pacote deve passar. O remetente explicitamente enumera essa rota no cabeçalho do pacote, identificando cada passo da propagação pelo endereço do próximo nó para o qual o pacote deve ser transmitido no seu caminho original em direção ao destino. A principal vantagem dessa abordagem é que não há

necessidade de anúncios periódicos de rotas, como em técnicas baseadas em vetor de distância (*distance vector*) e estado do enlace (*link state*), muito utilizadas em redes fixas tradicionais. Isso economiza tanto banda, como energia dos nós compondo a rede, já que esses, além de não ter que transmitir tais mensagens, também não necessitam de manter seus rádios ligados o tempo todo para ouvi-las. Roteamento baseado na origem é utilizado em diversos contextos de roteamento em redes fixas, construindo as rotas tanto estática como dinamicamente. Essa técnica também serviu de base para o *DSR (Dynamic Source Routing)* [21], uma técnica de roteamento projetada para redes móveis ad-hoc.

Roteamento Cartesiano é uma técnica de roteamento em que a rota do pacote é determinada pela posição do roteador em relação a do destino. A principal diferença e vantagem dessa técnica de roteamento em relação aos algoritmos tradicionais baseados em vetor de distância e estado do enlace é que não há necessidade de tabelas de roteamento, pois as comunicações são dependentes da topologia. Assim, tanto o *overhead* da rede como do roteador é reduzido. Por exemplo, uma decisão de roteamento que poderia levar de  $O(\log n)$  a  $O(n)$  utilizando tabelas de roteamento, leva  $O(1)$  com o roteamento Cartesiano. Essa abordagem requer manutenção de um estado mínimo no roteador e visa minimizar a complexidade do roteador, assim como o tempo de resposta do mesmo. Os roteadores mantêm, repassam ou descartam pacotes comparando a sua localização com o endereço do destino do pacote.

O roteamento baseado em curva utiliza o melhor dos dois métodos. Assim como no roteamento baseado na origem, o caminho é indicado pela origem, mas sem enumerar, de fato, todos os nós intermediários. Dessa forma, o roteamento baseado em curva resolve a desvantagem principal do roteamento baseado na origem, que é o maior tamanho do pacote, devido ao tamanho da especificação da rota. Como no roteamento Cartesiano, as decisões tomadas em cada nó são gulosas, mas não são baseadas na distância ao destino, mas sim na distância à curva desejada. A limitação do roteamento Cartesiano, que consiste em uma única política de propagação, em linha reta, é superada, já que, na prática, existem muitos serviços de rede que requerem roteamento que não siga o menor caminho. Um exemplo de tal situação ocorre quando uma rede de sensores sem fio é particionada devido ao uso excessivo de baterias ao longo de menores caminhos mais populares. Outro exemplo são topologias de rede em que roteamento em linha reta não é possível devido à presença de obstáculos, “buracos” de conectividade, ou outras restrições, tais como de segurança. Nesses casos, rotas curvas se tornam úteis para contornar as áreas em que não é desejável ou possível o gasto de recursos com roteamento de terceiros.

Roteamento sobre curva apresenta diversas características que o tornam um candidato ideal para o contexto de redes ad-hoc, tais como redes de sensores sem fio:

- *Comunicação trocada pela computação.* Os caminhos, ao invés de serem “descobertos”, são computados. Essa troca é vantajosa, considerando uma diferença de quase quatro ordens de grandeza entre os custos de envio de um pacote via um canal sem fio e de execução de uma instrução [47].
- *Separação do nome da rota da rota em si.* Essa característica é de grande importância no contexto de uma rede densa, onde nós intermediários podem se mover ou trocar de estado, ao desligar o rádio ou falhar, por exemplo, tornando um caminho discreto completamente inutilizável nesse caso.
- *Especificação da trajetória independente do endereço do destino.* Caso o destino seja conhecido, a técnica pode ser utilizada como suporte para roteamento; em caso contrário, a mesma pode ser utilizada para *multicast* ou *broadcast*.
- *Provisão “barata” de diversidade de rotas.* A provisão é barata, quando comparada a técnicas tradicionais de descoberta de rotas alternativas baseadas em inundação.
- *Utilização de funcionalidades disponíveis nos nós.* Muitas aplicações em redes de sensores sem fio necessitam que os nós sejam capazes de determinar a sua localização. Idealmente, cada nó seria equipado com um GPS, caso em que o nó mais próximo da trajetória transmitiria o pacote. Caso o GPS não seja disponível, o roteamento em curva pode utilizar posições aproximadas, dadas por algoritmos de posicionamento baseados em outras habilidades dos nós, tais como se comunicar com seus vizinhos [34, 39].

Além do simples caso do *unicast*, roteamento em curva tem vantagens significativas em várias outras importantes funções, tais como *broadcast*, descoberta de rotas e caminhos múltiplos.

## TBF

*TBF (Trajectory Based Forwarding)* [35] é o primeiro trabalho que propõe uma abordagem de roteamento sobre curva. A idéia principal do TBF é inserir uma equação de curva (trajetória) no pacote, e cada nó intermediário decide qual será o próximo nó a propagar o pacote com base na distância de seus

vizinhos em relação à trajetória contida no pacote. Para realizar tal decisão, cada nó possui uma tabela de vizinhos. Periodicamente, os nós vizinhos trocam entre si um pacote especial, denominado de *beacon*, para que cada nó atualize sua respectiva tabela. As principais vantagens do TBF são a representação compacta, uma vez que as curvas podem ser descritas utilizando poucos parâmetros, e a independência de nós, uma vez que a trajetória não depende de um nó específico.

No TBF, quando um nó intermediário recebe um pacote, ele escolhe um de seus vizinhos para propagar o pacote recebido. Em [33, 35], Niculescu e Nath fazem algumas sugestões para a escolha do próximo nó da rota.

- *Menor desvio.* O vizinho mais próximo à curva é o escolhido. Na figura 2.2, o nó  $N_2$  seria o escolhido;
- *Mais próximo ao destino.* O vizinho mais próximo ao destino é o escolhido. Na figura 2.2, o nó  $N_4$  seria o escolhido;
- *Escolha aleatória.* Um vizinho é escolhido aleatoriamente;
- *Maior energia.* O vizinho com maior energia é o escolhido.

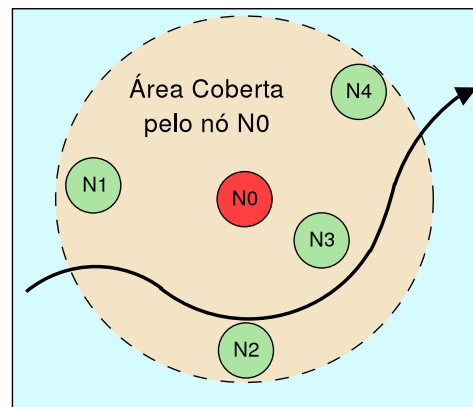


Figura 2.2: Políticas de escolha do próximo nó, sendo  $N_0$  o atual.

A Figura 2.3 ilustra o modo de operação básico do TBF. Quando um nó recebe um pacote de *beacon*, ele atualiza sua tabela de vizinhos (Figura 2.3, ponto B). Se o pacote recebido não for um *beacon*, mas um pacote de dados, o nó verifica se ele é o nó eleito para propagar o pacote recebido (Figura 2.3, ponto C). Se não for, o nó apenas descarta o pacote (Figura 2.3, ponto D). Contudo, se ele for o nó eleito para continuar o processo, ele seleciona o próximo nó da rota (Figura 2.3, ponto E). Essa escolha é baseada na própria tabela de vizinhos do nó corrente e em uma política previamente definida, por exemplo, o vizinho mais próximo do destino ou o vizinho mais próximo da curva. Depois da escolha, o nó transmite o pacote (Figura 2.3, ponto F).

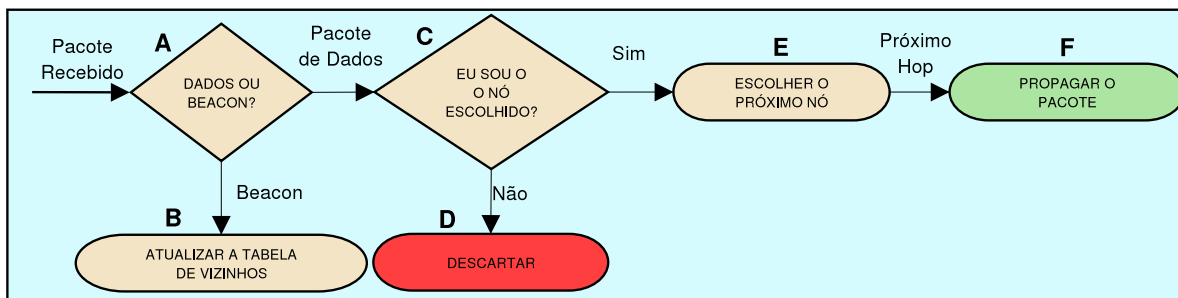


Figura 2.3: Funcionamento básico do TBF.

Apesar das vantagens apresentadas pelo TBF, ele apresenta três problemas. O primeiro é a sobrecarga necessária para a atualização das tabelas de vizinhos. Nesse caso, a troca periódica de *beacons* pelos nós eleva consideravelmente o número de pacotes transmitidos e, conseqüentemente, aumenta o consumo de energia. Em ambientes como as redes de sensores sem fio, em que nós adormecem periodicamente para economizar energia, tal solução tem um custo proibitivo. A segunda desvantagem refere-se a sua fraca tolerância a falhas para situações em que as mudanças na topologia da rede são mais freqüentes que as atualizações das tabelas de vizinho. Nesse caso, quando o nó selecionado está indisponível (por exemplo, o mesmo está dormindo) causa quebras de rota. Por outro lado, é importante destacar que o TBF é mais robusto que os demais protocolos do tipo *source-routing* porque o mesmo elimina as tabelas de rota. Neste ponto, a partir desses dois problemas, observa-se a existência de um compromisso entre a tolerância a falhas e a sobrecarga da atualização das tabelas de vizinho. Finalmente, o último problema do TBF ocorre em processos de comunicação de dados do nó-monitor para a rede (disseminação de dados) em que o TBF não permite que os pacotes contenham mais de uma trajetória. Nem sempre uma única curva pode ser gerada para disseminar informações para o conjunto de nós desejados. O TBF inviabiliza a existência de duas ou mais rotas em um mesmo pacote porque ele escolhe sistematicamente um único nó para continuar o processo de roteamento e, conseqüentemente, uma única curva.

### TEDD

Em [9], é proposta uma nova técnica de roteamento sobre curva. Essa técnica foi desenvolvida em paralelo e de maneira complementar ao presente trabalho, e as duas técnicas juntas constituem o protocolo *Trajectory and Energy-based Data Dissemination* (TEDD). Quando o nó-monitor deseja disseminar uma informação para a rede, ou para uma parte da mesma, ele aciona o módulo de geração de curvas, apresentados no capítulo 4, que recebe o mapa de energia como entrada e gera como saída um conjunto de equações

de curva. Em seguida, o nó-monitor cria um pacote (contendo o conjunto de curvas obtido e a informação a ser disseminada) e o transmite para seus vizinhos. Quando um nó recebe o pacote, ele decide se deve propagá-lo conforme a política de disseminação proposta em [9].

O TEDD estende os princípios do TBF, incorporando o uso do mapa de energia para determinar as “melhores” rotas ou trajetórias, em termos de energia. Além disso, o TEDD soluciona os problemas do TBF (identificados na seção 2.1.5) e, dessa forma, viabiliza o roteamento em curva para as redes de sensores sem fio. Para eliminar o uso de tabelas de vizinhos, o TEDD utiliza uma política de disseminação do tipo *receiver-based* (no TBF, a política é do tipo *sender-based*), ou seja, quem decide se deve propagar um pacote é o nó que o recebeu. Um nó-sensor toma essa decisão baseando-se exclusivamente em sua coordenada geográfica e nas informações contidas no pacote. O processo de decisão é simples: antes de retransmitir um pacote, o nó corrente espera obrigatoriamente um pequeno intervalo de tempo. Se após esse tempo nenhum nó-sensor vizinho tiver retransmitido o pacote, o nó pode retransmiti-lo. A idéia principal dessa técnica está relacionada ao cálculo do tempo de espera que pode ser obtido a partir de algumas políticas. Por exemplo, o tempo de espera pode ser proporcional (ou inversamente proporcional) à distância do nó até a curva, à distância do nó até um determinado ponto mais à frente na curva, ou à energia do nó. Eliminando as tabelas de vizinho, o TEDD dispensa as trocas de *beacons* necessárias para atualizar as tabelas. Conseqüentemente, verifica-se uma redução significativa do consumo de energia na rede. Outra vantagem do TEDD é o aumento da tolerância a falhas, uma vez que os nós intermediários não são escolhidos pelos seus antecessores. Além disso, o TEDD permite a coexistência de mais de uma curva em um mesmo pacote de roteamento. Isso garante uma disseminação de dados mais eficiente, porque partes arbitrárias da rede podem ser alcançadas por um mesmo pacote.

## 2.2 Mapa de Energia

Em redes de sensores sem fio, 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. Utilizando o mapa, é possível determinar se alguma parte da rede pode sofrer falhas devido à falta de energia [52]. Várias aplicações para as redes de sensores sem fio podem utilizar a informação fornecida pelo mapa, como algoritmos de disseminação de dados, de re-configuração, de fusão de dados, ou de gerenciamento da rede. O ponto importante é que o mapa de energia é fundamental para obter

controle sobre o consumo de energia em diferentes partes da rede e prolongar o tempo de vida da rede.

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 nó-monitor. Essa abordagem apresenta um custo elevado em termos de energia, devido ao grande fluxo de dados necessário 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.

Em [52], é proposta uma abordagem mais interessante em que o mapa é construído utilizando técnicas de agregação. Um nó-sensor apenas precisa enviar para o nó-monitor sua energia local quando existe uma queda significativa, comparando-se com a última vez que o nó reportou sua energia disponível. Ao longo do caminho para o nó-monitor, 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 ou 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. Em [52], 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.

Em [28, 29, 30, 31], é proposta uma abordagem eficiente baseada em predição. Cadeias de Markov são utilizadas para prever o consumo de energia de um nó-sensor e com essa informação construir o mapa de energia. Essa abordagem tira vantagem da existência de 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-monitor pode atualizar localmente a informação de energia de todos os nós da rede. Resultados de simulação apresentados em [29] mostram que o uso de modelos baseados em predição apresentam um bom desempenho e diminuem a quantidade de energia necessária para a construção do mapa. 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 [29] e em [7] é a utilização de técnicas de amostragem para reduzir ainda mais o custo de construção do mapa. Finalizando, é importante destacar que além de existi-



rem técnicas otimizadas para a construção do mapa, o custo de construção do mesmo pode ser dividido por todas as aplicações e/ou algoritmos que se beneficiam do mesmo.

A principal diferença entre a abordagem baseada em agregação e a baseada em predição é que, na primeira, cada nó envia para o nó-monitor apenas sua energia disponível. Na segunda, são enviados também os parâmetros de um modelo que descreve o consumo de energia no futuro. Nesse caso, cada nó envia para o nó-monitor sua energia disponível e também os parâmetros do modelo escolhido para representar seu consumo de energia. Com esses parâmetros, o nó-monitor pode atualizar localmente a energia disponível em cada nó da rede e, assim, diminuir a quantidade de pacotes tramitando na rede em função da construção do mapa de energia.



---

## Definição e Análise do Problema

---

Neste capítulo será definido e analisado o problema-alvo desta dissertação - o *Problema de Geração de Curvas de Roteamento em Redes de Sensores Sem Fio*. Primeiramente, será feita uma descrição geral do problema, juntamente com suas aplicações e estado da arte. Em seguida, serão analisados os subproblemas envolvidos. Para finalizar, serão apresentados os parâmetros que podem ser utilizados para avaliar a qualidade de uma solução para o problema estudado.

### 3.1 *Problema de Geração de Curvas de Roteamento em Redes de Sensores Sem Fio*

Dado um algoritmo de propagação de dados sobre curvas,  $A$ , consideremos uma rede de sensores sem fio,  $R$ , composta por uma superfície geográfica a ser sensoriada,  $S$ ; um conjunto de eventos, ou estímulos,  $E$ , que ocorrem dentro dos limites de  $S$ ; um conjunto de nós-sensores,  $N$ , representados por diversas propriedades, tais como, um conjunto de coordenadas,  $CN$ , um conjunto de energias,  $EN$ , um conjunto de raios de transmissão,  $RN$ , entre outros. Consideremos, ainda, um número  $M$  de trajetórias contínuas de um determinado tipo,  $T$ , com um determinado número de parâmetros,  $NPT$ . O *Problema de Geração de Curvas de Roteamento em Redes de Sensores Sem Fio* consiste em determinar, para cada uma das  $M$  curvas, os valores dos  $NPT$  parâmetros, de forma a atingir ou maximizar os objetivos do algoritmo de roteamento  $A$ .

Os objetivos do algoritmo de roteamento podem ser os mais diversos, como, por exemplo:

- Enviar uma informação para o maior número possível de nós;

- Enviar uma informação para uma determinada região da rede;
- Enviar uma informação para um ou mais nós, de forma que somente nós de maior energia sirvam de intermediários;
- Enviar uma informação para uma parte da rede, porém contornando regiões que apresentam maior vulnerabilidade a ataques externos.

O mecanismo de geração de curvas não pode funcionar de forma estática, nem com base em intervenção externa. Isso se deve à natureza das redes de sensores sem fio. Além de apresentar uma topologia tipicamente dinâmica, em que nós podem falhar ou acordar de um estado de dormência, a distribuição de energia pode mudar ao longo do tempo, devido a ocorrências não uniformemente distribuídas de eventos, ou utilização excessiva de certos nós para propagação de pacotes de terceiros, ou outras tarefas ligadas ao gerenciamento da rede. Assim, se torna necessário um mecanismo capaz de refletir e se adaptar a essas mudanças. Essa adaptação deve ser automática e se basear somente em informações de que os nós dispõem, já que a intervenção de uma entidade administrativa externa geralmente é considerada inviável e a rede deve funcionar de forma autônoma.

Uma das questões mais importantes a serem consideradas é se o mecanismo de geração de curvas deve operar de forma centralizada ou distribuída. O principal problema em um esquema distribuído é a estratégia para selecionar os pontos pelos quais a trajetória deve passar. Modelando o problema de forma a considerar todos os parâmetros que descrevem a configuração da rede em um determinado momento, infelizmente, pode requerer uma quantidade enorme de informações globais, impondo um custo de comunicação inaceitável. A decisão entre uma abordagem centralizada e distribuída, portanto, é crítica. De um lado, uma solução centralizada pode alcançar resultados próximos do ótimo através da exploração de informações globais, troca de comunicação por computação através da declaração de rotas, ao invés da descoberta das mesmas. Isso parece altamente desejável, se considerarmos as quatro ordens de grandeza de diferença entre os custos de envio de um pacote e de executar uma instrução. Por outro lado, uma solução distribuída não requer fornecimento de informações globais, ganha em flexibilidade e robustez, apesar de ser intrinsecamente “cega”, devido a limitações de uma visão local dos nós que tomam as decisões sobre a direção que o pacote deve seguir.

## 3.2 Aplicações

A aplicação mais direta de um mecanismo de geração de curvas é em algoritmos de roteamento em curvas, por exemplo o TBF e o TEDD, descritos na

Seção 2.1.5. A idéia básica é embutir os parâmetros que descrevem a trajetória contínua no cabeçalho do pacote e deixar que os nós tomem a decisão de propagá-lo ou não com base na sua posição relativa à equação da curva. Nesse caso, cada nó deve ter consciência de sua localização e ter a capacidade de calcular a sua distância em relação à curva. A distância não precisa ser calculada de forma precisa, já que isso pode envolver computações excessivamente complexas, mas calculada de forma aproximada. Um algoritmo aproximado e de funcionamento bastante simples, que calcula a distância de um ponto a um polinômio ou seção cônica é proposto em [9].

Além de ser essencial para algoritmos de roteamento sobre curvas, o mecanismo de geração de curvas pode ser utilizado por algoritmos de roteamento que adotam a abordagem de propagação reversa, como o DRP ou a Difusão Direcionada, descritos na Seção 2.1.3. Nesses algoritmos, as mensagens de dados sensoriados seguem o caminho inverso ao percorrido pela mensagem de consulta. A mensagem de consulta emitida pelo nó-monitor percorre a rede até atingir o nó-sensor que possui a informação requisitada através de técnicas pouco eficientes, tais como simples inundação. Durante essa fase, o caminho seguido pelas mensagens de consulta é armazenado pelos nós intermediários, de forma a poder ser utilizado pelo fluxo de mensagens de dados em direção oposta. O mecanismo de geração de curvas poderia ser usado para substituir a propagação baseada em inundação. Assim, a mensagem de consulta poderia ser enviada através de um algoritmo de roteamento em curva, e o caminho percorrido pela mesma poderia ser armazenado pelos nós intermediários da mesma forma que nas técnicas de propagação reversa. Para que as curvas de roteamento possam ser geradas, o nó-gerador das mesmas poderia usar informações sobre a direção ou a localização geográfica aproximada dos dados sensoriados, ou até mesmo realizar um *broadcast* baseado em curvas.

### 3.3 Estado da Arte

Em [35], os autores do TBF discutem técnicas de especificação e de codificação de trajetórias, mas não abordam o problema de geração das mesmas, ou seja, não se menciona como, nem baseado em que tipo de informação as trajetórias devem ser geradas. Simplesmente assume-se que as curvas são dadas *a priori*. Em [8, 9, 10, 11, 12], as curvas são geradas pelo método proposto nesta dissertação. O funcionamento desse método será descrito no Capítulo 4. No roteamento Cartesiano (discutido na Seção 2.1.5), a rota do pacote é calculada como a linha reta entre a posição do roteador e o destino final do pacote. As demais técnicas costumam trabalhar com rotas discretas, ou seja, especificadas através da enumeração dos nós que compõem a rota.

Assim, conclui-se que a solução proposta nesta dissertação é uma tentativa pioneira de resolver o *Problema de Geração de Curvas de Roteamento em Redes de Sensores Sem Fio*. Na seção seguinte, serão levantados e analisados os subproblemas que compõem esse problema.

### 3.4 Levantamento de Subproblemas

Nesta seção serão levantados e analisados os subproblemas que compõem o *Problema de Geração de Curvas de Roteamento em Redes de Sensores Sem Fio*.

#### 3.4.1 Dados de Entrada

A primeira pergunta que surge ao se tentar resolver o *Problema de Geração de Curvas de Roteamento em Redes de Sensores Sem Fio* é em que tipo de informação esse processo deve se basear. Informação sobre o estado da rede, as características da área sensoriada e os objetivos do algoritmo de roteamento podem ser utilizados nos cálculos. Os critérios podem ser os mais diversos possíveis, tais como:

- Relevância da área sensoriada e presença de obstáculos;
- Distribuição de eventos dentro da área sensoriada;
- Distribuição geográfica dos nós-sensores;
- Quantidade de energia contida nas baterias dos nós;
- Cobertura de sensoriamento;
- Condições de conectividade;
- Diferenças nas características físicas dos nós (redes hierárquicas);
- Área-alvo da disseminação ou coleta de dados.

Um fator de importância fundamental é o custo para obter essas informações, em termos de número e frequência de transmissões de dados necessárias. Quando se trata de uma abordagem centralizada, as informações devem ser coletadas de todos os nós da rede. A viabilidade de obtenção dessas informações deve ser cuidadosamente estudada. Quando se trata de uma solução distribuída, a decisão pode ser feita com base em dados locais ou, no máximo, dados dos nós vizinhos. Nesse caso, as decisões ficam limitadas pela visão local.

### 3.4.2 Representação de Trajetórias

Tendo definido o tipo de informação em que o processo de geração de curvas deve se basear, deve ser definido como as trajetórias serão representadas. Em [35], várias formas são discutidas, tais como, funções, equações ou representação paramétrica. Nesta seção serão levantadas e analisadas diversas formas de representação de curvas contínuas.

#### *Funcional*

No plano Cartesiano, uma representação explícita de uma curva planar é dada por

$$y = f(x), \quad (3.1)$$

onde  $f(x)$  é uma determinada função de  $x$ .

Uma representação explícita de curva permite computar diretamente o valor de  $y$ , para qualquer valor de  $x$ . Para representar explicitamente uma linha reta em um sistema de coordenadas Cartesianas, devemos fazer:

$$y = kx + b, \quad (3.2)$$

dado que a linha não seja vertical em relação ao eixo  $x$ . Para representar uma linha vertical, a função necessária é

$$x = c, \quad (3.3)$$

onde  $c$  é uma constante.

A vantagem da representação funcional é a facilidade com que o valor da variável dependente  $y$  pode ser calculado para qualquer ponto do domínio da variável independente  $x$ . A desvantagem é que curvas que possuem valores múltiplos de  $y$  para um mesmo valor de  $x$ , como, por exemplo, no caso de um círculo ou uma reta vertical, não podem ser representadas e causam instabilidade numérica nos métodos computacionais.

#### *Equacional*

A representação explícita é satisfatória quando a função é injetora (quaisquer elementos distintos do domínio apresentam imagens distintas no contra-domínio) e a curva não apresenta tangentes verticais. No entanto, essa técnica impossibilita a representação de muitas curvas importantes na prática, tais como círculos, elipses e outras seções cônicas. Uma equação implícita, do tipo

$$f(x, y) = 0, \quad (3.4)$$

evita muitas dificuldades de imagens múltiplas e tangentes verticais, inerentes à forma explícita.

Um círculo, cujo centro se encontra na origem, por exemplo, pode ser representado pela seguinte equação:

$$x^2 + y^2 - 1 = 0, \quad (3.5)$$

Para representar a mesma curva através da forma explícita, o círculo deveria ser dividido em dois segmentos, com

$$y = +\sqrt{(r^2 - x^2)}, \quad (3.6)$$

para a metade superior e

$$y = -\sqrt{(r^2 - x^2)}, \quad (3.7)$$

para a metade inferior.

A vantagem da representação equacional é o seu maior poder de representação, que engloba curvas, tais como seções cônicas, e a sua maior estabilidade computacional. Além disso, esse método é tão compacto quanto a representação funcional, em termos de número de parâmetros necessários para codificar uma curva. A desvantagem dessa técnica é que se torna necessário resolver explicitamente a equação para poder determinar pontos pertencentes à curva. No entanto, como proposto em [9], algoritmos aproximados bastante simples podem ser utilizados para isso.

### *Paramétrica*

Curvas definidas de forma explícita ou implícita são às vezes chamadas de curvas não-paramétricas. Uma maneira alternativa de descrever curvas é a forma paramétrica, que utiliza um parâmetro auxiliar para representar a posição de um ponto. Por exemplo, um círculo unitário com centro na origem poderia ser representado da seguinte forma:

$$x = r\cos(u) \quad y = r\sin(u) \quad (3.8)$$

onde  $u \in [0, 2\pi]$  é um parâmetro angular e  $r$  é o raio do círculo.

Um ponto numa curva paramétrica é especificado por um único valor do parâmetro. Portanto, a forma paramétrica é independente de qualquer sistema de coordenadas. Assim, a descrição paramétrica da curva permite que transformações de coordenadas, tais como, translação e rotação, muito utilizadas em aplicações de computação gráfica, sejam realizadas facilmente.

As vantagens da forma paramétrica são sua simplicidade e poder de representação. Além disso, o cálculo de pontos na curva é feito diretamente. Essa



é a técnica preferida em aplicações de computação gráfica. A desvantagem dessa técnica é a dificuldade de obter modelos genéricos, dado que as curvas paramétricas constituem um conjunto mais geral de curvas, do que os conjuntos de polinômios ou seções cônicas, por exemplo. Uma maneira de contornar esse problema é a representação por partes, em que um modelo diferente é aplicado a cada segmento da curva. A dificuldade dessa abordagem é definir um modelo para cada segmento. Além disso, o número de parâmetros para representar a curva inteira cresce, tornando a técnica menos atraente para aplicações em redes de sensores sem fio, onde uma grande quantidade de dados necessários para codificar a trajetória implica em um tamanho inaceitável do pacote a ser transmitido.

### *Splines*

Um spline é um tipo de curva paramétrica, composta por seções de polinômios, satisfazendo certas condições de continuidade nos pontos de fronteira entre os segmentos. A condição de continuidade pode ser de primeira ordem, caso em que as curvas consecutivas compartilhem a mesma tangente (primeira derivada) no ponto de encontro, ou de segunda ordem, caso em que a “velocidade” (segunda derivada) seja a mesma antes e depois do ponto de fronteira. Essa característica faz com que uma seqüência de vários polinômios tenha aparência de uma curva contínua única. Para garantir as condições de continuidade, os polinômios devem ter grau, no mínimo, três. Existem vários tipos de splines, sendo os tipos mais comuns os splines Bézier e B-splines [41].

A grande vantagem da representação por splines é a sua flexibilidade e poder de expressão. A desvantagem dos splines no contexto de roteamento sobre curva é, por ser um tipo de curva paramétrica, possuir um grande número de parâmetros necessários para especificar cada segmento de curva. Splines são muito utilizados em aplicações de computação gráfica, onde a precisão é mais importante do que o número de parâmetros.

### *Fractais*

Os fractais foram nomeados no início dos anos 80 por Benoît Mandelbrot, o “pai dos fractais”, para classificar certos objetos intrincados que não possuem dimensão inteira (1, 2 ou 3) mas sim fracionária (dimensão 1,85 por exemplo). Os fractais podem apresentar uma infinidade de formas diferentes, não existindo uma aparência consensual. Contudo, existem duas características muito freqüentes nesta geometria: auto-semelhança e complexidade infinita. Distante do rigor e do formalismo matemático, pode-se definir fractais como: “objetos que apresentam auto-semelhança e complexidade infinita, ou seja, têm sempre cópias aproximadas de si mesmo em seu interior.”

Helge von Koch (1870 - 1924), matemático suíço, introduziu a Curva de Koch. A Curva de Koch é uma forma fractal clássica, simples de ser entendida. Partindo de um triângulo equilátero, divide-se cada lado em três segmentos. Os segmentos intermediários são então substituídos por dois segmentos semelhantes que vêm a formar os lados de um triângulo equilátero menor. Isto resulta numa figura na forma de uma estrela com 12 lados (seis pontas). Realizando o mesmo processo em cada um dos 12 lados, e assim sucessivamente, obtém-se uma figura em evolução constante, que lembra um floco de neve. A Figura 3.1 ilustra a Curva de Koch.

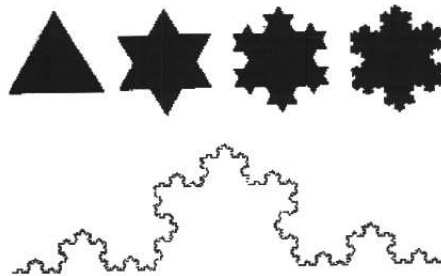


Figura 3.1: Curva de Koch, da esquerda para a direita, quatro iterações e abaixo várias iterações.

As vantagens da representação de curvas através de fractais é a precisão que pode ser atingida na representação de curvas ou superfícies irregulares. Os fractais, assim como os splines, são muito utilizados em computação gráfica. No entanto, há grande dificuldade em controlar o comportamento de um fractal. O processo de geração de um fractal a partir de um conjunto de pontos exige funções complexas de interpolação e geralmente é feito de segmento em segmento, como no processo de interpolação de splines.

### *Trajetórias Compostas*

Trajetoórias compostas podem ter diversos componentes, ou a trajetória pode ser especificada como uma combinação de vários componentes simples, tais como, componentes de Fourier. Quanto mais componentes de Fourier são especificados no pacote, melhor a precisão da trajetória. Existe um interessante compromisso entre a precisão da curva e a quantidade de parâmetros necessários para representá-la.

Trajetoórias compostas podem ter utilidade em disseminações do tipo *multicast*, em que uma mesma informação pode ser enviada para regiões distintas da rede através de uma trajetória complexa, que pode ter segmentos de reta combinados com segmentos de seções cônicas, por exemplo.

A vantagem das trajetetoórias compostas é a sua flexibilidade. No entanto, deve-se tomar cuidado para que o número de componentes seja pequeno, da-

das as restrições de quantidade de dados que deve ser transmitida no cabeçalho do pacote.

### 3.4.3 Codificação de Trajetórias

Tendo definido a(s) forma(s) de representação das curvas, juntamente com o número e o tipo de parâmetros necessários para isso, é importante definir como esses parâmetros serão codificados e inseridos nos cabeçalhos dos pacotes de dados.

Uma abordagem seria a representação por vetor, onde cada nó saberia como interpretar cada posição do vetor, dado um conjunto fixo de tipos de trajetórias. O vetor poderia conter uma tupla da seguinte forma:

$$t = (\text{tipoDaCurva}, n, p_1, p_2, \dots, p_n) \quad (3.9)$$

onde *tipoDaCurva* é o tipo da curva, por exemplo, um polinômio; *n* é o número de parâmetros, e  $p_1, \dots, p_n$  seriam os parâmetros em si.

Vale a pena ressaltar que, dependendo da precisão, o número de bits necessários para representar cada parâmetro pode variar. Assim, ao diminuir a precisão dos valores dos parâmetros, a quantidade de dados transmitidos é diminuída.

### 3.4.4 Ajuste de Curvas

Interpolação é o procedimento de estimação de valores de propriedades em pontos ou áreas não medidos, utilizando um conjunto limitado de observações obtidas através de medição. A interpolação é chamada de global caso uma única função/equação de curva seja determinada. Interpolação global é possível quando se tem disponível um modelo para representar o tipo de curva a ser estimada, por exemplo, um polinômio ou uma seção cônica. A interpolação é chamada de local no caso em que funções/equações diferentes sejam adotadas localmente e repetidamente em porções pequenas da área total. Exemplos de interpolação local são interpolações de curvas paramétricas, tais como splines, ou fractais. Quando a curva ajustada passa por todos os pontos disponíveis, a interpolação é exata, caso contrário, a mesma é aproximada.

Existem três métodos básicos de interpolação exata:

- *Vizinho mais próximo.* O mesmo valor do ponto observado é atribuído aos pontos vizinhos, como mostrado na Figura 3.2.
- *Interpolação linear.* Uma função linear é atribuída a intervalos entre pontos adjacentes, como mostrado na Figura 3.3.

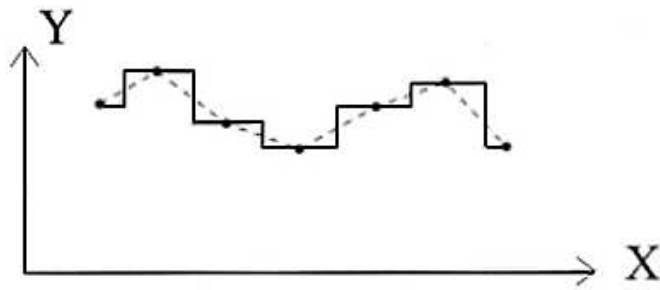


Figura 3.2: Vizinho mais próximo.

$$y = (x - x_1)(y_{i+1} - y_i)/(x_{i+1} - x_i) + y_i, x_i \leq x \leq x_{i+1} \quad (3.10)$$

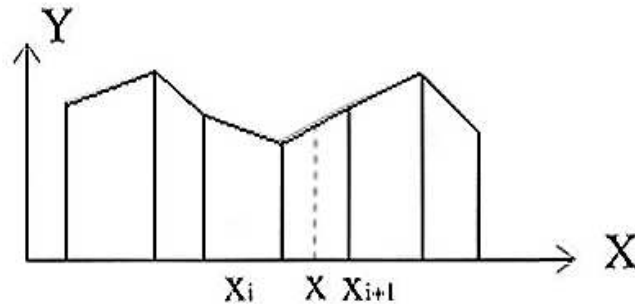


Figura 3.3: Interpolação linear.

- *Interpolação exata por spline cúbico.* Um polinômio de terceiro grau é aplicado entre pontos adjacentes, sob a condição de que a primeira e a segunda derivadas sejam contínuas.

Existem três métodos básicos de interpolação aproximada:

- *Média móvel.* Um intervalo de  $-d$  a  $+d$  recebe o valor da média das observações na região, como mostrado na Figura 3.4.
- *Interpolação aproximada por spline.* Uma curva cúbica é determinada utilizando quatro observações adjacentes.
- *Mínimos quadrados.* Uma técnica de ajuste de curvas cuja base é a minimização da soma dos quadrados das distâncias entre os pontos de observação e a curva.

A técnica de mínimos quadrados merece uma atenção especial, devido à diversidade de modelos a que pode ser aplicada. Essa é a técnica que será utilizada para realizar o ajuste de curvas neste trabalho e, portanto, será apresentada em mais detalhes a seguir.

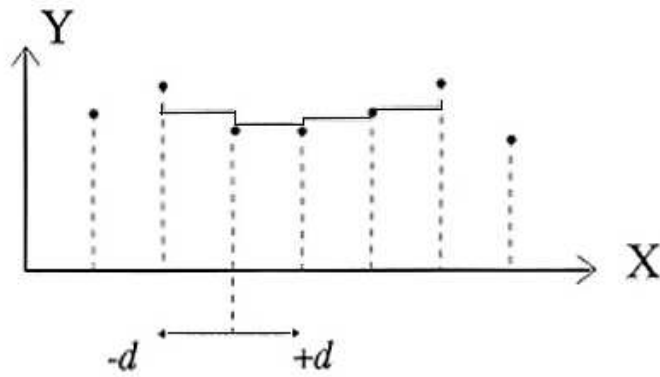


Figura 3.4: Média móvel.

### Mínimos Quadrados

Aproximação por mínimos quadrados, também chamada de regressão linear [32], tenta modelar o relacionamento entre uma ou mais variáveis explicativas e uma variável de resposta, através do ajuste de uma equação linear a um conjunto de dados obtidos através de observação. Formalmente, dadas  $n$  observações e  $p$  parâmetros estimados, o modelo de regressão linear múltipla pode ser representado da seguinte forma:

$$Y = X\beta + \epsilon, \quad (3.11)$$

onde  $Y$  é um vetor de tamanho  $n$ ;  $X$  é uma matriz  $n \times p$ ;  $\beta$  é um vetor de tamanho  $p$  com os parâmetros estimados; e  $\epsilon$  é um vetor de tamanho  $n$  de erros.

Utilizando o princípio de mínimos quadrados, o objetivo é minimizar

$$(Y - X\beta)'(Y - X\beta) \quad (3.12)$$

Tomando a derivada em relação a  $\beta$  e igualando a zero, obtemos o seguinte sistema de equações normais:

$$(X'X)\beta = X'Y, \quad (3.13)$$

onde o vetor estimado

$$\beta = (X'X)^{-1}X'Y \quad (3.14)$$

segue da solução do sistema de equações normais.

### 3.4.5 Arquitetura do Processo

Tendo definido o tipo de dados que servirá de entrada para o processo de geração das curvas de roteamento, a forma de representação e codificação das trajetórias e a técnica de ajuste de curvas, algumas questões ligadas à arquitetura do processo ainda devem ser resolvidas. Entre elas:

- *Qual o número ideal de curvas?* A utilização de mais de uma curva em um mesmo pacote é uma técnica interessante, se considerarmos aplicações de *broadcast* e *multicast*. Outra aplicação para múltiplas trajetórias é o fornecimento de caminhos alternativos para casos de falhas ou mudanças na configuração da rede (alguns nós podem entrar em estados de “dormência” e causar perdas de pacotes). O número de curvas necessárias para atingir um determinado objetivo deve ser cuidadosamente definido, de acordo com os requisitos da aplicação.
- *Como devem ser distribuídas?* Tendo definido o número ideal de curvas, é necessário determinar como as mesmas devem ser distribuídas, se podem ou devem possuir interseções, etc.
- *Com que freqüência devem ser distribuídas?* Outro ponto importante é a freqüência com que as curvas devem ser renovadas e disseminadas para a rede. Fatores tais como mudanças no estado da rede, mudanças nos objetivos da aplicação, ou restrições de capacidade de processamento dos nós que irão hospedar o mecanismo de geração das curvas podem ser utilizados para resolver esse problema.

## 3.5 Parâmetros de Qualidade

Antes de propor uma solução para o *Problema de Geração de Curvas de Roteamento em Redes de Sensores Sem Fio*, é importante definir parâmetros que podem ser usados para medir a qualidade da mesma. O conjunto de curvas de roteamento pode variar de acordo com os requisitos da aplicação. O objetivo de uma disseminação de dados pode ser atingir o maior número possível de nós (*broadcast*), poupar os nós que possuem baixas reservas de energia, maximizar o número de *hops* entre a origem e o destino, ou evitar certas regiões por motivos de segurança, entre outros. Alguns critérios podem, no entanto, ser definidos como os mais comuns para a maioria das aplicações, tais como:

- *Geração dinâmica.* As curvas devem ser geradas dinamicamente, com base no estado de rede mais atual de que se dispõe.

- *Cobertura.* Um dos critérios mais importantes é o número de nós atingidos por uma disseminação. Esse requisito pode surgir tanto no contexto de um *broadcast*, como no contexto de uma disseminação *multicast*. O objetivo de uma disseminação pode ser enviar uma determinada informação para nós contidos em uma determinada região geográfica. Nesse caso, não só é importante que o número de nós atingidos seja alto dentro da área-alvo da disseminação, mas também que o número de nós atingidos seja baixo fora da área-alvo.
- *Razão de pacotes recebidos por pacotes enviados ( $R_x/T_x$ ).* Não é razoável maximizar o número de pacotes recebidos se o número de pacotes transmitidos também é maximizado. O extremo dessa situação é a disseminação por inundação, onde, para cada pacote recebido, um pacote é enviado. É importante minimizar o número de transmissões necessárias para atingir um número grande de nós.
- *Descontinuidades (quebras de trajetória).* Outro fator importante é que o pacote não seja perdido ao longo da sua trajetória por causa de “buracos” na rede, ou seja, os nós pelos quais passa a curva de roteamento devem estar dentro do raio de transmissão um do outro. Em caso contrário, o pacote se perde no meio do caminho.
- *Quantidade de parâmetros transmitidos.* O número de parâmetros necessários para codificar as trajetórias deve ser controlado, já que cada pacote transmitido terá que conter no seu cabeçalho a(s) sua(s) curva(s) de roteamento. E, quanto maior o número de parâmetros, maior a sobrecarga de dados em cada pacote transmitido e, conseqüentemente, maior a quantidade de energia necessária para transmiti-los.
- *Custo computacional.* Caso o algoritmo de geração de curvas possa ser executado somente em nós-monitores, o mesmo pode utilizar métodos computacionalmente complexos. No entanto, a complexidade do algoritmo deve ser compatível com o tempo de resposta exigido pela aplicação. Caso as curvas devam ser geradas de forma distribuída pelos nós-sensores, o algoritmo deve ser o mais simples e computacionalmente leve possível, devido a restrições de memória, capacidade de processamento e quantidade de energia dos nós-sensores.

O mecanismo de geração de curvas de roteamento deve ser capaz de priorizar um ou vários objetivos, possivelmente optando por um compromisso entre os mesmos, de acordo com os requisitos da aplicação.

Com isso pode-se finalizar a análise do *Problema de Geração de Curvas de Roteamento em Redes de Sensores Sem Fio*. No próximo capítulo, será

apresentada a solução proposta e defendida nesta dissertação.



---

## Solução Proposta

---

**O** *Problema de Geração de Curvas de Roteamento em Redes de Sensores Sem Fio* é um problema que, pelo nosso conhecimento, até o presente momento não teve nenhuma proposta de solução na literatura. Portanto, a solução proposta nesta dissertação é uma tentativa pioneira de resolvê-lo. O problema consiste de várias etapas, como mostrado no capítulo anterior. Primeiramente, devem ser definidas as entradas para o processo. Esse assunto vai ser abordado na Seção 4.1. Em seguida, deve ser definida uma forma de representação das trajetórias, juntamente com a técnica de ajuste de curvas. Esses assuntos serão tratados na Seção 4.2. Em seguida, deve ser definida uma arquitetura, capaz de direcionar o processo de geração de curvas de acordo com os objetivos do algoritmo de roteamento. Na Seção 4.3 será apresentada a arquitetura de geração de curvas proposta nesta dissertação. Para finalizar, na Seção 4.4, será feita uma análise da solução proposta.

### 4.1 Entrada: Mapa de Energia

A primeira pergunta que surge ao se tentar resolver o *Problema de Geração de Curvas de Roteamento em Redes de Sensores Sem Fio* é em que tipo de informação esse processo deve se basear. Nós decidimos utilizar como entrada o mapa de energia da rede. O mapa de energia da rede é composto por um conjunto de coordenadas e um conjunto de medidas de energia em cada uma dessas coordenadas. As coordenadas podem ser as posições dos nós na área sensoriada, ou podem ser simplesmente pontos localizados na superfície dessa área. A medida de energia em cada coordenada pode ser a quantidade

de energia restante na bateria de cada nó-sensor, ou pode ser calculada como uma interpolação ou soma das energias dos nós que cobrem cada ponto.

O mapa de energia foi escolhido como entrada de dados para o processo de geração de curvas por duas razões principais:

- Importância do recurso energia no contexto de redes de sensores sem fio;
- Existência de uma técnica eficiente de obtenção do mapa.

Em muitas aplicações, os sensores serão lançados em uma área remota, o que impossibilitará o acesso a esses dispositivos para manutenção. Nesse cenário, o tempo de vida de um sensor dependerá da quantidade de energia que o mesmo possui. Portanto, todas as operações do sistema, tais como comunicação e processamento de dados, devem levar em consideração o consumo de energia.

O mapa de energia utilizado nas simulações foi obtido através do modelo de predição baseado em cadeias de Markov, proposto em [29]. O custo de obtenção do mapa de energia utilizando essa abordagem é aceitável para redes de sensores sem fio, e esse custo é detalhado em [29]. Todavia, destaca-se que o custo para obter o mapa de energia deve ser amortizado e distribuído entre todas as aplicações que o utilizarem.

Neste trabalho, nós utilizamos o mapa de energia, mas vale ressaltar que outros tipos de mapas podem ser utilizados para gerar as curvas de roteamento. Como exemplo, podemos citar o mapa de vulnerabilidade da rede, que poderia atribuir valores a diferentes regiões com base em fatores ligados à segurança do sistema, ou o mapa de eventos da área sensoriada, que poderia mapear probabilidades de ocorrência de eventos a diferentes partes da área. Um mecanismo de predição de eventos teria que ser desenvolvido para isso. Resumindo, o mapa de energia é só um exemplo, sendo que a entrada de dados para o mecanismo de geração de curvas pode variar de aplicação para aplicação.

#### *4.1.1 Seleção de Pontos*

Tendo definido que a entrada de dados para o processo será o mapa de energia da rede, um conjunto de pontos deve ser selecionado para servir de entrada para o algoritmo de ajuste de curvas. Existem duas formas de geração do mapa: como um conjunto de pontos geográficos, ou como um conjunto de nós. Se o mapa é gerado como um conjunto de pontos no espaço, a energia associada a cada coordenada é uma interpolação de energias disponíveis nos nós que cobrem esse ponto. Se o mapa é gerado a partir de um conjunto de nós, o conjunto de coordenadas utilizado é o das coordenadas dos próprios

nós, e a energia em cada coordenada é a energia disponível em cada nó. Neste trabalho, nós utilizamos ambas as formas e obtivemos melhores resultados com a segunda, já que o número total de coordenadas é reduzido, sem perda de precisão, reduzindo o tempo de processamento do algoritmo de ajuste de curvas.

Tendo gerado o mapa de energia, várias estratégias podem ser utilizadas para selecionar um subconjunto do total de coordenadas usadas para servir de entrada para o algoritmo de ajuste de curva. O critério principal para essa seleção é a quantidade de energia disponível em cada coordenada. A idéia é forçar as trajetórias a passarem pelos pontos de maior energia de forma a evitar que nós de pouca energia participem do processo de propagação de dados de terceiros. Outro critério utilizado pode ser a densidade de nós em cada região da rede. Quanto maior a densidade, maior a conectividade da região e maior a probabilidade de entrega do pacote. Isso ocorre porque os nós são programados para entrar em estados de “sono” e, portanto, sempre existe a possibilidade de uma trajetória se quebrar em caso de nenhum nó estar acordado para receber e propagar o pacote.

Neste trabalho, os pontos de entrada para o processo de ajuste de curva foram escolhidos baseando-se na combinação dos critérios de energia e densidade. Para cada nó da rede, a soma da energia de todos os seus vizinhos, juntamente com a sua própria, foi calculada. Em seguida, os nós foram ordenadas em ordem decrescente desse parâmetro, e a metade superior foi selecionada.

## 4.2 Representação e Ajuste de Curvas

Neste trabalho, nós decidimos optar por duas formas de representação de trajetórias: funcional (polinômios) e equacional (seções cônicas). A vantagem da representação funcional é a facilidade com que o valor da variável dependente  $y$  pode ser calculado para qualquer ponto do domínio da variável independente  $x$ . A vantagem da representação equacional é o seu maior poder de representação, que engloba curvas, tais como seções cônicas, e a sua maior estabilidade computacional. Além disso, esse método é tão compacto quanto a representação funcional, em termos de número de parâmetros necessários para codificar uma curva. Nós achamos essas duas formas de representação suficientemente expressivas para garantir a flexibilidade necessária para propagar pacotes por áreas de maior energia, dado que os pontos de entrada são selecionados a partir do mapa de energia da rede.

Como método de ajuste de curvas, nós optamos pela Regressão Linear Múltipla (ou Mínimos Quadrados), apresentada na Seção 3.4.4. As razões para tal

escolha são a simplicidade da técnica e a sua aplicação direta para o tipo de curvas e o tipo de dados de entrada tratados aqui.

A regressão polinomial é um caso especial de regressão linear múltipla. A relação entre a variável de resposta  $y$  e a variável explicativa  $x$  é expressa pelo seguinte modelo:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \dots + \beta_g x^g + \epsilon \quad (4.1)$$

Nesse caso, simplificando a notação  $\sum_{i=1}^n$  para  $\Sigma$ , e aplicando (3.13), o sistema de equações normais é definido da seguinte forma:

$$\begin{bmatrix} n & \Sigma x_i & \dots & \Sigma x_i^g \\ \Sigma x_i & \Sigma x_i^2 & \dots & \Sigma x_i^{g+1} \\ \Sigma x_i^2 & \Sigma x_i^3 & \dots & \Sigma x_i^{g+2} \\ \dots & \dots & \dots & \dots \\ \Sigma x_i^g & \Sigma x_i^{g+1} & \dots & \Sigma x_i^{2g} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \dots \\ \beta_g \end{bmatrix} = \begin{bmatrix} \Sigma y_i \\ \Sigma x_i y_i \\ \Sigma x_i^2 y_i \\ \dots \\ \Sigma x_i^g y_i \end{bmatrix} \quad (4.2)$$

Para seções cônicas, por outro lado, o processo de ajuste não é tão direto assim, já que eles resultam em um sistema não-linear. No entanto, se minimizarmos o quadrado das diferenças de área, ao invés das diferenças de distância, podemos obter um problema linear [5, 25]. Representemos uma seção cônica genérica por uma equação de segundo grau:

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \quad (4.3)$$

O conjunto de equações normais para esse caso pode ser expresso da seguinte forma:

$$\begin{bmatrix} x_1^2 - y_1^2 & x_1 y_1 & x_1 & y_1 & 1 \\ x_2^2 - y_2^2 & x_2 y_2 & x_1 & y_1 & 1 \\ x_3^2 - y_3^2 & x_3 y_3 & x_1 & y_1 & 1 \\ x_4^2 - y_4^2 & x_4 y_4 & x_1 & y_1 & 1 \\ x_5^2 - y_5^2 & x_5 y_5 & x_1 & y_1 & 1 \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} = \begin{bmatrix} -x_1^2 - y_1^2 \\ -x_2^2 - y_2^2 \\ -x_3^2 - y_3^2 \\ -x_4^2 - y_4^2 \\ -x_5^2 - y_5^2 \\ \dots \end{bmatrix} \quad (4.4)$$

onde  $a = 1 + \beta_0$ ,  $b = \beta_1$ ,  $c = 1 - \beta_0$ ,  $d = \beta_2$ ,  $e = \beta_3$  e  $f = \beta_4$ .

Vários métodos iterativos existem para resolver esse problema [24, 38, 16]. Nós utilizamos o algoritmo LSQR, proposto em [38], já que ele apresenta um comportamento mais estável e obtém soluções mais precisas em menos iterações, sempre que a matriz de coeficientes é moderada ou severamente mal-condicionada. Os requisitos computacionais desse algoritmo são resumidos na Tabela 4.1, onde  $n$  é o número de pontos ou medições (linhas) e  $p$  é o número de parâmetros estimados (colunas).

Além disso, o procedimento requer o produto  $Ax$  e o produto  $A^t y$  em cada

Tabela 4.1: Requisitos computacionais do algoritmo LSQR.

Memória	$n + 2p$
Número de multiplicações de ponto flutuante por iteração	$3n + 5p$

iteração, onde  $A$  é a matriz  $n \times p$ , tal que  $Ax = y$ . Para obter a utilização de memória mostrada, os produtos matriz-vetor são da seguinte forma:

$$y \leftarrow y + Ax \tag{4.5}$$

$$x \leftarrow x + A^t y \tag{4.6}$$

onde  $\leftarrow$  significa que um dos vetores dados é sobrescrito pela expressão mostrada. O número máximo de iterações utilizado foi  $4np$  (valor sugerido pelos autores do algoritmos).

### 4.3 Arquitetura

Tendo definido o tipo de dados que servirá de entrada para o processo de geração das curvas de roteamento, a forma de representação das trajetórias e a técnica de ajuste de curvas, algumas questões ligadas à arquitetura do processo ainda devem ser resolvidas. A arquitetura de geração de trajetórias proposta neste trabalho está ilustrada na Figura 4.1 e será detalhada ao longo desta seção.

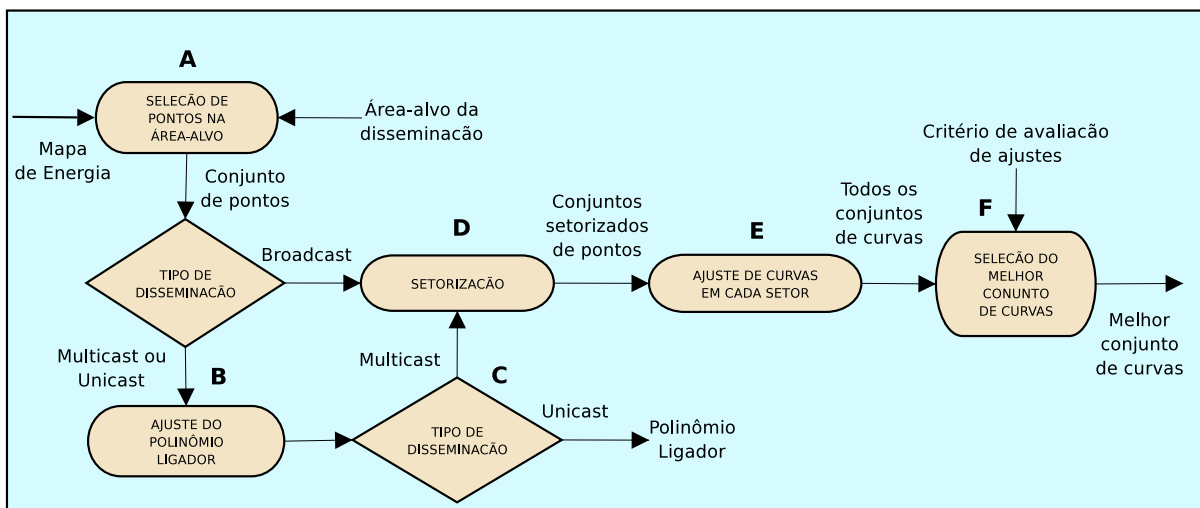


Figura 4.1: Processo de geração de curvas.

O processo apresenta pequenas variações de acordo com o tipo de disseminação. Como pode ser observado na Figura 4.1(Ponto A), a primeira etapa do processo é a seleção de pontos na área-alvo da disseminação. Se a disseminação for do tipo *broadcast*, os pontos são selecionados a partir do mapa de

energia inteiro, através do procedimento descrito na Seção 4.1.1. Caso a disseminação seja do tipo *multicast*, dois conjuntos de pontos são selecionados: um dentro da área compreendida entre o nó-monitor e o início da área-alvo e o outro somente dentro da área-alvo. Como será explicado mais adiante, o primeiro conjunto de pontos será usado para realizar o ajuste de uma curva chamada *polinômio-ligador*. O polinômio-ligador terá a função de “túnel” entre o nó-monitor e a área-alvo. O segundo conjunto de pontos será utilizado para realizar a disseminação de dados dentro da área-alvo. Caso a disseminação seja do tipo *unicast*, a seleção de pontos será feita a partir da área compreendida entre o nó-monitor e o nó-destino. Nesse caso, existirá apenas uma curva de roteamento, também denominada polinômio-ligador, que ligará o ponto de origem ao ponto de destino.

Nas disseminações do tipo *broadcast* e *multicast* será empregado um procedimento, chamado *setorização*, que será explicado na seção seguinte.

### 4.3.1 Setorização da Rede

Dado um conjunto de pontos que nós gostaríamos de forçar a participar do processo de propagação de pacotes e dados os tipos das curvas (polinômios e seções cônicas), devemos determinar quantas curvas (ou trajetórias) são necessárias para atingir um determinado objetivo. O objetivo poderia ser disseminar dados para toda a rede (*broadcast*), para uma parte da rede (*multicast*) ou para um nó específico (*unicast*).

Ao introduzir o conceito de *setores de rede*, que dividem a rede em setores angulares iguais, centrados em um nó-monitor, o problema de determinar o melhor número de trajetórias pode ser visto como o problema de determinar o melhor número de setores de rede, inserindo uma trajetória em cada setor. A curva correspondente a cada setor de rede é ajustada com base somente nos pontos localizados dentro da área do seu setor. Exemplos de diferentes conjuntos de setores de rede podem ser vistos nas Figuras 4.2(a) a 4.2(f).

Um número arbitrário de setores de rede poderia ser usado. No entanto, não é razoável que esse número seja muito grande, já que isso resultaria em um número inaceitavelmente grande de parâmetros a serem transmitidos em cada pacote e um número inaceitavelmente pequeno de pontos em cada setor, comprometendo a qualidade do procedimento de ajuste. Portanto, um limite máximo pode ser estabelecido para o número de setores usado. Esse limite depende tanto do tipo de curva utilizado, como da posição do nó-monitor. Se a curva for do tipo polinômio, cada setor pode ter um ângulo bastante reduzido, dado o formato do polinômio, que pode assumir até a forma de uma reta. Caso a curva seja do tipo seção cônica, o ângulo do setor deve ser maior, de forma a acomodar toda a curva. Quando o nó-monitor é localizado em um dos cantos

da rede, a setorização é realizada em um ângulo total de  $90^\circ$ . Quando o mesmo é posicionado no centro, esse ângulo é de  $360^\circ$ . Assim, o número máximo de setores pode ser maior nesse último caso. Essa situação está ilustrada na Figura 4.3.

Quando o ajuste polinomial é realizado, além do número de setores de rede, o grau do polinômio pode influenciar na qualidade do ajuste. Portanto, o ajuste de curva é feito não somente para conjuntos diferentes de setores e tipos de curva diferentes, mas também para graus diferentes do polinômio. O grau máximo de polinômio utilizado neste trabalho foi 4, já que graus maiores implicam em complicações para o algoritmo de cálculo de distância do nó à curva. Depois de gerados todos os conjuntos de curvas possíveis, é selecionado o conjunto que apresenta a melhor *qualidade média*. Esse processo será explicado na seção seguinte.

### 4.3.2 Seleção do Melhor Conjunto de Curvas

Fixado um número máximo de setores de rede (*maxNumSetores*), os tipos de curva utilizados e o grau máximo do polinômio, a seleção do melhor conjunto de curvas pode ser feita através do cálculo da qualidade média de cada conjunto e da escolha daquele, cuja qualidade média é a melhor. Os conjuntos de setores compreendem todos os conjuntos que possuem de um a *maxNumSetores* setores. A qualidade média de um conjunto de curvas pode ser calculada como a média aritmética das qualidades de ajuste de cada curva contida no conjunto:

$$qualidadeMedia_{tipoCurva} = \left( \frac{\sum_{i=1}^{numSetores} qualidadeAjuste(Curva_i)}{numSetores} \right),$$

*numSetores* = 1 . . . *maxNumSetores*

onde *maxNumSetores* depende da posição do nó-monitor e do tipo de curva utilizado e *tipoCurva* compreende seções cônicas e polinômios de grau variando de 1 a 4. O fator *qualidadeAjuste(Curva<sub>i</sub>)* pode ser calculado com base em diferentes critérios, dependendo dos requisitos da aplicação. O objetivo pode ser maximizar a cobertura da curva ou minimizar o envolvimento de nós de baixa energia no processo de propagação, por exemplo. Neste trabalho, os seguintes *critérios de avaliação de ajuste* foram utilizados:

- *Máxima energia média*: maximiza a energia média dos nós dentro do raio de cobertura da curva ( $distancia(no, curva) \leq raio\_comunicacao\_no$ );
- *Máxima cobertura*: maximiza o número de nós dentro do raio de cobertura da curva.

Nas Figuras 4.2(a) a 4.2(f), vários *snapshots* de conjuntos de curvas gerados para um determinado mapa de energia são mostrados. As Figuras 4.2(a) a 4.2(c) mostram conjuntos de seções cônicas geradas para esse cenário. Podemos observar que as trajetórias evitam as áreas de baixa energia. Se o critério de *máxima energia média* ou *máxima cobertura* é utilizado para selecionar um desses conjuntos de curvas, o melhor conjunto é aquele com quatro setores. Esse resultado é obtido porque a *qualidade média* das quatro curvas participantes é melhor segundo os dois critérios. De fato, podemos ver que esse conjunto apresenta melhor cobertura e passa por pontos de maior energia.

As Figuras 4.2(d) a 4.2(f) mostram conjuntos de polinômios de quarto grau, gerados para o mesmo cenário. Podemos observar que, mais uma vez, as trajetórias evitam as áreas de menor energia. Se o critério de *máxima energia média* é aplicado para selecionar um desses conjuntos, o melhor é o que contém apenas um setor de rede. Esse resultado é obtido, pois a energia média dos nós dentro da área de cobertura dessa curva é mais alta do que em outros conjuntos. Por outro lado, se o critério de *máxima cobertura* for utilizado, o melhor conjunto é aquele com oito setores de rede. Esse comportamento é natural, já que quanto maior o número de curvas, em geral, maior é o número de nós cobertos pelas mesmas.

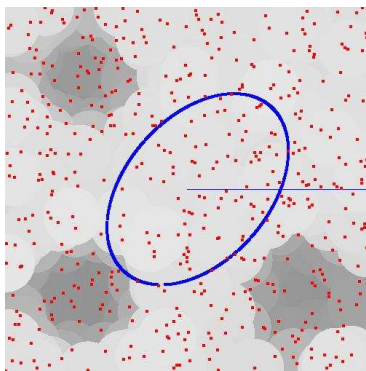
Nas seções seguintes, serão ilustrados diversos cenários de setorização e ajuste de curvas.

### 4.3.3 *Broadcast*

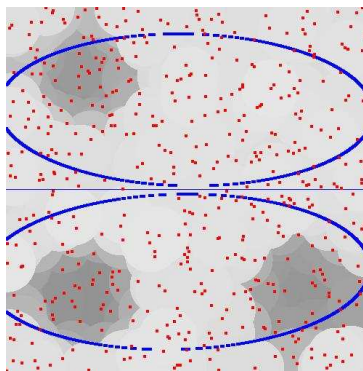
O processo de geração de curvas para disseminação do tipo *broadcast* está ilustrado na Figura 4.1. A primeira etapa do processo consiste em selecionar os pontos que irão servir de entrada para o processo de ajuste de curvas dentro da área-alvo (Figura 4.1, ponto A). Nesse caso, a área-alvo é todo o mapa de energia. O processo de seleção de pontos está descrito na Seção 4.1.1. Em seguida, é feita uma setorização da rede, seguindo o procedimento descrito na Seção 4.3.1 (Figura 4.1, ponto D). Ao serem gerados todos os conjuntos setorizados de pontos, são realizados os ajustes de cada curva pertencente a cada setor de cada conjunto de setores (Figura 4.1, ponto E). A curva dentro de cada setor é ajustada com base nos pontos dentro daquele setor e tem a restrição de interceptar o ponto de origem (posição do nó-monitor). Após serem gerados todos os conjuntos de curvas, o melhor conjunto é selecionado de acordo com o critério de avaliação de ajustes usado (Figura 4.1, ponto F). A saída do processo é o melhor conjunto de curvas (segundo um determinado critério), que realiza uma disseminação do tipo *broadcast*.

As Figuras 4.4(a) e 4.4(b) ilustram dois conjuntos de curvas selecionados

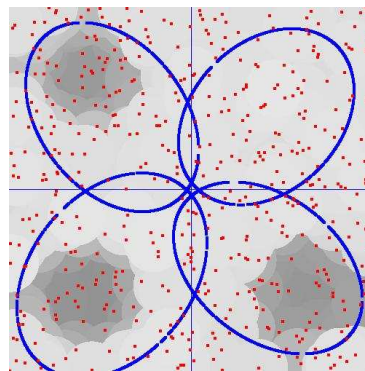




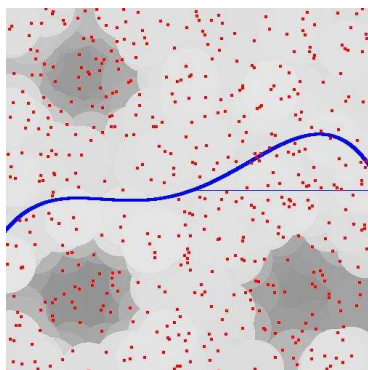
(a) Conjunto com 1 setor.



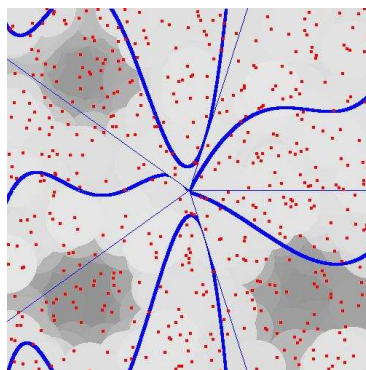
(b) Conjunto com 2 setores.



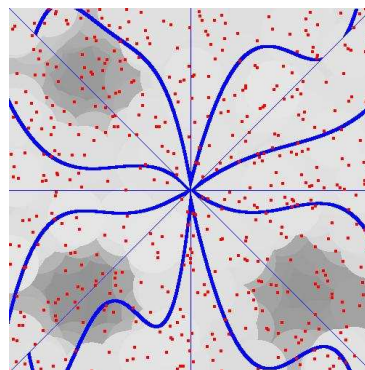
(c) Conjunto com 4 setores.



(d) Conjunto com 1 setor.

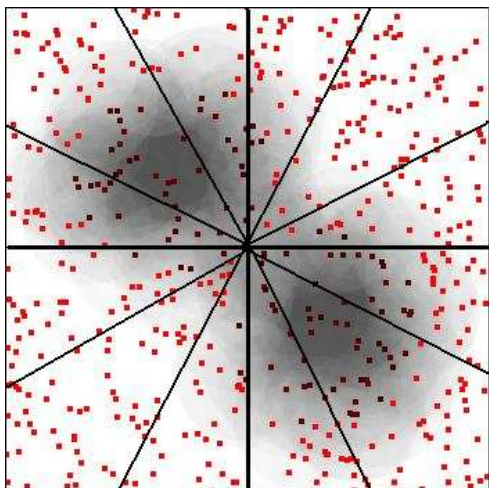


(e) Conjunto com 5 setores.

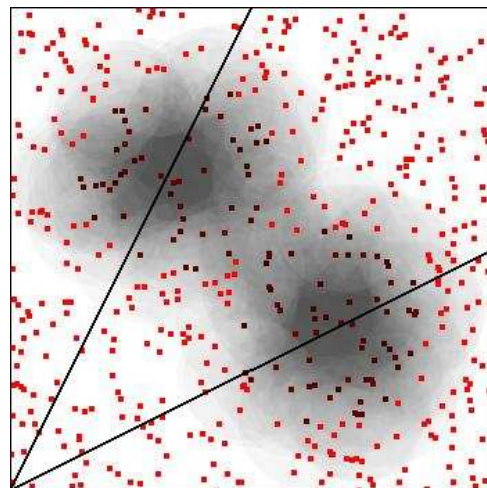


(f) Conjunto com 8 setores.

Figura 4.2: Seções cônicas e polinômios de quarto grau sobre um mapa de energia.



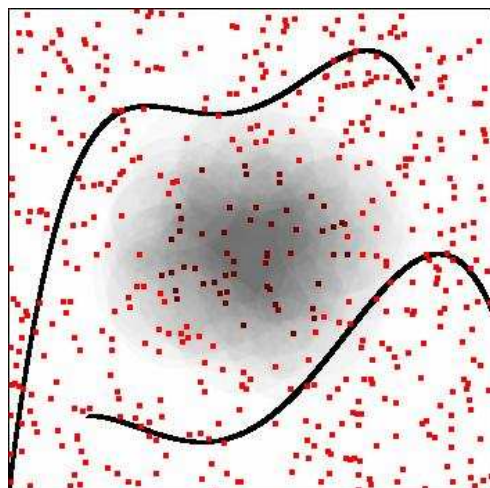
(a) Setorização com nó-monitor no centro.



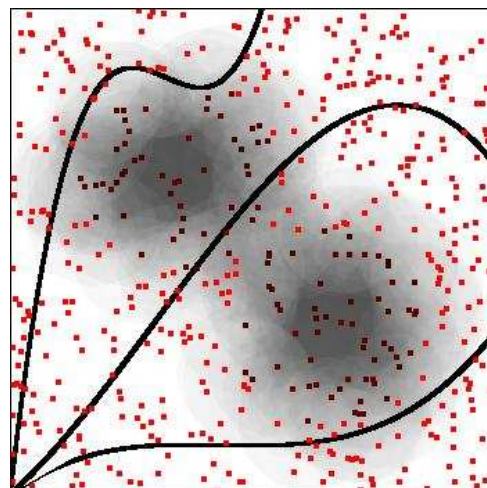
(b) Setorização com nó-monitor no canto inferior esquerdo.

Figura 4.3: Número máximo de setores depende da posição do nó-monitor.

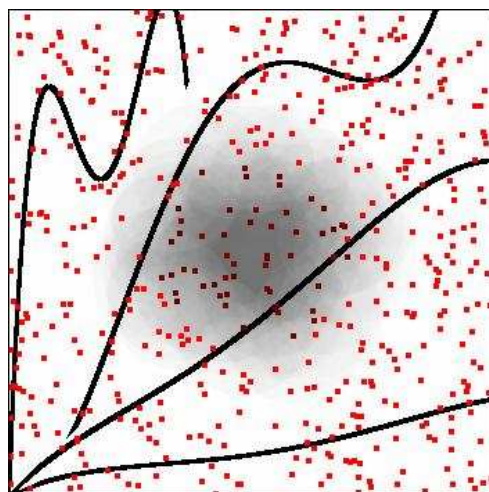
para dois mapas de energia diferentes, utilizando o critério de *energia máxima*. As Figuras 4.4(c) e 4.4(d) ilustram os mesmos cenários, porém utilizando o critério de *cobertura máxima*. O número máximo de setores utilizado foi igual a quatro. Pode-se observar que, quando a energia remanescente nos nós é priorizada, são escolhidos conjuntos com menos setores, porém com curvas que evitam ao máximo as regiões de baixa energia. Quando a cobertura é priorizada, por outro lado, o número máximo de setores é utilizado e, dentro dos limites de cada setor, a curva é ajustada de forma a se aproximar dos nós de maior energia.



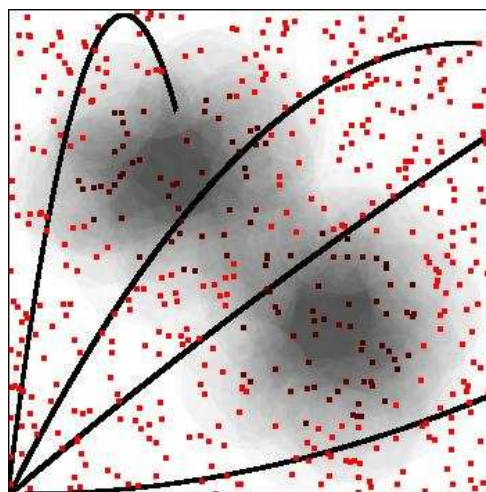
(a) Critério de avaliação de ajuste = *energia máxima*.



(b) Critério de avaliação de ajuste = *energia máxima*.



(c) Critério de avaliação de ajuste = *cobertura máxima*.



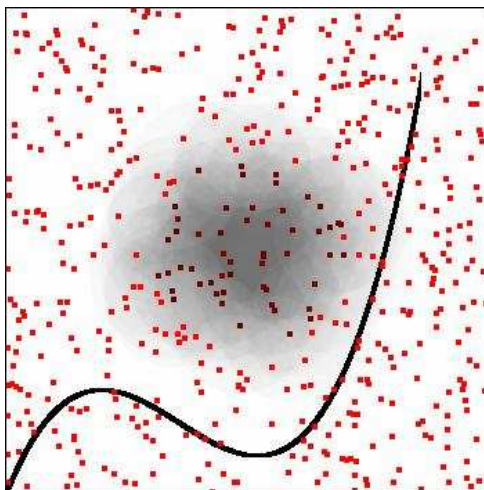
(d) Critério de avaliação de ajuste = *cobertura máxima*.

Figura 4.4: Conjuntos de curvas para realizar *broadcast*. Mapa de energia (35X35) com uma e duas regiões de baixa de energia.

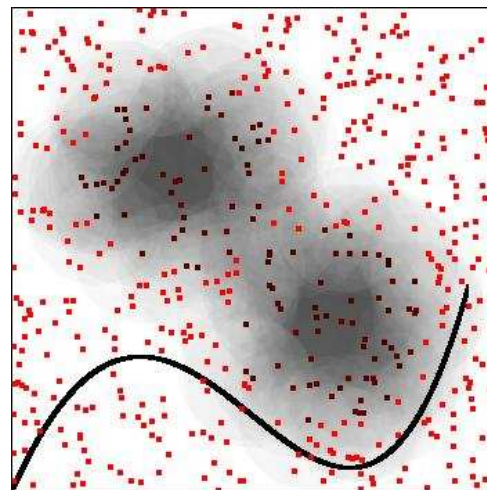


### 4.3.4 Unicast

O processo de disseminação do tipo unicast está ilustrado na Figura 4.1. A seleção de pontos (Figura 4.1, ponto A) é feita utilizando todos os pontos que possuem suas coordenadas  $x \leq x_{destino}$ . Em seguida (Figura 4.1, ponto B), é realizado o ajuste de uma curva, chamada polinômio-ligador, que faz a conexão entre o ponto de origem (nó-monitor) e o ponto de destino (nó-alvo da disseminação). O ajuste do polinômio-ligador é feito utilizando o critério de energia máxima, para que ele possa evitar as regiões de baixa energia. Além disso, uma restrição que force o polinômio a passar pelo ponto de destino deve ser adicionada. A saída do processo é uma única curva - o polinômio-ligador - que intercepta o nó-monitor e o nó-destino, evitando regiões de baixa energia no caminho. A Figura 4.5 ilustra a curva de unicast gerada para dois mapas de energia diferentes e dois pontos de destino diferentes.



(a) Ponto de destino = (30, 30).



(b) Ponto de destino = (33, 15).

Figura 4.5: Conjuntos de curvas para realizar *unicast*. Mapa de energia (35X35) com uma e duas regiões de baixa de energia.

### 4.3.5 Multicast

O processo de disseminação de dados do tipo *multicast* também está ilustrada na Figura 4.1. Neste trabalho, o termo “disseminação do tipo *multicast*” é usado para designar uma situação em que o nó-monitor está interessado em disseminar dados para uma determinada região geográfica. Esse conceito de *multicast* é um pouco diferente do conceito geralmente utilizado para designar um processo de disseminação para um subconjunto qualquer de nós. No caso tratado aqui, esse subconjunto-destino de nós deve agrupar nós localizados em uma determinada região geográfica. A seleção de pontos (Figura 4.1, ponto

A) é realizada em duas etapas: a primeira seleciona os pontos fora da área-alvo, para servir de entrada para o ajuste do polinômio-ligador, e a segunda seleciona pontos dentro da área-alvo, para fazer o ajuste de curvas que irão disseminar dados dentro da área.

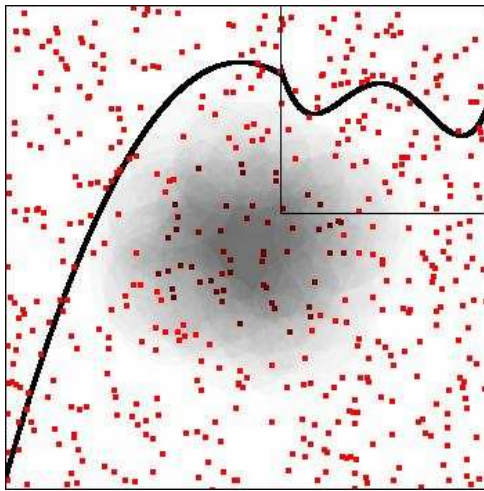
A geração do polinômio-ligador é feita seguindo o procedimento *unicast*, descrito na Seção 4.3.4, com uma diferença: o ponto-destino do polinômio deve ser um ponto localizado sobre a fronteira da área-alvo. Esse ponto de interseção entre o polinômio-ligador e a área-alvo será a origem das curvas de disseminação dentro da área-alvo. Para não sobrecarregar os nós localizados nesse ponto, o mesmo deve ser escolhido dinamicamente, com base na quantidade de energia e densidade de nós no ponto de interseção. Assim, dados todos os pontos localizados na fronteira da área-alvo, o ponto de interseção será aquele que possui a maior soma de energias de nós que o cobrem.

Para gerar as curvas de disseminação dentro da área-alvo, o processo é similar ao *broadcast*, descrito na Seção 4.3.3, porém é realizado somente dentro da área-alvo. Além disso, o processo de setorização possui como ponto de origem não a posição do nó-monitor, como no caso de *broadcast*, mas o ponto de interseção do polinômio-ligador com a área-alvo. Para escolher o melhor conjunto de curvas, tanto o critério de *energia máxima*, como o critério de *cobertura máxima* podem ser utilizados. A saída do processo é o polinômio-ligador, mais as curvas de disseminação dentro da área-alvo.

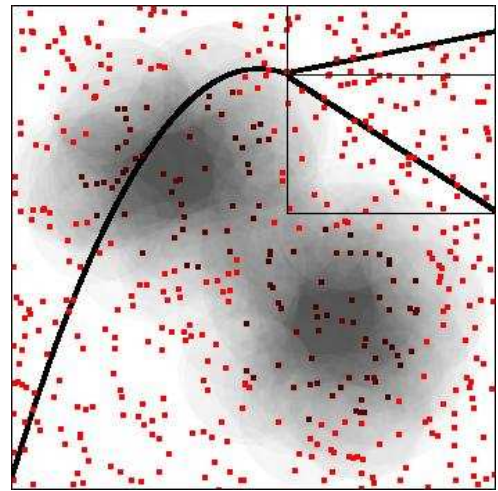
As Figuras 4.6(a) e 4.6(b) ilustram dois conjuntos de curvas de *multicast*, orientadas para a região alvo delimitada pelo retângulo (20,20) - (35, 35), utilizando o critério de *energia máxima*. As Figuras 4.6(c) e 4.6(d) ilustram os mesmos cenários, porém utilizando o critério de *cobertura máxima*. Pode-se observar que o polinômio-ligador contorna a área de baixa energia em todos os casos. Quando o critério de *energia máxima* é utilizado, um número menor de setores é utilizado dentro da área-alvo. Quando o critério de *cobertura máxima* é utilizado, além do número de setores ser maior, o grau do polinômio utilizado também é maior. Essas características garantem que a cobertura seja maximizada.

## 4.4 Considerações Finais

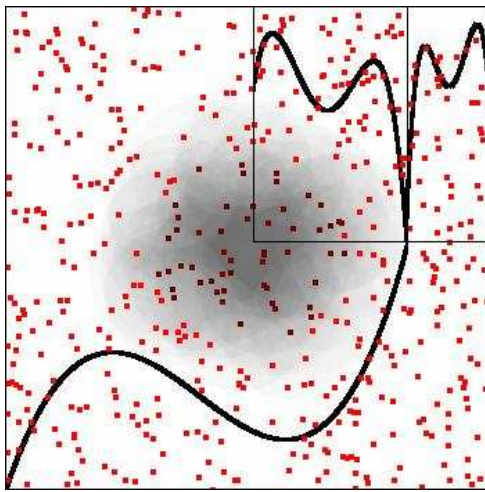
É interessante ressaltar que a estratégia de geração de trajetórias proposta nesta dissertação não é restrita aos cenários de rede aqui ilustrados. O mapa de energia usado como entrada pode ser relativo a uma rede com forma arbitrária. Além disso, o número e localização de nós-monitores também pode variar. No caso em que o número de nós-monitores é maior do que um, cada nó-sensor pode participar em mais de uma trajetória, possivelmente propa-



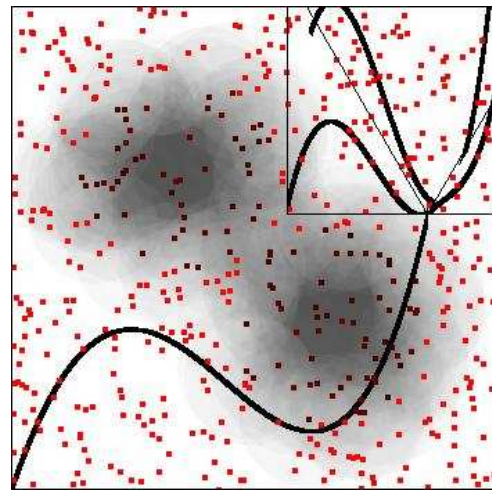
(a) Critério de avaliação de ajuste = *energia máxima*.



(b) Critério de avaliação de ajuste = *energia máxima*.



(c) Critério de avaliação de ajuste = *cobertura máxima*.



(d) Critério de avaliação de ajuste = *cobertura máxima*.

Figura 4.6: Conjuntos de curvas para realizar *multicast*. Mapa de energia (35X35) com uma e duas regiões de baixa de energia. Área-alvo = (20,20)-(35,35).

gando pacotes originados por diferentes nós-monitores. Essa solução apresenta duas características importantes para um sistema autônomo: flexibilidade e adaptabilidade.

Outra consideração importante é em relação ao processo de codificação de trajetórias. Os parâmetros das curvas podem ser embutidos no cabeçalho de todos os pacotes, ou podem ser pré-configurados nos nós-sensores na primeira vez que os mesmos recebem o pacote com um dado conjunto de curvas. Nas disseminações seguintes, as curvas podem ser referenciadas através de identificadores criados na fase de pré-configuração. Nesse caso, o nó-monitor deve ser capaz de atualizar esses valores periodicamente. Dessa maneira, o número de dados transmitidos em cada pacote é economizado.

A solução para o *Problema de Geração de Curvas de Roteamento em Redes de Sensores Sem Fio* proposta neste capítulo, apresenta as seguintes características:

- *Geração Dinâmica.* As curvas são geradas dinamicamente, com base na versão mais atual do mapa de energia de que o nó-monitor dispõe.
- *Consideração do fator energia.* Todo o processo é baseado no mapa de energia da rede.
- *Controle de cobertura.* A seleção do melhor conjunto de curvas é feita de forma a maximizar o número de nós cobertos pelas curvas.
- *Controle de quebras de trajetória.* A quebra de trajetórias, devido a trechos não cobertos por nenhum nó, é minimizada através da estratégia de seleção de pontos, que é baseada na densidade e na quantidade de energia em cada ponto que serve de entrada para o procedimento de ajuste de curvas.
- *Controle de número de parâmetros transmitidos.* O número máximo de parâmetros transmitidos pode ser limitado pelo número máximo de setores de rede e pelo grau dos polinômios.
- *Adaptabilidade aos requisitos da aplicação.* A aplicação pode definir os critérios de avaliação para os ajustes de curva. Neste trabalho foram feitos testes considerando o critério de *energia máxima* e de *cobertura máxima*, no entanto, outros parâmetros poderiam facilmente ser aplicados.
- *Custo computacional.* A solução proposta utiliza uma abordagem centralizada, ou seja, todos os cálculos são realizados no nó-monitor. Assume-se que o mesmo não possui restrições de energia e poder computacional.

Portanto, o custo para gerar as curvas pode ser considerado irrelevante do ponto de vista da rede.

No próximo capítulo, serão apresentados os resultados de simulação para validar o mecanismo de geração dinâmica de curvas de roteamento proposto.





---

## Resultados de Simulação

---

Neste capítulo serão descritos e analisados os experimentos realizados para validar o funcionamento do mecanismo de geração dinâmica de curvas proposto neste trabalho. Os experimentos foram realizados através de simulação de um ambiente de disseminação de dados em uma rede de sensores sem fio. Uma série de topologias de rede foram geradas aleatoriamente. Variou-se tanto a localização dos nós-sensores, como a sua distribuição de energia. Para cada topologia, simulou-se uma série de disseminações de dados realizadas por um nó-monitor. Dois tipos de disseminação foram analisados: *broadcast* e *multicast*.

O mecanismo de geração de curvas foi usado em conjunto com dois algoritmos de propagação em curva: TBF [33, 35] e TEDD [8, 10, 11, 12]. O desempenho desses algoritmos foi comparado ao desempenho de dois algoritmos baseados em inundação: *flooding* e *gossiping* [13]. Esses protocolos foram descritos no Capítulo 2.

Na Seção 5.1, será descrito o ambiente de simulação, juntamente com os cenários de rede utilizados. Na Seção 5.2, serão apresentadas as métricas utilizadas. Na Seção 5.3, será analisado o desempenho das curvas geradas no contexto de disseminação de dados para a rede inteira (*broadcast*), num cenário de distribuição uniforme de energia. Na Seção 5.4, será analisada a capacidade das curvas geradas de evitar regiões de baixa energia. Na Seção 5.5, serão mostrados e discutidos os resultados obtidos com disseminação de dados para uma região específica da rede (*multicast*).

## 5.1 Ambiente de Simulação

Para avaliar o desempenho do mecanismo de geração dinâmica de curvas, foi utilizado o simulador ns-2.26 (*Network Simulator 2.26*) [36]. Foi simulado o comportamento de dois protocolos de roteamento em curva, TEDD e TBF, e dois protocolos baseados em inundação, *gossiping* (com probabilidade 0.4) e *flooding*. O protocolo utilizado na camada MAC foi o padrão do ns-2.26 - uma versão simplificada do protocolo 802.11 [37].

Para modelar o consumo de energia dos nós-sensores, utilizou-se o Modelo de Disseminação de Energia Baseado em Estados (*State-based Energy Dissipation Model (SEDM)*), proposto em [29]. No SEDM, os nós possuem vários modos de operação com diferentes níveis de ativação e, conseqüentemente, diferentes níveis de consumo de energia. Cada nó possui quatro modos de operação:

- Modo 1: sensor desligado, processador *idle*, e rádio desligado;
- Modo 2: sensor e processador ligados, e rádio desligado;
- Modo 3: sensor e processador ligados, e rádio recebendo;
- Modo 4: sensor e processador ligados, e rádio transmitindo.

Os valores de energia utilizados para cada componente básico de um nó-sensor (sensor, processador e rádio) foram obtidos a partir dos manuais do *Mica2* [27] e estão resumidos na Tabela 5.1. A partir dessa tabela, e sabendo que os sensores trabalham com uma tensão de 3 V, obtém-se os respectivos valores de energia para cada um dos quatro modos:

- Modo 1: 30  $\mu W$ ;
- Modo 2: 24,9  $mW$ ;
- Modo 3: 48,9  $mW$ ;
- Modo 4: 101,1  $mW$ .

<b>Componentes</b>	Ativo	Inativo ( <i>Idle</i> )
Processador	24 $mW$	24 $\mu W$
Rádio	24 $mW$ (RX) / 76.2 $mW$ (TX)	6 $\mu W$
Sensor Temperatura	0.9 $mW$	-

Tabela 5.1: Consumo de cada componente básico de um nó-sensor.

O mapa de energia utilizado nas simulações foi obtido através do modelo de predição baseado em cadeias de Markov, proposto em [29]. O custo de

obtenção do mapa de energia utilizando essa abordagem é aceitável para redes de sensores sem fio, e o mesmo é detalhado em [29]. Todavia, destaca-se que esse custo deve ser amortizado e distribuído entre todas as aplicações que o utilizarem. Para não influenciar os resultados dos experimentos, o custo do mapa de energia não foi considerado nas simulações.

A rede de sensores sem fio considerada no presente trabalho é composta por nós estáticos, homogêneos e dispostos aleatoriamente na área de simulação. A recarga da bateria é considerada inviável. A rede possui um único nó-monitor sem restrições de energia e localizado no canto inferior esquerdo da mesma. Em cada disseminação de dados, o nó-monitor recalcula o novo conjunto de trajetórias, baseando-se no mapa de energia corrente da rede. O custo desse processo não é considerado do ponto de vista da rede, porque todos os cálculos envolvidos são realizados no nó-monitor, e esse não possui restrições de energia.

Neste trabalho, é assumido que cada nó conhece sua própria localização e que o nó-monitor conhece a localização de todos os nós da rede. Além disso, todos os nós conhecem a localização do nó-monitor. O problema da localização é muito importante em RSSFs [34] e encontra-se em discussão no meio acadêmico. A maioria das soluções propostas são baseadas em técnicas que avaliam a intensidade do sinal, o ângulo de chegada e a distância entre os nós. Destaca-se que o uso de GPS (*Global Position System*) não é interessante em redes de sensores sem fio, porque o mesmo elevaria o custo de produção dos micro sensores. Uma das características das RSSFs é que o custo dos dispositivos deve ser o menor possível. Além disso, dependendo do ambiente, o GPS não pode ser utilizado, por exemplo, no fundo de oceanos ou em florestas densas.

Os principais parâmetros de simulação são apresentados na Tabela 5.2. Foram distribuídos de forma aleatória 500 nós-sensores sobre uma área de  $35 \times 35m$ . A energia inicial de cada nó foi de 40 *Joules*. O raio de transmissão foi fixado em  $5m$ . Nesse cenário, cada nó apresenta, em média, 27 vizinhos. Contudo, durante a simulação, esse valor é reduzido, já que nós adormecem para economizar energia. Conseqüentemente, durante uma disseminação de dados, nem todos os nós podem ser alcançados, uma vez que um ou mais nós podem estar dormindo ou apenas sensoriando (rádio desligado). Todos os resultados de simulação apresentados neste capítulo foram obtidos através da média de 33 simulações, realizadas utilizando diferentes sementes de randomização. O tempo total de cada simulação é de 1000 segundos. Durante cada simulação, o nó-monitor envia 200 mensagens, em instantes de tempo distribuídos uniformemente ao longo da simulação.

A máquina utilizada para executar as simulações foi Intel(R) Pentium(R) 4

Parâmetro	Valor
Área de simulação	35×35m <sup>2</sup>
Número de nós	500
Energia inicial	40 J
Raio de comunicação	5m
Número médio de vizinhos	27
Número de simulações	33
Duração de cada simulação	1000s
Número de mensagens disseminadas em cada simulação	200

Tabela 5.2: Parâmetros utilizados nas simulações.

## 5.2 Métricas

O mecanismo de geração de curvas foi usado em conjunto com dois algoritmos de propagação em curva: TBF [33, 35] e TEDD [8, 10, 11, 12]. O desempenho desses algoritmos foi comparado ao desempenho de dois protocolos baseados em inundação: *flooding* e *gossiping* [13].

As métricas utilizadas para avaliar o desempenho dos algoritmos de disseminação de dados foram:

- Cobertura: número de pacotes recebidos ao longo do tempo de vida da rede (Rx);
- Número de pacotes transmitidos ao longo do tempo de vida da rede (Tx);
- Razão Rx/Tx;
- Energia média da rede ao longo do seu tempo de vida;
- Número de nós mortos ao longo do tempo de vida da rede.

## 5.3 Broadcast: Distribuição Uniforme de Energia

Nesta seção será analisado o desempenho das curvas geradas no contexto de disseminação de dados para a rede inteira, num cenário de distribuição uniforme de energia. Nesse caso, a disseminação de dados possui dois objetivos: maximizar a cobertura da rede e reduzir o número de transmissões. Contudo, quando se reduz o número de transmissões, reduz-se a cobertura da rede; e quando se eleva o número de transmissões, eleva-se a cobertura da rede. Existem situações em que o primeiro objetivo é o mais relevante. Em outras situações, o segundo objetivo deve prevalecer. Uma métrica objetiva é

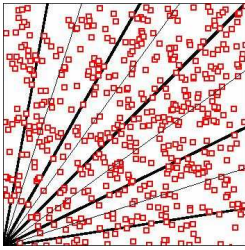
a razão entre o número de pacotes recebidos e o número de pacotes enviados na rede.

O critério de *cobertura máxima*, introduzido na Seção 4.3.2, foi utilizado aqui para selecionar o melhor conjunto de curvas. O número máximo de setores de rede utilizado foi igual a 5.

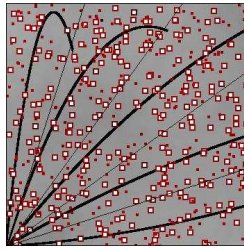
Na Figura 5.1, é ilustrada a evolução do mapa de energia durante o tempo de vida da rede para uma única simulação de cada protocolo. Além disso, é mostrada a cobertura da rede em três instantes: 0, 500 e 1000 segundos. Os quadrados brancos grandes representam os nós que receberam o pacote disseminado, os quadrados pretos pequenos indicam os nós que não receberam o pacote disseminado. Uma vez que o parâmetro *número máximo de setores* da rede foi fixado em 5, o número máximo de curvas permitido por disseminação para maximizar a cobertura da rede é igual a 5. É importante destacar que, apesar do processo de geração de curvas ser dinâmico, não existem mudanças significativas nos conjuntos de curvas gerados durante o tempo de vida da rede. Isso ocorre porque o nó-monitor sempre realiza disseminações para a rede toda e, como os nós dormem periodicamente, verifica-se um rodízio dos nós que transmitem o pacote. Conseqüentemente, o consumo de energia na rede é homogêneo.

Ao analisarmos a Figura 5.1, podemos observar que o consumo de energia do *flooding* é superior aos demais protocolos. TEDD, TBF e *gossiping* apresentam consumos similares. É importante ressaltar que a energia gasta pelo TBF para trocar mensagens do tipo *beacons*, para manter as suas tabelas de vizinhos, não foi levada em consideração nesses experimentos. Caso o consumo de energia com envio de pacotes de *beacons* fosse considerado nos experimentos, o desempenho do TBF seria inferior ao *flooding*. Essa escolha foi feita, pois o objetivo deste trabalho não é analisar os aspectos das diferentes técnicas de propagação em curva, mas sim comparar o desempenho dos algoritmos baseados em curvas com outras técnicas. É devido a esse detalhe que o consumo do TBF é mais baixo do que do TEDD nas figuras apresentadas. Como vai ser analisado mais adiante, apesar de o *gossiping* apresentar um consumo de energia similar ao do TEDD, o número de transmissões realizadas por ele é mais alto.

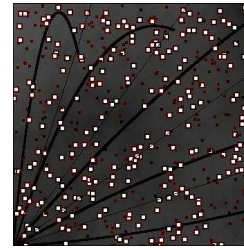
As Figuras 5.2(a) a 5.2(e) mostram a porcentagem de nós cobertos, o número de pacotes transmitidos, a energia média da rede, a porcentagem de nós mortos e a razão entre os números de pacotes recebidos e transmitidos. Na Figura 5.2(a) podemos observar que a cobertura do *flooding* começa com 80% e cai para zero após aproximadamente 800 segundos de simulação. O TEDD apresenta a melhor cobertura entre os protocolos restantes e o TBF apresenta um desempenho abaixo do esperado. A cobertura do TBF não ultrapassa



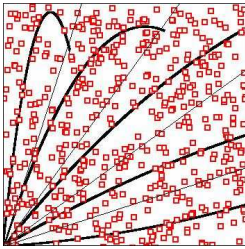
(a) TEDD,  $T = 0s$ :  $C = 60\%$ ,  $E = 100\%$ .



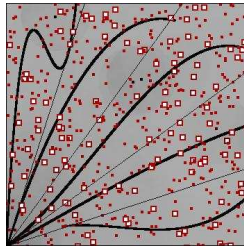
(b) TEDD,  $T = 500s$ :  $C = 55\%$ ,  $E = 60\%$ .



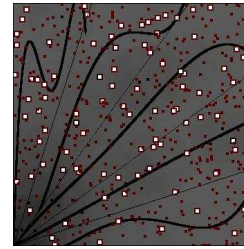
(c) TEDD,  $T = 1000s$ :  $C = 45\%$ ,  $E = 23\%$ .



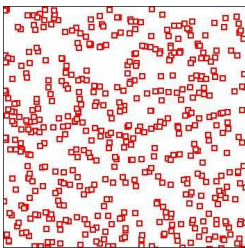
(d) TBF,  $T = 0s$ :  $C = 30\%$ ,  $E = 100\%$ .



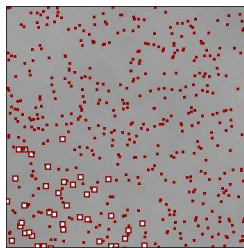
(e) TBF,  $T = 500s$ :  $C = 25\%$ ,  $E = 65\%$ .



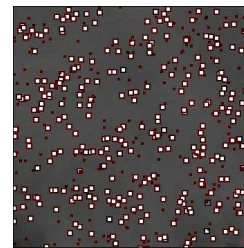
(f) TBF,  $T = 1000s$ :  $C = 20\%$ ,  $E = 33\%$ .



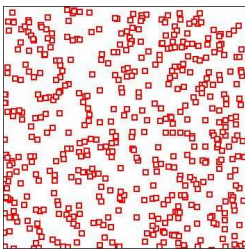
(g) Gossiping,  $T = 0s$ :  $C = 45\%$ ,  $E = 100\%$ .



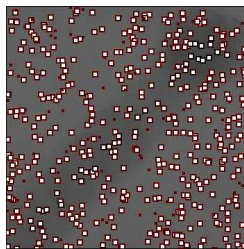
(h) Gossiping,  $T = 500s$ :  $C = 45\%$ ,  $E = 60\%$ .



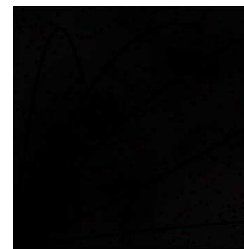
(i) Gossiping,  $T = 1000s$ :  $C = 40\%$ ,  $E = 23\%$ .



(j) Flooding,  $T = 0s$ :  $C = 80\%$ ,  $E = 100\%$ .

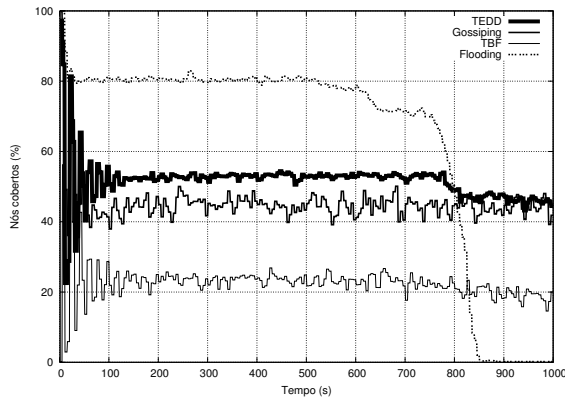


(k) Flooding,  $T = 500s$ :  $C = 80\%$ ,  $E = 35\%$ .

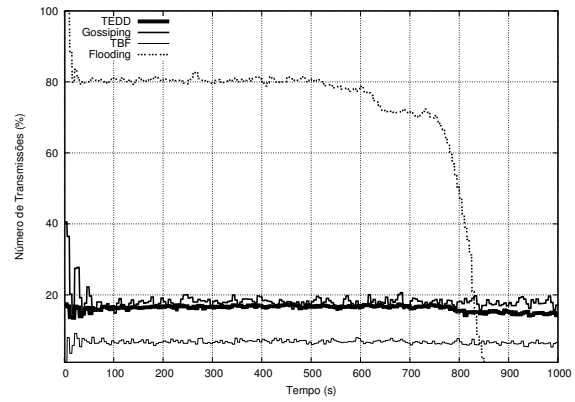


(l) Flooding,  $T = 1000s$ :  $C = 0\%$ ,  $E = 0\%$ .

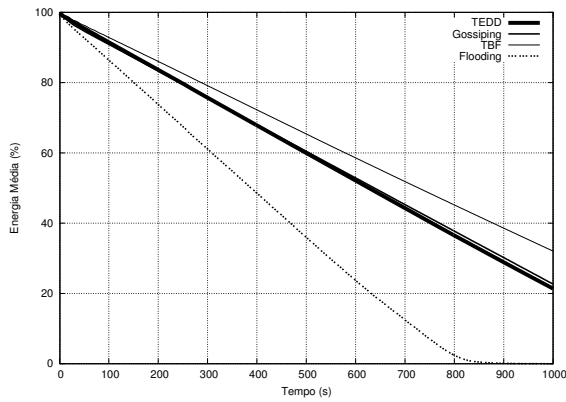
Figura 5.1: *Broadcast*: Mapa de energia e cobertura da rede. ( $T$ =tempo,  $C$ =cobertura,  $E$ =energia média)



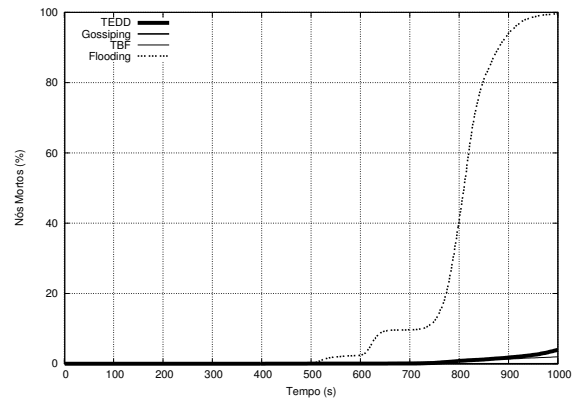
(a) Porcentagem de nós cobertos.



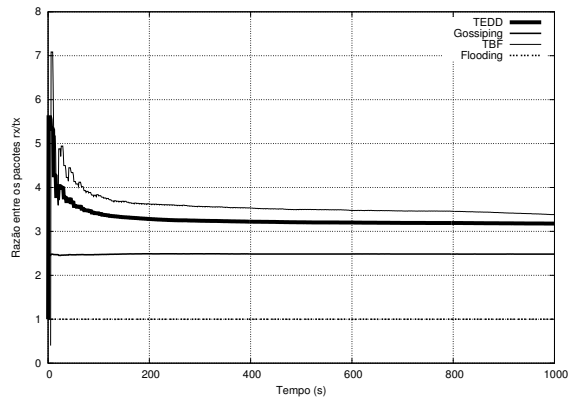
(b) Número de pacotes transmitidos.



(c) Energia Média.



(d) Porcentagem de nós mortos.



(e) Razão entre os pacotes recebidos e transmitidos (Rx/Tx).

Figura 5.2: *Broadcast*: Resultados numéricos.

30%, e isso se deve ao seu funcionamento baseado em tabelas de vizinhos, que torna a propagação dos pacotes muito vulnerável a mudanças nos estados dos nós. Assim, o TBF apresenta um alto índice de quebra de trajetórias e, conseqüentemente, pior cobertura. Na Figura 5.2(b) podemos observar que o *flooding* transmite o mesmo número de pacotes que envia, e que o *gossiping* transmite mais pacotes do que ambos os protocolos baseados em curva. Vale ressaltar que o TBF transmite menos pacotes que o TEDD pois os pacotes de *beacons* não foram considerados nos experimentos. As Figuras 5.2(c) e 5.2(d) comprovam os resultados de consumo de energia, vistos na Figura 5.1. A Figura 5.2(e) mostra a razão entre os pacotes recebidos (Rx) e transmitidos (Tx). Podemos ver que os algoritmos baseados em curva apresentam um desempenho bastante superior aos algoritmos baseados em inundação. Enquanto no *flooding*  $Rx = Tx$  e no *gossiping*  $Rx = 2.5Tx$ , no TBF e TEDD  $Rx > 3Tx$ .

Um ponto importante é que, mesmo a rede apresentando uma densidade alta (uma média de 27 vizinhos para cada nó), o *flooding* não apresenta uma cobertura de 100%, mas de 80% no início da simulação. Isso se deve ao fato da topologia da rede ser dinâmica, já que nós adormecem periodicamente para economizar energia.

## 5.4 Broadcast: Evitando Regiões de Baixa Energia

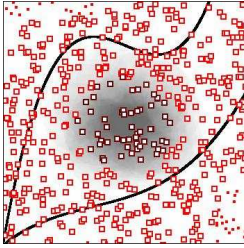
O cenário considerado nesta seção é caracterizado por uma região de baixa energia, localizada no centro da região de sensoriamento, como ilustrado na Figura 5.3. A região crítica é um círculo de raio igual a  $7m$ , cujo centro é o centro da rede. O número de nós existentes dentro dessa região é igual a 53. Nesse cenário, o principal objetivo da disseminação de dados é evitar o fluxo de dados dentro da região crítica de baixa energia. Outros objetivos, tais como elevar a cobertura da rede e reduzir o número de transmissões, também são desejáveis. O critério utilizado para selecionar o melhor conjunto de curvas foi o de *energia máxima*. Na Figura 5.3, é ilustrada a evolução do mapa de energia, assim como da cobertura da rede<sup>1</sup>.

Na Figura 5.3 podemos observar que os algoritmos baseados em curva apresentam um consumo de energia menor na região de baixa energia. O *flooding* e o *gossiping*, por outro lado, matam praticamente todos os nós de baixa energia no instante  $t = 500$ . Os resultados fora da região de baixa energia são análogos aos da Figura 5.1. No entanto, pode-se perceber que o desempenho do *gossiping* foi levemente inferior devido à presença da região de baixa energia.

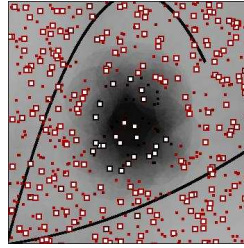
---

<sup>1</sup>Assim como na seção anterior, os quadrados brancos grandes representam os nós que receberam o pacote disseminado, e os quadrados pretos pequenos indicam os nós que não receberam o pacote disseminado.

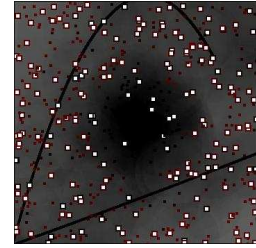




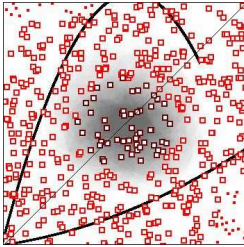
(a) TEDD, T = 0s: Cd = 30%, Ed = 40%, Cf = 50%, Ef = 100%.



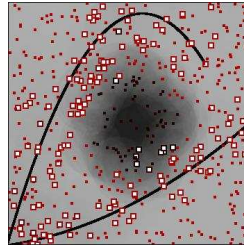
(b) TEDD, T = 500s: Cd = 20%, Ed = 12%, Cf = 48%, Ef = 60%.



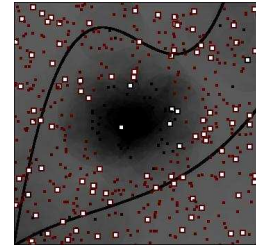
(c) TEDD, T = 1000s: Cd = 0%, Ed = 0%, Cf = 40%, Ef = 22%.



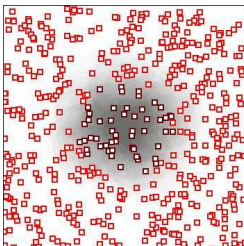
(d) TBF, T = 0s: Cd = 20%, Ed = 40%, Cf = 20%, Ef = 100%.



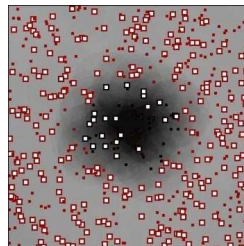
(e) TBF, T = 500s: Cd = 20%, Ed = 12%, Cf = 18%, Ef = 65%.



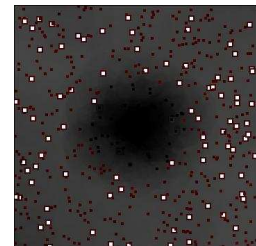
(f) TBF, T = 1000s: Cd = 0%, Ed = 0%, Cf = 16%, Ef = 31%.



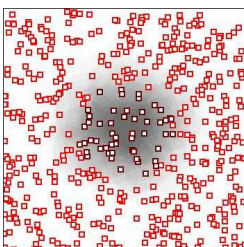
(g) Gossiping, T = 0s: Cd = 30%, Ed = 40%, Cf = 25%, Ef = 100%.



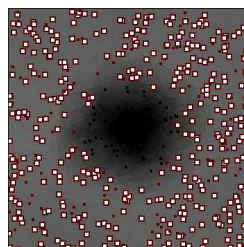
(h) Gossiping, T = 500s: Cd = 20%, Ed = 8%, Cf = 20%, Ef = 60%.



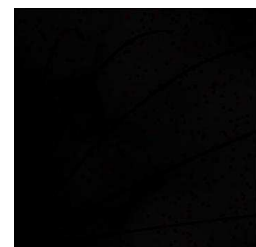
(i) Gossiping, T = 1000s: Cd = 0%, Ed = 0%, Cf = 20%, Ef = 22%.



(j) Flooding, T = 0s: Cd = 80%, Ed = 40%, Cf = 80%, Ef = 100%.



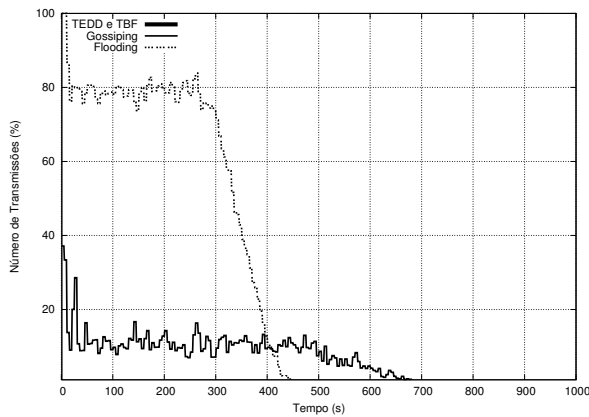
(k) Flooding, T = 500s: Cd = 0%, Ed = 0%, Cf = 80%, Ef = 35%.



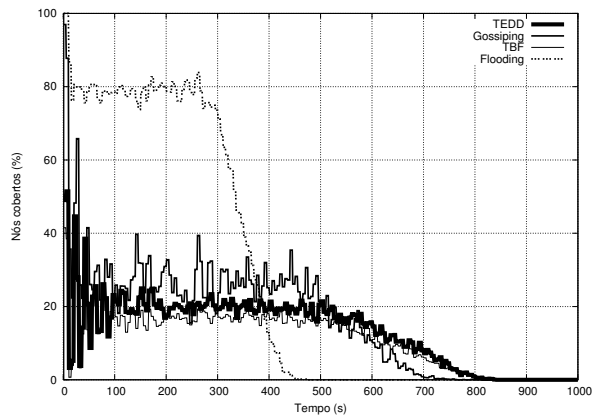
(l) Flooding, T = 1000s: Cd = 0%, Ed = 0%, Cf = 0%, Ef = 0%.

Figura 5.3: *Broadcast com região de baixa energia*: Mapa de energia e cobertura da rede (T=tempo, Cd/Cf, Ed/Ef = Cobertura, Energia dentro/fora da região de baixa energia).

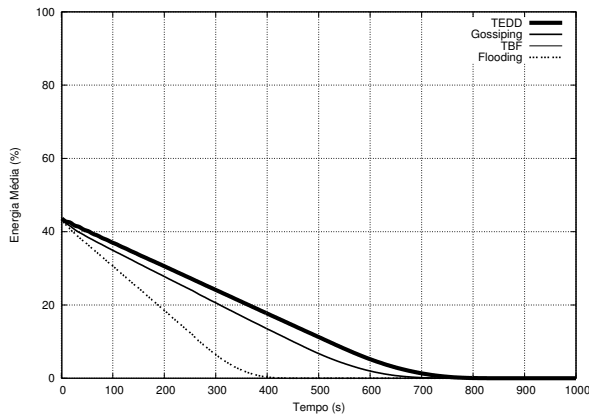
A Figura 5.4 apresenta os resultados numéricos obtidos para os nós localizados dentro da região de baixa energia. Na Figura 5.4(a), observa-se que os protocolos baseados em curvas eliminam completamente transmissões de dados dentro da região crítica. Na Figura 5.4(b), podemos ver que a cobertura dos protocolos baseados em curvas é similar à cobertura do *gossiping*. Nas Figuras 5.4(c) e 5.4(d), podemos ver que TEDD e TBF prolongam o tempo de vida dos nós localizados na região de baixa energia em até 60% em relação ao *flooding*. A Figura 5.5 apresenta os resultados de simulação fora da região de baixa energia. Podemos ver que esses resultados são análogos aos apresentados nas Figuras 5.2(a) a 5.2(e), com a diferença de que o desempenho do *gossiping* se mostra um pouco inferior devido à presença da região de baixa energia, e a razão entre os pacotes recebidos (Rx) e transmitidos (Tx) do TEDD e do TBF apresenta uma melhora em relação a esse cenário.



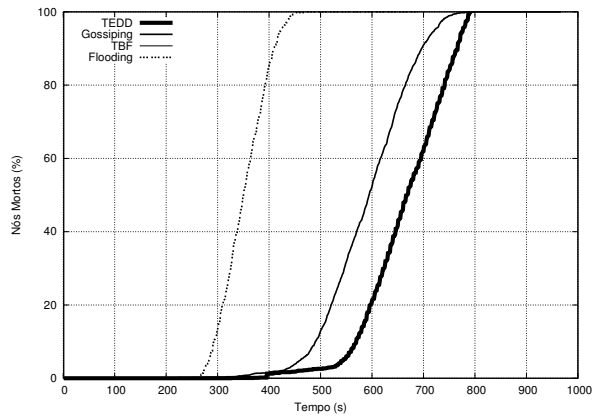
(a) Número de pacotes transmitidos (TEDD e TBF não transmitem pacotes).



(b) Porcentagem de nós cobertos.

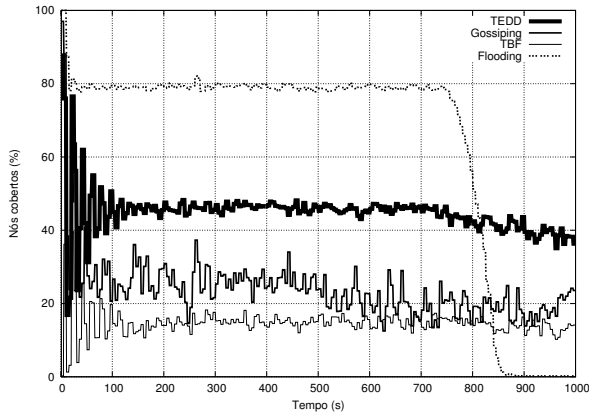


(c) Energia Média.

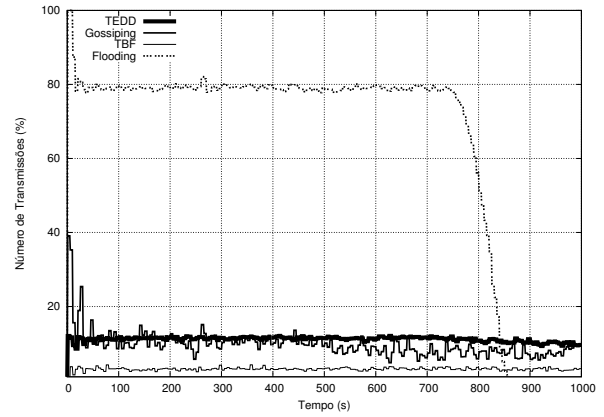


(d) Porcentagem de nós mortos.

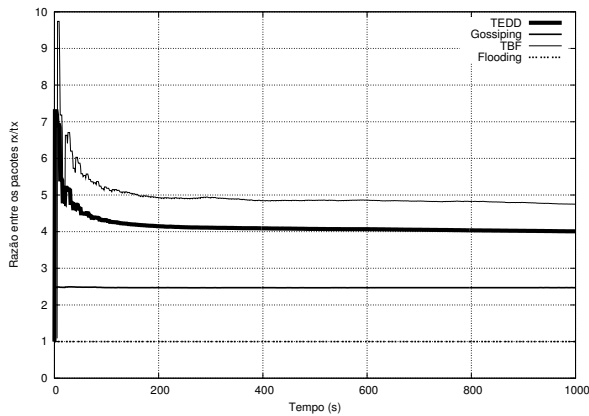
Figura 5.4: *Broadcast* com região de baixa energia: Resultados dentro da região de baixa energia. (As curvas correspondentes aos desempenhos do TEDD e do TBF coincidem nas Figuras (a), (c) e (d).)



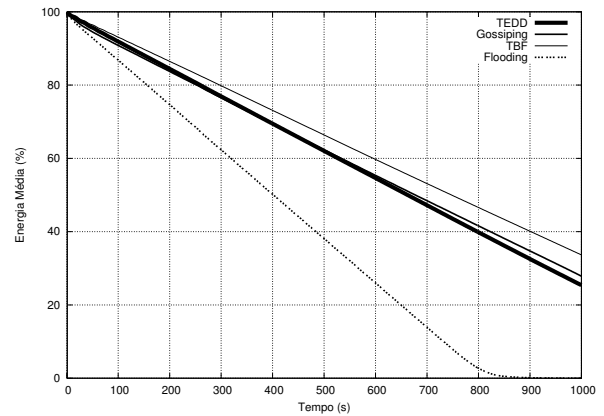
(a) Porcentagem de nós cobertos.



(b) Número de pacotes transmitidos.



(c) Razão entre os pacotes recebidos/transmitidos.



(d) Energia Média.

Figura 5.5: *Broadcast* com região de baixa energia: Resultados fora da região de baixa energia.

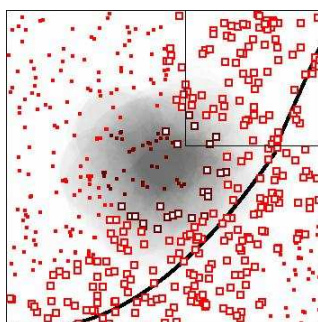
## 5.5 Multicast: Disseminação para uma Área-Alvo

Nesta seção é avaliado um cenário que contém uma área inicial de baixa energia, localizada no centro da rede, e o nó-monitor deseja disseminar informações para os nós localizados em uma área-alvo, localizada no canto superior direito da região de sensoriamento, como mostrado na Figura 5.6. Os dados são transmitidos do nó-monitor para a área-alvo ao longo de um polinômio-ligador. Dentro da área-alvo, o processo de disseminação é semelhante ao *broadcast* na rede toda, porém é limitado àquela região.

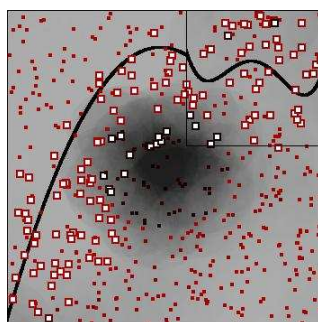
Neste caso, a disseminação de dados possui três objetivos de igual relevância: garantir a melhor cobertura possível dentro da região-alvo; garantir o menor número possível de transmissões realizadas na rede como um todo; e prolongar o tempo de vida dos nós localizados na região de baixa energia. Nesse cenário, o desempenho dos algoritmos baseados em curvas é comparado com o de uma versão dinâmica do *gossiping*, descrita a seguir. No *gossiping* dinâmico, quando um nó localizado fora da região alvo recebe um pacote, ele o propaga com uma probabilidade igual a 0.4. Por outro lado, se o nó estiver dentro da região alvo, ele propaga o pacote com probabilidade igual a 1. Nesse caso, dentro da região alvo, o *gossiping* opera igual ao *flooding*.

Na Figura 5.6, podemos observar que o TEDD apresenta melhor cobertura dentro da área-alvo e o menor gasto de energia, tanto na região de baixa energia (50% menos que o *gossiping* no instante  $t=500$ ), como na rede inteira. TBF apresenta um consumo de energia mais eficiente do que os demais algoritmos, pois os pacotes de *beacons* não são considerados nos cálculos. Além disso, podemos observar que o TBF não apresenta uma cobertura boa dentro da área-alvo. Isso se deve à alta taxa de quebras de trajetórias, causada pelo seu funcionamento baseado em tabelas de vizinhos.

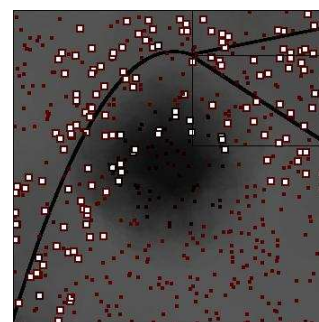
A Figura 5.7(a) ilustra a cobertura dentro da região-alvo, comprovando os resultados discutidos acima. A Figura 5.7(b) apresenta o número de transmissões na rede como um todo. Podemos observar que os algoritmos baseados em curva apresentam um desempenho melhor de acordo com essa métrica, fato mais uma vez retratado na Figura 5.7(c), que mostra a razão entre a cobertura dentro da região alvo e o número total de pacotes transmitidos na rede toda. Apesar do TEDD alcançar a razão de aproximadamente um, o seu resultado é bom, porque o pacote deve ser deslocado do canto inferior esquerdo até o canto superior direito - onde a disseminação é realizada. Outro ponto importante é a existência da região de baixa energia no centro da rede, que obriga o TEDD a percorrer um caminho maior até chegar na região alvo, evitando o roteamento na região crítica. Além disso, verifica-se que o valor da razão obtida pelo TEDD é significativamente melhor que a dos demais protocolos avaliados.



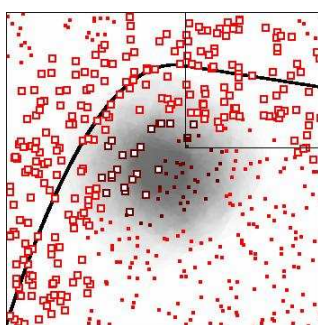
(a) TEDD,  $T = 0s$ : Cd = 50%, Ed = 40%, Et = 100%.



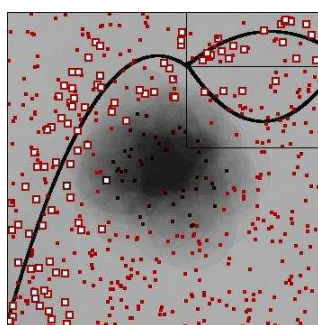
(b) TEDD,  $T = 500s$ : Cd = 38%, Ed = 10%, Et = 60%.



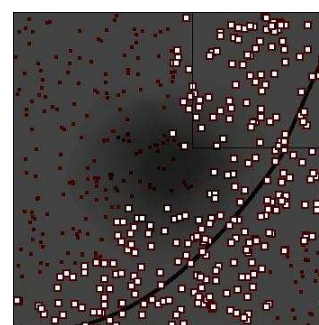
(c) TEDD,  $T = 1000s$ : Cd = 30%, Ed = 0%, Et = 25%.



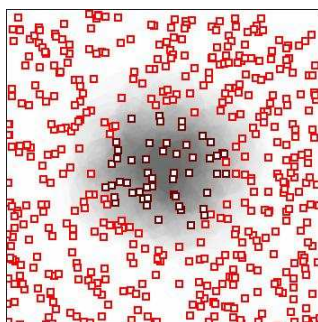
(d) TBF,  $T = 0s$ : Cd = 10%, Ed = 40%, Et = 100%.



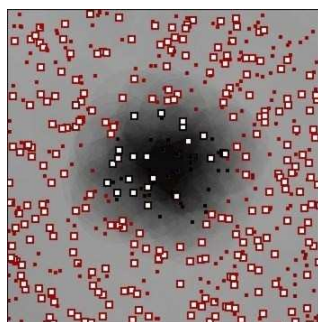
(e) TBF,  $T = 500s$ : Cd = 10%, Ed = 10%, Et = 65%.



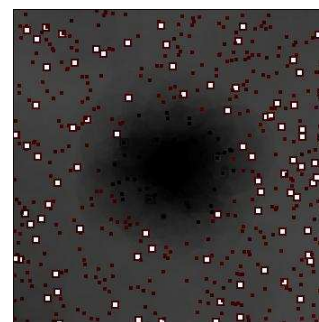
(f) TBF,  $T = 1000s$ : Cd = 10%, Ed = 0%, Et = 33%.



(g) *Gossiping* dinâmico,  $T = 0s$ : Cd = 30%, Ed = 40%, Et = 100%.



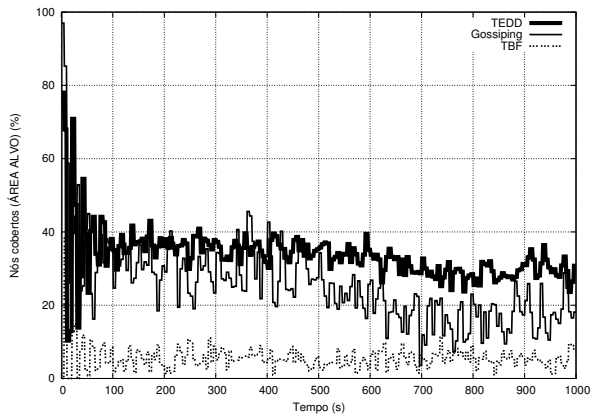
(h) *Gossiping* dinâmico,  $T = 500s$ : Cd = 30%, Ed = 5%, Et = 50%.



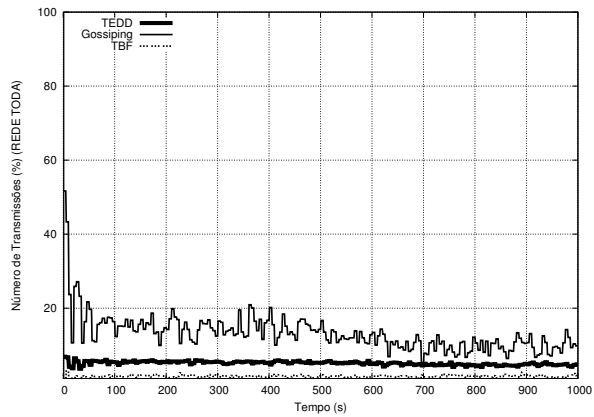
(i) *Gossiping* dinâmico,  $T = 1000s$ : Cd = 20%, Ed = 0%, Et = 17%.

Figura 5.6: *Multicast*: Mapa de energia e cobertura da rede ( $T$ =tempo, Cd = Cobertura dentro da área-alvo, Ed= Energia dentro da região de baixa energia, Et = Energia na rede toda).

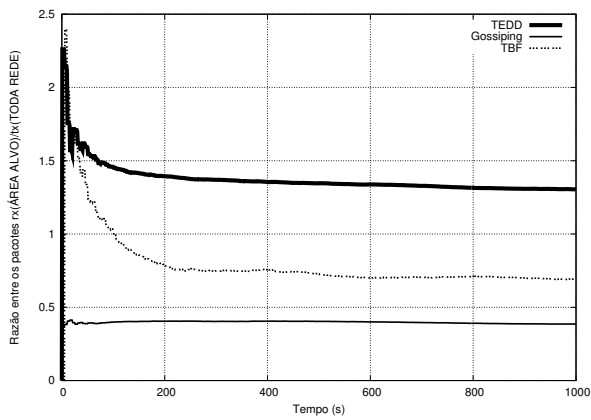
A Figura 5.7(d) apresenta o número de pacotes transmitidos dentro da região de baixa energia. Mais uma vez, podemos comprovar que os protocolos baseados em curva são capazes de eliminar o tráfego de dados em regiões de baixa energia.



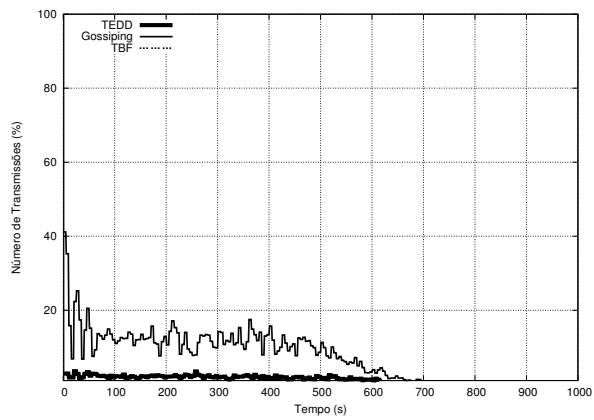
(a) Porcentagem de nós cobertos dentro da região alvo.



(b) Número de pacotes transmitidos pela rede toda.



(c) Razão entre os pacotes recebidos (região alvo) / transmitidos (rede toda).



(d) Número de pacotes transmitidos dentro da região de baixa energia.

Figura 5.7: *Multicast*: Resultados numéricos.

## 5.6 Conclusões

Através da análise dos resultados experimentais, podemos concluir que protocolos baseados em curva nem sempre são a melhor opção quando o objetivo da disseminação de dados é maximizar a cobertura. No entanto, se a razão  $R_x/T_x$  for um fator mais importante, a disseminação baseada em curvas se torna uma alternativa atraente.

Em cenários onde a rede apresenta regiões de baixa energia claramente delimitadas, comprovou-se que os algoritmos baseados em curva são os mais



adequados, já que o tempo de vida dos nós de baixa energia pode ser estendido em até 60% quando comparado ao *flooding*. Em cenários de *multicast*, onde a disseminação é voltada para uma determinada região-alvo, os algoritmos baseados em curva também se mostraram superiores, pois evitam o tráfego de dados fora da região-alvo, conduzindo as mensagens, através de um processo de tunelamento ao longo de um polinômio-escoador, até a área-alvo, onde então é feito um *broadcast*, limitado às fronteiras da área-alvo.

Outra conclusão decorrente dos resultados de simulação é a não adequação do mecanismo de tabelas de vizinhos, presente no TBF, ao ambiente de redes de sensores sem fio. Assim, recomenda-se o uso de protocolos baseados no receptor (*receiver-based*), ao invés de protocolos baseados no remetente (*sender-based*).

O desempenho dos algoritmos baseados em curva foi comparado somente aos algoritmos baseados em inundação, devido ao fato deste trabalho estar abordando o problema de disseminação de dados partindo do nó-monitor em direção à rede, e não o problema de coleta de dados, que pressupõe um fluxo de dados partindo dos nós-sensores em direção ao nó-monitor. Os algoritmos de roteamento apresentados no Capítulo 2, tais como SPIN, DRP, Difusão Direcionada, LEACH e TTDD, são voltados para a coleta de dados, ao invés da disseminação. Assim, os mesmos não foram utilizados como base de comparação nos experimentos apresentados neste capítulo.





---

## Conclusões e Trabalhos Futuros

---

**O** *Problema de Geração de Curvas de Roteamento em Redes de Sensores Sem Fio* é um problema que, pelo nosso conhecimento, até o presente momento não teve nenhuma proposta de solução na literatura. Portanto, a solução proposta nesta dissertação é uma tentativa pioneira de resolvê-lo. O problema consiste de várias etapas. Primeiramente, devem ser definidas as entradas para o processo. Em seguida, deve ser definida uma forma de representação para as trajetórias, juntamente com a técnica de ajuste de curvas. Em seguida, deve ser definida uma arquitetura, capaz de direcionar o processo de geração de curvas de acordo com os objetivos do algoritmo de roteamento.

Nós decidimos utilizar como entrada o mapa de energia da rede por duas razões principais:

- Importância do recurso energia no contexto de redes de sensores sem fio;
- Existência de uma técnica eficiente de obtenção do mapa.

Neste trabalho, o mapa de energia foi obtido através do modelo de predição baseado em cadeias de Markov, proposto em [29]. O custo de obtenção do mapa de energia utilizando essa abordagem é aceitável para redes de sensores sem fio, e o mesmo é analisado em detalhes em [29]. Todavia, destaca-se que esse custo deve ser amortizado e distribuído entre todas as aplicações que o utilizarem.

Tendo gerado o mapa de energia, várias estratégias podem ser utilizadas para selecionar um subconjunto do total de coordenadas usadas para servir de entrada para o algoritmo de ajuste de curva. Neste trabalho, os pontos de

entrada para o processo de ajuste de curva foram escolhidos baseando-se na combinação dos critérios de energia e densidade.

Nós decidimos optar por duas formas de representação de trajetórias: funcional (polinômios) e equacional (seções cônicas). A vantagem da representação funcional é a facilidade com que o valor da variável dependente  $y$  pode ser calculado para qualquer ponto do domínio da variável independente  $x$ . A vantagem da representação equacional é o seu maior poder de representação, que engloba curvas, tais como seções cônicas, e a sua maior estabilidade computacional. Além disso, esse método é tão compacto quanto a representação funcional, em termos de número de parâmetros necessários para codificar uma curva. Essas duas formas de representação se mostraram suficientemente expressivas para garantir a flexibilidade necessária para propagar pacotes por áreas de maior energia. Como método de ajuste de curvas, nós optamos pela regressão linear múltipla. As razões para tal escolha são a simplicidade da técnica e a sua aplicação direta para o tipo de curvas e o tipo de dados de entrada tratados aqui.

A arquitetura de geração de trajetórias proposta neste trabalho apresenta pequenas variações de acordo com o tipo de disseminação. Se a disseminação for do tipo *broadcast*, os pontos de entrada são selecionados a partir do mapa de energia inteiro. Caso a disseminação seja do tipo *multicast*, dois conjuntos de pontos são selecionados: um dentro da área compreendida entre o nó-monitor e o início da área-alvo e o outro somente dentro da área-alvo. O primeiro conjunto de pontos é usado para realizar o ajuste de uma curva chamada *polinômio-ligador*. O polinômio-ligador tem a função de “túnel” entre o nó-monitor e a área-alvo. O segundo conjunto de pontos é utilizado para realizar a disseminação de dados dentro da área-alvo. Caso a disseminação seja do tipo *unicast*, a seleção de pontos é feita a partir da área compreendida entre o nó-monitor e o nó-destino. Nesse caso, existe apenas uma curva de roteamento, também denominada polinômio-ligador, que liga o ponto de origem ao ponto de destino.

A solução para o *Problema de Geração de Curvas de Roteamento em Redes de Sensores Sem Fio* proposta nesta dissertação apresenta as seguintes características:

- *Geração Dinâmica*. As curvas são geradas dinamicamente, com base na versão mais atual do mapa de energia de que o nó-monitor dispõe.
- *Consideração do fator energia*. Todo o processo é baseado no mapa de energia da rede.
- *Controle de cobertura*. A seleção do melhor conjunto de curvas é feita de forma a maximizar o número de nós cobertos pelas curvas.

- *Controle de quebras de trajetoria.* A quebra de trajetórias, devido a trechos não cobertos por nenhum nó, é minimizada através da estratégia de seleção de pontos de entrada, que é baseada na densidade e na quantidade de energia em cada ponto que serve de entrada para o procedimento de ajuste de curvas.
- *Controle de número de parâmetros transmitidos.* O número máximo de parâmetros transmitidos pode ser limitado pelo número máximo de setores de rede e pelo grau dos polinômios.
- *Adaptabilidade aos requisitos da aplicação.* A aplicação pode definir os critérios de avaliação para os ajustes de curva. Neste trabalho foram feitos testes considerando o critério de *energia máxima* e de *cobertura máxima*, no entanto, outros parâmetros poderiam facilmente ser aplicados.
- *Custo computacional.* A solução proposta utiliza uma abordagem centralizada, ou seja, todos os cálculos são realizados no nó-monitor. Assume-se que o mesmo possui reservas ilimitadas de energia e poder computacional. Portanto, o custo de gerar as curvas pode ser considerado irrelevante do ponto de vista da rede.

Os experimentos foram realizados através de simulação de um ambiente de disseminação de dados em uma rede de sensores sem fio. Uma série de topologias de rede foram geradas aleatoriamente. Variou-se tanto a localização dos nós-sensores, como a distribuição de energia nos mesmos. Para cada topologia, simulou-se uma série de disseminações de dados realizadas por um nó-monitor. Dois tipos de disseminação foram analisados: *broadcast* e *multicast*. O mecanismo de geração de curvas foi usado em conjunto com dois algoritmos de propagação em curva: TBF e TEDD. O desempenho desses algoritmos foi comparado ao desempenho de dois protocolos baseados em inundação: *flooding* e *gossiping*.

Através da análise dos resultados experimentais, pudemos concluir que protocolos baseados em curva nem sempre são a melhor opção quando o objetivo da disseminação de dados é maximizar a cobertura. No entanto, se a razão  $Rx/Tx$  for um fator mais importante, a disseminação baseada em curvas se torna uma alternativa atraente. Em cenários onde a rede apresenta regiões de baixa energia claramente delimitadas, comprovou-se que os algoritmos baseados em curva são os mais adequados, já que o tempo de vida dos nós de baixa energia pode ser estendido em até 60% em relação ao *flooding*. Em cenários de *multicast*, onde a disseminação é voltada para uma determinada região-alvo, os algoritmos baseados em curva também se mostraram superiores, pois evitam o tráfego de dados fora da região-alvo, conduzindo os mesmos

através de um processo de tunelamento, ao longo de um polinômio-escoador, até a área-alvo, onde então é feito um *broadcast*, limitado às fronteiras da área-alvo.

As contribuições desse trabalho foram publicadas nas seguintes revistas: *IEEE Communications Magazine (Ad Hoc and Sensor Networks Series)* [11] e *Journal on Selected Areas in Communication (Special Issue on Autonomics)* [8]; e conferências: *XXIII Simpósio Brasileiro de Redes de Computadores (SBRC)* [12] e *The Second IEEE Annual Conference on Wireless On demand Network Systems and Services* [10].

Várias questões relacionadas ao *Problema de Geração de Curvas de Roteamento em Redes de Sensores Sem Fio* ainda permanecem em aberto. Entre elas:

- Utilização de outros mapas como entrada, além do mapa de energia.
- Melhoria do processo e setorização, permitindo setores com ângulos diferentes.
- Proposta de um esquema para codificação das trajetórias, de forma a minimizar a quantidade de dados transmitidos em cada pacote.
- Proposta de uma solução distribuída para o *Problema de Geração de Curvas de Roteamento em Redes de Sensores Sem Fio*.
- Geração de Curvas para coleta de dados.

Temos a intenção de abordar esses problemas em nossos trabalhos futuros.

# Referências Bibliográficas

---

- [1] *Alert: Automated local evaluation in real time.* <http://www.alertsystems.org>.
- [2] *Corie: Environmental observation and forecasting system for the columbia river.* <http://www.ccalmr.ogi.edu/CORIE/>.
- [3] *Tinyos: An open-source operating system designed for wireless embedded sensor networks.* <http://www.tinyos.net/>.
- [4] I. F. AKYILDIZ, W. SU, Y. SANKARASUBRAMANIAM, AND E. CAYIRCI, *Wireless sensor networks: A survey*, *Computer Networks*, 38 (2002), pp. 393–422.
- [5] M. BERMAN, *Estimating the parameters of a circle when angular differences are known*, *Journal Royal Stat. Soc.*, 32 (1983), pp. 1 – 6.
- [6] D. A. COFFIN, D. J. V. HOOK, S. M. MCGARRY, AND S. R. KOLEK, *Declarative ad-hoc sensor networking*, in *SPIE Integrated Command Environments Conference*, 2000, pp. 109 – 120.
- [7] M. D. VAL MACHADO, R. A. F. MINI, AND A. A. F. LOUREIRO, *Um modelo de amostragem para a construção do mapa de energia em redes de sensores sem fio*, in *VI Workshop de Comunicação sem Fio e Computação Móvel*, Fortaleza, Brazil, October 2004.
- [8] M. DO V. MACHADO, O. GOUSSEVSKAIA, R. A. MINI, A. A. LOUREIRO, G. R. MATEUS, AND J. M. NOGUEIRA, *Data dissemination in autonomic wireless sensor networks (aceito para publicação)*, *Journal on Selected Areas in Communication, Special Issue on Autonomics*, (2005).
- [9] M. DO VAL MACHADO, *Disseminação de dados baseada em trajetória e energia para redes de sensores sem fio*, Master's thesis, Federal University of Minas Gerais, 2005.

- [10] M. DO VAL MACHADO, O. GOUSSEVSKAIA, R. A. MINI, C. G. REZENDE, A. A. LOUREIRO, G. R. MATEUS, AND J. M. NOGUEIRA, *Data dissemination using the energy map*, in The Second IEEE Annual Conference on Wireless On demand Network Systems and Services, January 2005.
- [11] O. GOUSSEVSKAIA, M. DO V. MACHADO, R. A. MINI, A. A. LOUREIRO, G. R. MATEUS, AND J. M. NOGUEIRA, *Data dissemination based on the energy map*, IEEE Communications Magazine, Ad Hoc and Sensor Networks Series, 43 (2005), pp. 134–143.
- [12] O. GOUSSEVSKAIA, M. DO VAL MACHADO, R. A. MINI, C. G. REZENDE, A. A. LOUREIRO, G. R. MATEUS, AND J. M. NOGUEIRA, *Trajectory and energy-based data dissemination in wireless sensor networks*, in XXIII Simpósio Brasileiro de Redes de Computadores (SBRC), May 2005.
- [13] Z. HAAS, J. HALPERN, AND L. LI, *Gossip-based ad hoc routing*, in Proceedings of the IEEE Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2002, vol. 3, 2002, pp. 1707– 1716.
- [14] W. R. HEINZELMAN, A. CHANDRAKASAN, AND H. BALAKRISHNAN, *Energy-efficient communication protocol for wireless microsensor networks*, in Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8, IEEE Computer Society, January 2000, pp. 3005 – 3014.
- [15] W. R. HEINZELMAN, J. KULIK, AND H. BALAKRISHNAN, *Adaptive protocols for information dissemination in wireless sensor networks*, in MOBICOM 99, Seattle, WA USA, 1999, pp. 174–185.
- [16] M. R. HESTENES AND E. STIEFEL, *Methods of conjugate gradients for solving linear systems*, Journal Res. N. B. S., 49 (1952), pp. 409 – 436.
- [17] M. HORTON, D. CULLER, K. PISTER, J. HILL, R. SZEWCZYK, AND A. WOO, *Mica: The commercialization of microsensor motes*, Sensors Online, 19 (2002), pp. 40 – 48.
- [18] L. HUGHES, O. BANYASAD, AND E. HUGHES, *Cartesian routing*, Computer Networks, 34 (2000), pp. 455–466.
- [19] C. INTANAGONWIWAT, R. GOVINDAN, AND D. ESTRIN, *Directed diffusion: A scalable and robust communication paradigm for sensor networks*, in Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, Boston, MA USA, 2000, pp. 56–67.

- [20] C. INTANAGONWIWAT, R. GOVINDAN, D. ESTRIN, J. HEIDEMANN, AND F. SILVA, *Directed diffusion for wireless sensor networking*, IEEE/ACM Trans. Netw., 11 (2003), pp. 2–16.
- [21] D. B. JOHNSON AND D. A. MALTZ, *Dynamic source routing in ad hoc wireless networks*, in Mobile Computing, Imielinski and Korth, eds., vol. 353, Kluwer Academic Publishers, 1996.
- [22] J. M. KAHN, R. H. KATZ, AND K. S. J. PISTER, *Next century challenges: Mobile networking for smart dust*, in In Proceedings of MOBICOM, Seattle, 1999, pp. 271–278.
- [23] J. KULIK, W. HEINZELMAN, AND H. BALAKRISHNAN, *Negotiation-based protocols for disseminating information in wireless sensor networks*, in Wireless Networks, vol. 8, Hingham, MA, USA, 2002, Kluwer Academic Publishers, pp. 169–185.
- [24] C. L. LAWSON AND R. J. HANSON, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, N. J., 1974.
- [25] B. M. AND P. SOMLO, *Efficient procedures for fitting circles and ellipses with application to sliding termination measurements*, IEEE Inst. & Meast., IM-35 (1986), pp. 31–35.
- [26] A. MAINWARING, D. CULLER, J. POLASTRE, R. SZEWCZYK, AND J. ANDERSON, *Wireless sensor networks for habitat monitoring*, in WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, New York, NY, USA, 2002, ACM Press, pp. 88–97.
- [27] MICA2, *MTS/MDA sensor and data acquisition boards user's manual*. [www.xbow.com](http://www.xbow.com), 2004.
- [28] R. A. F. MINI, *Mapa de energia baseado em predição para redes de sensores sem fio*, PhD thesis, Departamento de Ciência da Computação - Universidade Federal de Minas Gerais, Jan - 2004.
- [29] R. A. F. MINI, M. DO VAL MACHADO, A. A. F. LOUREIRO, AND B. NATH, *Prediction-based energy map for wireless sensor networks*, Ad Hoc Networks Journal, 3 (2005), pp. 235–253.
- [30] R. A. F. MINI, B. NATH, AND A. A. F. LOUREIRO, *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, October 2002.

- [31] —, *Prediction-based approaches to construct the energy map for wireless sensor networks*, in XXI Simpósio Brasileiro de Redes de Computadores (SBRC), Natal, RN, Brasil, 19 a 23 de Maio 2003.
- [32] D. C. MONTGOMERY, E. A. PECK, AND G. G. VINING, *Introduction to Linear Regression Analysis*, John Wiley & Sons, 2001.
- [33] B. NATH AND D. NICULESCU, *Routing on a curve*, ACM SIGCOMM Computer Communication Review, 1 (2003), pp. 155–160.
- [34] D. NICULESCU AND B. R. BADRINATH, *Ad hoc positioning system (APS)*. GLOBECOM 2001, November 2001.
- [35] D. NICULESCU AND B. NATH, *Trajectory-Based Forwarding and its Applications*, in MOBICOM 03, USA, 2003, pp. 260–272.
- [36] NS2, *The network simulator*. [www.isi.edu/nsnam/ns](http://www.isi.edu/nsnam/ns), 2002.
- [37] L. M. S. C. OF THE IEEE COMPUTER SOCIETY, *Wireless LAN medium access control (MAC) and physical layer (PHY) specification, IEEE Std 802.11*, 1997.
- [38] C. C. PAIGE AND M. A. SAUNDERS, *LSQR: Sparse Linear Equations and Least Squares Problems*, ACM Transactions on Mathematical Software, 8 (1982), pp. 195 – 209.
- [39] A. SAVVIDES, C.-C. HAN, AND M. B. STRIVASTAVA, *Dynamic fine-grained localization in ad-hoc networks of sensors*, in The 7th Annual International Conference on Mobile Computing and Networking, Rome, Italy, July 2001, pp. 166–179.
- [40] L. SCHWIEBERT, S. K. GUPTA, AND J. WEINMANN, *Research challenges in wireless networks of biomedical sensors*, in The seventh annual international conference on Mobile computing and networking 2001, Rome, Italy, July 2001, pp. 151–165.
- [41] C. SHENE, *Introduction to computing with geometry notes*. <http://www.cs.mtu.edu/shene/COURSES/cs3621/NOTES/notes.html>.
- [42] K. SOHRABI AND G. J. POTTIE, *Performance of a novel self-organization protocol for wireless ad hoc sensor networks*, in IEEE 50th Vehicular Technology Conference, 1999, pp. 1222–1226.
- [43] M. B. SRIVASTAVA, R. R. MUNTZ, AND M. POTKONJAK, *Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving environments*, in Mobile Computing and Networking, 2001, pp. 132–138.



- [44] S.SINGH AND C.S.RAGHAVENDRA, *PAMAS: Power aware multi-access protocol with signaling for ad hoc networks*, ACM Computer Communication Review, 8 (1998), pp. 5–26.
- [45] H. WANG, J. ELSON, L. GIROD, D. ESTRIN, AND K. YAO, *Target classification and localization in habitat monitoring.*, in IEEE ICASSP, 2003.
- [46] H. WANG, D. ESTRIN, AND L. GIROD, *Preprocessing in a tiered sensor network for habitat monitoring*, in EURASIP JASP Special Issue on Sensor Networks, vol. 4, 2003, pp. 392–401.
- [47] B. WARNEKE, M. LAST, B. LIEBOWITZ, AND K. S. J. PISTER, *Smart dust: Communicating with a cubic-millimeter computer*, Computer, 34 (2001), pp. 44–51.
- [48] J. H. WEI YE AND D. ESTRIN, *An energy-efficient MAC protocol for wireless sensor networks.*, in 12th IEEE International Conference on Computer Networks, INFOCOM, June 2002.
- [49] F. YE, H. LUO, J. CHENG, S. LU, AND L. ZHANG, *A two-tier data dissemination model for large-scale wireless sensor networks*, MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking, (2002), pp. 148–159.
- [50] F. YE, H. LUO, S. LU, AND L. ZHANG, *Dissemination protocols for large sensor networks*, Wireless sensor networks, (2004), pp. 109–128.
- [51] F. YE, G. ZHONG, S. LU, AND L. ZHANG, *Gradient broadcast: A robust data delivery protocol for large scale sensor networks*, ACM WINET (Wireless Networks), 11 (2005).
- [52] Y. J. ZHAO, R. GOVINDAN, AND D. ESTRIN, *Residual energy scans for monitoring wireless sensor networks*, in IEEE Wireless Communications and Networking Conference (WCNC'02), Orlando, FL, USA, March 2002, pp. 17–21.