ROTEAMENTO FAVORECENDO AGREGAÇÃO DE DADOS EM REDES DE SENSORES

Jorgito Matiuzzi Stochero

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

Aprovada por:

Prof. José Ferreira de Rezende, Dr.

Prof. Aloysio de Castro Pinto Pedroza, Dr.

Prof. Luís Fernando Gomes Soares, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

JUNHO DE 2004

Stochero, Jorgito Matiuzzi

Roteamento favorecendo agregação de dados em redes de sensores [Rio de Janeiro] 2004

XIV, 93 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia Elétrica, 2004)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Redes de sensores
2. roteamento
3. agregação de dados

I. COPPE/UFRJ II. Título (série)

À minha esposa Cíntia e meus filhos Diego e Bruna.

Agradecimentos

À Universidade Federal do Amazonas, especialmente à Professora Marly Guimarães, pela acolhida e oportunidade de iniciar o curso de mestrado.

Aos professores do Laboratório do Grupo de Teleinformática e Automação (GTA) e do Laboratório de Processamento de Sinais (LPS) da COPPE, pelos conhecimentos transmitidos, especialmente ao professor Eduardo da Silva, pela coordenação acadêmica do curso.

Ao professor José Ferreira de Rezende, pela orientação segura deste trabalho, pela dedicação e disponibilidade em qualquer momento, e sobretudo, pela motivação extra que o seu entusiasmo pela pesquisa me proporcionou.

Aos companheiros do Laboratório do GTA, especialmente Antonio, Ingrid e Celso, pelo esforço que compartilhamos ao longo do trabalho e pela ajuda mútua.

Ao PEE/COPPE, pelas instalações e equipamentos utilizados. À SUFRAMA, pelo financiamento da pesquisa. Ao Exército Brasileiro, pela liberação em tempo parcial para o desenvolvimento deste trabalho.

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

ROTEAMENTO FAVORECENDO AGREGAÇÃO DE DADOS EM REDES DE SENSORES

Jorgito Matiuzzi Stochero

Junho/2004

Orientador: José Ferreira de Rezende
Programa: Engenharia Elétrica

Este trabalho apresenta uma proposta de agregação de dados em redes de sensores sem fio, para economizar energia e aumentar o tempo de vida da rede. Foram estabelecidas rotas que favorecem a fusão de dados de múltiplas fontes, formando uma árvore de agregação com duas características básicas: número suficiente de nós agregadores, para redução do número de mensagens que chega ao destino; e pequena distância dos mesmos às fontes, para redução do número de transmissões.

A proposta foi implementada como uma modificação do protocolo de roteamento da Difusão Direcionada, escolhido por ser centrado em dados e por usar endereçamento baseado em atributos, facilitando assim o processamento distribuído na rede. Ao protocolo original, foi adicionado um módulo de software, chamado de PAR (Roteamento Privilegiando Agregação), usado para construir a árvore de agregação.

A proposta foi avaliada por meio de simulação. Para a verificação da eficiência do módulo PAR, foi empregada uma técnica de fusão de dados baseada em inferência bayesiana e o desempenho da rede foi analisado com e sem o PAR. Foram utilizados vários cenários e os resultados evidenciaram o impacto positivo do emprego do módulo PAR na redução do número de transmissões e na conseqüente economia de energia da rede, ao custo de um maior retardo no estabelecimento das rotas.

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

ROUTING FAVORING DATA AGGREGATION IN SENSOR NETWORKS

Jorgito Matiuzzi Stochero
June/2004

Advisor: José Ferreira de Rezende
Department: Electrical Engineering

This work presents a proposal for data aggregation in wireless sensor networks in order to save energy and increase network lifetime. Routes were established to favor data aggregation from multiple sources into common flows, creating an aggregation tree with two primary features: sufficient aggregation nodes to reduce the number of messages that reach the destination; and small distance from these nodes to the sources, to reduce the number of transmissions (hops).

The proposal was implemented as a modification to the Directed Diffusion protocol, chosen because of its data centric and attribute based addressing features, which simplify distributed processing in the network. A software module was added to the original protocol, called PAR (Privileged Aggregation Routing), to build the desired aggregation tree.

The proposal was evaluated by means of simulation. To verify PAR efficiency, a data fusion technique based on bayesian inference was deployed and network performance was analyzed with and without the PAR module. Many scenarios were used during simulation and the results showed that the PAR module contributed for a significant reduction of transmissions in the network, saving energy, with the cost of a greater delay in route setup.

Lista de Acrônimos

API :	*Application Program Interface*;
CDMA :	*Code Division Multiple Access*;
CMOS :	*Complementary Metal Oxide Semiconductor*;
DARPA :	*Defense Advanced Research Projects Agency*;
EAR :	*Energy Aware Routing*;
GPS :	*Global Positioning System*;
IEEE :	*Institute of Electrical and Electronics Engineers*;
LEACH:	*Low Energy Adaptative Clustering Hierarchy*;
LWIM :	*Low power Integrated Microsensors*;
MAC :	*Medium Access Control*;
MANET:	*Mobile Ad Hoc Network*;
MEMS :	*Micro Electro Mechanical Systems*;
MRAM :	*Magnetoresistive Random Access Memory*;
NS-2 :	*Network Simulator-2*;
PAR :	*Privileged Aggregation Routing*;
PDA :	*Personal Digital Assistant*;
PR :	*Positive Reinforcement*;
RSSF :	Rede de Sensores Sem Fio;
SCADDS :	*Scalable Coordination Architectures for Deeply Distributed Systems*;
SMAC :	*Sensor Medium Access Control*;
SPIN :	*Sensor Protocols for Information via Negotiation*;
TDMA :	*Time Division Multiple Access*;
VINT :	*Virtual Internetwork Testbed*;
WINS :	*Wireless Integrated Micro Sensors*;

Sumário

Resumo **v**

Abstract **vi**

Lista de Acrônimos **vii**

Lista de Figuras **xi**

Lista de Tabelas **xiii**

1 Introdução **1**

1.1 Motivação 1

1.2 Objetivos 3

1.3 Trabalhos Relacionados 4

1.4 Organização da Dissertação 5

2 Conceitos Básicos **6**

2.1 Redes Móveis Sem Fio 7

2.2 Redes de Sensores 9

2.3 Arquitetura 12

2.4 Pilha de Protocolos 17

2.4.1 Camada Física 17

2.4.2 Camada de Enlace 18

2.4.3 Camada de Rede 20

2.5 Processamento Distribuído com Agregação de Dados 25

2.5.1 Roteamento Centrado em Dados 26

2.5.2 Endereçamento Baseado em Atributos 28

2.5.3 Técnicas de Agregação de Dados 29

3 Mecanismo de Roteamento para Agregação de Dados em RSSF 31

3.1 Arquitetura Distribuída Baseada em Filtros 32

3.1.1 Difusão Direcionada 33

3.1.2 Regras de Casamento 34

3.1.3 Filtros 35

3.2 Roteamento Privilegiando a Agregação (Filtro PAR) 36

3.2.1 Configuração do Filtro 37

3.2.2 Primeira Fase: Armazenamento e Comparação de Mensagens . 41

3.2.3 Segunda Fase: Envio ao Filtro Gradiente 44

3.3 Considerações Finais 45

4 Simulações e Resultados 47

4.1 Ambiente de Simulação 47

4.1.1 Cenários de Simulação 49

4.2 Análise da Árvore de Agregação 51

4.3 Análise dos Parâmetros de Agregação 54

4.3.1 Efeito da Densidade de Nós na Rede 56

4.3.2 Efeito do Temporizador na Construção da Árvore de Agregação 58

4.3.3 Efeito da Variação do Número de Fontes 60

4.3.4 Efeitos Colaterais 61

4.4 Análise do Consumo de Energia 64

4.4.1 Número de Mensagens no Sorvedouro 64

4.4.2 Número Total de Transmissões na Rede 66

4.5 Considerações Finais 69

5 Conclusões e Trabalhos Futuros 71

5.1 Conclusões 72

5.2 Sugestões para Trabalhos Futuros 74

Referências Bibliográficas 75

Lista de Figuras

2.1 Rede de Sensores 10

2.2 Arquitetura de um Sensor 14

2.3 Famílias de Protocolos de Roteamento em RSSFs. 21

3.1 Filtro PAR 37

3.2 Agregação duplicada (***I*** e ***A***) de mensagens originadas em F_1 e F_2. . . 39

3.3 Supressão da agregação em ***A*** de duas mensagens já agregadas em ***I***. 40

3.4 Exemplo de nó agregador 42

3.5 Exemplo de Descarte de mensagem 43

3.6 Propagação pelo Filtro Gradiente 44

3.7 Propagação pelo Filtro PAR 45

4.1 Cenário de Simulação 50

4.2 Rotas Mais Comuns - Difusão Pura. 51

4.3 Rotas mais comuns - Filtro PAR. 52

4.4 Efeitos da Densidade dos nós na descoberta de nós agregadores. . . . 56

4.5 Efeitos da Densidade dos nós distância das agregações às fontes. . . . 57

4.6 Efeitos do Temporizador em grade de 10 x 10 nós. 58

4.7 Efeitos do Temporizador em grade de 12 x 12 nós. 59

4.8 Efeitos do Temporizador em grade de 14 x 14 nós. 60

4.9 Efeitos do Temporizador em grade de 16 x 16 nós. 60

4.10 Parâmetros para Agregação em Função do Número de Fontes. 61

4.11 Tempo de Descoberta de Rotas. 62

4.12 Tamanho Médio das Rotas Fonte/Sorvedouro. 63

4.13 Efeitos da Densidade dos nós no número de mensagens no sorvedouro. 65

4.14 Efeitos do Temporizador no consumo de energia. 66

4.15 Efeitos do Temporizador no consumo de energia. 66

4.16 Número de mensagens no sorvedouro em função do número de fontes. 67

4.17 Efeitos da Densidade dos nós no número de transmissões na rede. . . 67

4.18 Efeitos do Temporizador no consumo de energia. 68

4.19 Efeitos do Temporizador no consumo de energia. 68

4.20 Número de transmissões em função do número de fontes. 69

Lista de Tabelas

4.1 Rotas Descobertas - Difusão Pura 53

4.2 Nós Agregadores - Difusão Pura 54

4.3 Rotas Descobertas - Filtro PAR 54

4.4 Nós Agregadores - Filtro PAR 55

Capítulo 1

Introdução

1.1 Motivação

No final dos anos 90, na medida em que o número de componentes de comunicação e processamento de dados embutidos em um único circuito integrado aumentava, surgiu a possibilidade de implementação de dispositivos sensores de baixo custo e pequeno volume. Com a interligação destes dispositivos por enlaces de rádio, eles passaram a realizar tarefas difíceis para um sensor tradicional, como por exemplo, a detecção de pontos de perigo em uma área de desastre.

Estes dispositivos são chamados de nós sensores (ou microsensores) e desde então se iniciou a pesquisa em um novo ramo na tecnologia de redes móveis, conhecido como redes de sensores sem fio (RSSFs). Este ramo de estudo, caracterizado pela habilidade de conectar uma coleção desses nós desassistidos a uma rede fixa, provendo informação de forma confiável em um ambiente colaborativo, é importante para muitas aplicações.

Entre as aplicações possíveis podem ser incluídas: militares, podendo-se citar como exemplo atual o uso de redes de sensores (acústicos, sísmicos, e térmicos) pelo exército norte-americano na localização de guerrilheiros da rede terrorista Al-Kaeda em cavernas do Afeganistão em 2002 [1]; médicas, com a monitoração remota de pacientes e a possibilidade em futuro próximo do uso de sensores dentro do corpo humano; e monitoração de plantas industriais e comerciais para fins de segurança ou interação entre meio-ambiente e ser humano.

Esta diversidade de aplicações atuais e presumidas antecipa os impactos dessa nova tecnologia sobre o cotidiano da vida moderna, e suas possibilidades comerciais foram objeto de diversas reportagens recentes em publicações não técnicas, como as revistas *Business Week* [2] e *Time* [3]. Entretanto, para que tais aplicações se tornem realidade, existem vários obstáculos tecnológicos a serem vencidos e a tecnologia efetivamente amadureça para emprego comercial.

Entre estes obstáculos está a fusão de todos os componentes de um sensor em um único circuito integrado de tamanho diminuto e baixa potência. Outro obstáculo é o desenvolvimento de protocolos de roteamento ad hoc com múltiplos saltos, que estabeleçam comunicações seguras para trânsito de informações com o mínimo consumo de energia possível.

Esta redução no dispêndio de energia tem sido o principal objetivo das pesquisas envolvendo protocolos de comunicação. Uma forma eficaz de se economizar energia é distribuir a carga de processamento em nós intermediários, agregando os dados enquanto são propagados em múltiplos saltos desde os pontos de geração da informação (nós fontes) até os seus pontos de consumo (sorvedouros). Isto permite a redução do número de transmissões necessárias para que a informação seja conduzida ao seu destino. Dessa maneira, aumentando-se os requisitos de processamento nos nós intermediários (de baixo custo energético), pode-se reduzir a utilização dos rádios (de alto custo energético), o que provoca um saldo positivo no balanço da energia da rede [4].

Entretanto, a agregação de dados em redes de sensores ainda é um tópico de pesquisa incipiente, pois os trabalhos têm se limitado a remover mensagens duplicadas originadas na mesma fonte ou ao emprego de mecanismos simples, como o cálculo da média e de valores máximos e mínimos. Técnicas mais avançadas de fusão de dados, bastante empregadas nos sensores tradicionais de grande porte para aumentar a acurácia das medidas, ainda precisam ser experimentadas em redes de sensores. Ao invés de serem empregadas para aumentar a acurácia, que já é uma característica de uma RSSF, elas são processadas com o novo objetivo de manter a acurácia com um menor número de transmissões.

Para que o emprego de técnicas paramétricas de fusão de dados seja eficiente, algumas condições precisam ser estabelecidas, tanto no aspecto temporal, que consiste na definição de *quando* agregar, como no aspecto de localização dos pontos de agregação, consistindo na definição de *onde* agregar, levando ao estabelecimento de uma árvore de agregação na rede. Todos estes aspectos devem ser convenientemente explorados para que se possa extrair o máximo da rede com o menor consumo de energia, mantendo-se a acurácia com o mínimo de custos em processamento.

1.2 Objetivos

Existem atualmente diversas abordagens para o estabelecimento de protocolos de roteamento em redes de sensores. Tendo em vista que os nós da rede trabalham de forma colaborativa para responder a uma tarefa de sensoriamento de um fenômeno comum, e de que a força da RSSF em relação aos sensores tradicionais reside neste fato, um protocolo de roteamento que propicie processamento distribuído em todos os nós envolvidos irá contribuir para o sucesso da tarefa e para a redução do consumo de energia.

Ao mesmo tempo, com o emprego de técnicas de agregação de dados em nós intermediários, será possível reduzir a quantidade de dados circulando sem comprometer a qualidade da informação que chega ao destino. Isto propicia a economia da energia, extremamente limitada nas baterias que alimentam os nós, aumentando o tempo efetivo de utilização da rede.

Sendo assim, o objetivo do trabalho descrito nesta dissertação é estabelecer um protocolo de roteamento que crie empiricamente uma árvore de agregação, por meio da seleção de rotas com nós intermediários com maior potencial para agregação de dados, escolhendo assim onde e quando a agregação deve ocorrer para minimizar o número de transmissões na rede.

Para tal, foi explorada uma arquitetura distribuída e centrada em dados, já existente, chamada de Difusão Direcionada [5] e nela implementado um novo mecanismo para descoberta de rotas.

Este mecanismo visa maximizar a capacidade de agregação de dados da arquitetura, enquanto minimiza os custos em termos do estabelecimento e manutenção da árvore de agregação.

1.3 Trabalhos Relacionados

Existem vários grupos de pesquisa estudando os protocolos de roteamento *ad hoc* específicos para redes de sensores, abordando principalmente a economia de energia como métrica principal de desempenho. Alguns pesquisadores têm proposto protocolos de roteamento [6] cujo objetivo é a construção de uma árvore reversa com várias fontes de dados convergindo para um destino comum onde o usuário se conecta, de forma a obter uma agregação ótima. Entretanto, estes protocolos consomem muita energia devido à necessidade do conhecimento global da topologia da rede.

Outras alternativas, como o roteamento hierárquico baseado em agrupamentos de nós [7], favorecem a agregação dentro de cada grupo. No entanto, estes protocolos apresentam um elevado nível de complexidade na seleção do nó agregador, além da necessidade de balanceamento de energia entre os nós selecionados. Uma variação desta abordagem é a execução do roteamento baseado em cadeias de nós [8], que minimiza a complexidade na seleção dos nós agregadores (todos os elos da cadeia fazem agregação), mas pode provocar perda de sensibilidade na rede.

Outra abordagem envolve protocolos de roteamento centrados em dados, que favorecem o processamento dentro da rede. O EAR (*Energy Aware Routing*) [9] é um exemplo de protocolo que utiliza esse paradigma e não assume nenhum conhecimento da topologia da rede. Neste sentido, ele é bastante similar ao protocolo no qual este trabalho se baseou.

Os impactos da agregação de dados em redes de sensores sem fio também têm sido objeto de pesquisa [6, 10, 11, 12], onde são examinados os aspectos relativos ao consumo de energia, robustez e retardo. Os resultados obtidos mostram que uma significante economia de energia pode ser obtida com a agregação de dados, de forma que o emprego de técnicas adequadas para cada aplicação sobre uma árvore de agregação previamente estabelecida deve proporcionar uma maior economia de energia na rede.

1.4 Organização da Dissertação

Esta dissertação está estruturada em cinco capítulos. O capítulo 2 descreve os conceitos básicos sobre redes de sensores sem fio, especialmente a sua pilha de protocolos e a importância do processamento distribuído na rede. O capítulo 3 apresenta o mecanismo proposto para seleção de rotas privilegiando agregação de dado. O capítulo 4 apresenta as simulações realizadas para verificar o mecanismo e os resultados alcançados. Finalmente, o capítulo 5 traz as conclusões e sugestões para trabalhos futuros.

Capítulo 2

Conceitos Básicos

REDES móveis sem fio diferem de redes tradicionais porque se utilizam de dispositivos computacionais, como PDAs, laptops, câmeras e celulares que impõem novos requisitos para os protocolos de comunicação. Normalmente possuem escassos recursos, como tempo de vida da bateria, capacidade de processamento e memória. Além disso, a rede deve reagir de forma satisfatória a mudanças freqüentes e não anunciadas no ambiente, provocadas pela alta variabilidade da largura de banda disponível, novas posições físicas dos nós e problemas relacionados à segurança, interferência e consumo de energia.

Redes de Sensores Sem Fio (RSSFs) são uma forma muito especializada de rede móvel, onde os dispositivos computacionais são pequenos nós remotos (fontes de informação), capazes de sensoriar, processar e comunicar, que coletam dados do ambiente, interagem uns com os outros e passam a informação adiante até que ela atinja um nó especial (sorvedouro), que pode ser acessado pelo usuário. Devido ao seu pequeno tamanho, um sensor pode ser colocado dentro ou perto do fenômeno de interesse, aumentando a acurácia da observação. Este capítulo apresenta os conceitos básicos relacionados a RSSFs.

A seção 2.1 apresenta uma visão geral de redes móveis sem fio. A seção 2.2 traz conceitos iniciais de uma rede de sensores sem fio. A seção 2.3 entra em detalhes da arquitetura de hardware e software dos nós sensores. A seção 2.4 descreve as características peculiares e soluções em protocolos para atendê-las, em especial nas camadas físicas, de enlace e de rede. Finalmente, a seção 2.5 apresenta a idéia do processamento distribuído e agregação de dados na rede, que será o objeto deste trabalho.

2.1 Redes Móveis Sem Fio

Uma rede é considerada móvel quando um subconjunto de seus nós muda de lugar durante a sua operação normal, isto é, enquanto transmite ou recebe dados. Para garantir flexibilidade ao fazê-lo, o acesso à rede é normalmente sem fio. Vantagens incluem o fator estético (sem cabos), redução dos custos de infraestrutura e tempo de instalação, bem como facilidade de planejamento. Desvantagens incluem uma baixa qualidade de serviço (QoS) e o alto custo dos dispositivos, ainda que seja esperada uma queda nos preços devido aos recentes avanços na miniaturização de componentes.

Outro fator negativo é a restrição do uso de freqüências, devido a regulação governamental. Além das desvantagens já citadas, o suprimento de energia, a interoperabilidade e a segurança também são fatores de preocupação em redes sem fio. Dessa forma, os principais requisitos de projeto de redes móveis sem fio são a operação global e sem licença, baixa potência de transmissão, robustez, interoperabilidade com redes existentes, segurança e transparência para aplicações.

Existem dois tipos de redes móveis: estruturadas ou não-estruturadas, neste caso chamadas de *ad hoc*. Em muitos casos, uma parte da rede é fixa, garantindo a infraestrutura necessária para acesso das estações móveis e para interconexão com redes externas. Este é o caso da telefonia celular e redes locais sem fio interligadas com redes padrão Ethernet. Nestes casos, o equipamento que interliga a parte fixa com a móvel da rede é chamado de Ponto de Acesso (LAN sem fio) ou Estação Base (Sistema Celular). Todas as comunicações da rede são feitas entre um nó móvel e o ponto de acesso, simplificando o projeto. Entretanto, existe perda de flexibilidade quando os pontos de acesso são danificados ou tornam-se indisponíveis.

Além disso, uma vez que vários nós móveis se comunicarão com o mesmo ponto de acesso, mecanismos de controle devem ser implementados, com ou sem colisões. Da mesma forma, quando um nó móvel se afasta do seu ponto de acesso e a comunicação se torna difícil, ele precisa achar outro ponto de acesso próximo da sua nova localização, transferir a comunicação para este novo ponto e continuar a comunicação sem interrupção ou perda de dados. Este processo é chamado *handoff* [13].

Também existem redes móveis onde nós fixos não são desejados ou mesmo possíveis. Tais redes são muito utilizadas em aplicações militares e de defesa civil. Esta são chamadas redes móveis *ad hoc*, ou MANETs (*Mobile ad hoc networks*)[14]. Neste caso, todos os nós são móveis, não há infraestrutura e a topologia da rede é dinâmica por natureza. A transmissão de dados é feita diretamente de nó para nó, e para alcançar um destino além de seu alcance de transmissão, um nó precisa usar um terceiro (ou mais) nó para encaminhar as mensagens.

Isto caracteriza uma comunicação colaborativa, uma vez que inexistem roteadores específicos, mas cada nó intermediário age como um, encaminhando mensagens. Como estes nós intermediários também se movem durante a transmissão, surgem problemas relacionados com falhas na rota, e esta tarefa colaborativa de roteamento pode drenar rapidamente as escassas reservas de energia comuns a nós móveis sem fio. Os protocolos desenvolvidos para estas redes necessitam resolver estas questões e ainda manter parâmetros ótimos em termos de latência e perda de dados.

Os principais desafios das redes móveis são a Internet móvel e o acesso ubíquo, que proporcionariam ao ser humano estar sempre *on-line*, sem os entraves de cabos e equipamentos pesados. Entretanto, as soluções atuais para redes móveis não podem responder a todas as peculiaridades de um tipo diferente de computação móvel: redes de sensores.

Uma rede de sensores sem fio é uma forma muito especializada de rede *ad hoc*. Consiste de um grupo de sensores ou nós interconectados por enlaces sem fio para desenvolver tarefas específicas, como vigilância, monitoração ambiental e médica. RSSFs diferem de MANETs porque são densamente povoadas de nós, seus nós são suscetíveis a falhas e porque usualmente utilizam o paradigma de comunicação por difusão, enquanto MANETS comunicam-se ponto-a-ponto. Uma RSSF também é altamente especializada, pois quando esta é disposta no terreno, o objetivo é coletar informação sobre o fenômeno de interesse e provê-la ao usuário, enquanto o objetivo das MANETs é prover comunicações fim-a-fim entre usuários.

Desta forma, MANETs são centradas em nós, uma vez que cada nó leva informação ao usuário, enquanto RSSF são centradas em dados, com implicações nos protocolos de roteamento. Uma vez que a identificação de um nó específico não é relevante, mas os dados que ele pode levar ao usuário e o número de nós sensores pode ser muitas ordens de magnitude maior que de uma rede *ad hoc*, os nós de uma RSSF podem não ter um Identificador Global Único (ID), o que traz implicações para o esquema de endereçamento da rede.

2.2 Redes de Sensores

A idéia de construir redes de sensores surgiu de avanços recentes em comunicações sem fio e eletrônica. É previsto que no futuro próximo dispositivos de sensoriamento multifuncionais de baixo custo e potência estarão largamente disponíveis, possuindo componentes de sensoriamento, processamento e comunicações de dados em um circuito integrado, a um custo menor que 1 (um) dólar. De acordo com [15], estes dispositivos (nós sensores) devem ser menores do que $1cm^3$, pesar menos do que 100g e dissipar baixa potência (da ordem de 500mw).

Uma RSSF é composta de centenas a milhares de nós sensores, densamente dispostos em uma área geográfica de interesse. Cada nó tem a capacidade de sensoriar o fenômeno de interesse, seja ele sísmico, eletromagnético, ou acústico, transformar os dados brutos adquiridos em uma descrição de alto nível desse fenômeno e gerenciar os protocolos de comunicação para transmissão. Devido ao seu pequeno tamanho, os nós individualmente possuem capacidade limitada, mas sua eficiência aumenta quando milhares agem colaborativamente na mesma tarefa. Devido à sua natureza *ad hoc*, os protocolos das RSSFs devem ser capazes de se auto-organizar para descobrir quais nós têm informação, quais precisam dessa informação e qual a melhor forma de transferi-la de uns para os outros com o mínimo de consumo de energia.

Quando comparada com seus concorrentes tradicionais no conjunto de sensores, como radares terrestres, aeroembarcados e satélites, as vantagens de uma RSSF incluem:

- Maior tolerância a falhas, por meio de um alto nível de redundância (milhares de pequenos nós ao invés de uma grande instalação);
- Melhoria no desempenho e na razão sinal/ruído (SNR), pelo uso de múltiplos sensores com diferentes coordenadas espaciais e o emprego de técnicas de fusão de dados;
- Emprego em regiões onde o suprimento contínuo de energia não está disponível;
- Melhoria da acurácia pela disposição dos sensores dentro ou muito próximos do objeto de interesse.

De acordo com o trabalho apresentado em [16], os nós sensores dispersos em uma área de interesse constituem um campo de sensores, tal qual ilustrado na figura 2.1.

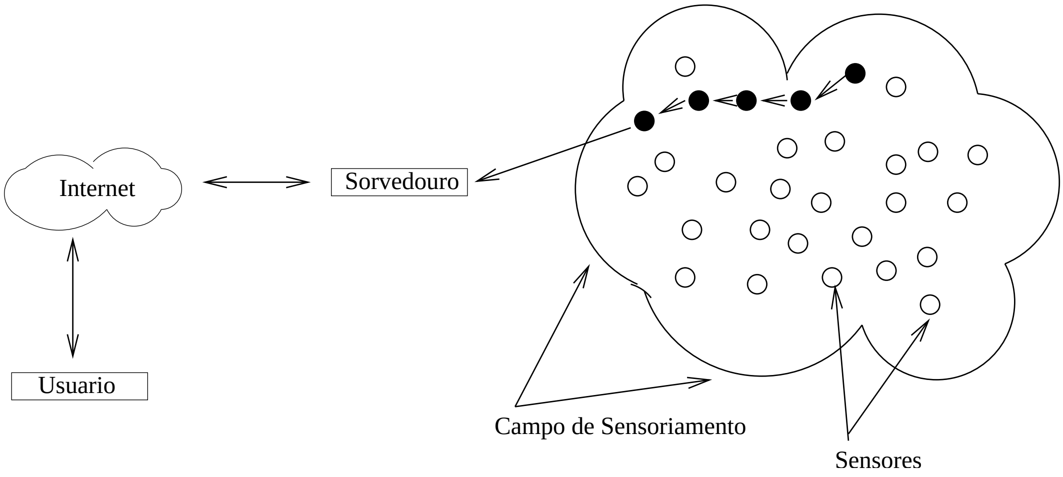


Figura 2.1: Rede de Sensores

Ainda que tenham potenciais diferentes de sensoriamento (uns podem detectar eventos sísmicos, outros acústicos), os nós possuem a mesma capacidade de enviar dados ao *sorvedouro*. Este, por sua vez, é um nó especial que requer os dados, seja para seu próprio uso (no caso de hospedar uma aplicação consumidora dos dados), seja para repassar a outra rede ou unidade de processamento fora do campo de sensoriamento. *Fontes* são os nós que produzem as informações requeridas e as enviam até que alcancem o sorvedouro.

Desta forma, os dados fluem das fontes ao sorvedouro, assim como as tarefas de sensoriamento fluem do sorvedouro às fontes. Dependendo do estado da rede, o mesmo nó pode atuar como sorvedouro ou como fonte, fazendo com que a topologia da rede seja flexível e específica para cada tarefa para a qual é exigida.

Como pode ser observado na figura 2.1, as mensagens devem seguir um caminho de múltiplos saltos até o sorvedouro, uma vez que este normalmente está fora do alcance direto de transmissão das fontes (baixa potência é uma característica básica de um sensor). Para chegar até o sorvedouro, uma mensagem deve ser transmitida da fonte para um vizinho, isto é, um nó dentro do seu alcance de transmissão. A seguir, este nó é encarregado de encaminhá-la para mais próximo do sorvedouro, até que eventualmente este último seja alcançado. Assim, qualquer nó deve possuir a capacidade de rotear dados de outros nós ao mesmo tempo em que desempenha suas próprias tarefas, usando os escassos recursos de energia de sua bateria.

Quando a energia de um determinado nó é completamente consumida, ele não interage mais com a rede, uma vez que a recarga não é economicamente viável. Quando o número de nós inteiramente *depletados* aumenta, o roteamento se torna mais difícil e drena mais energia dos nós remanescentes, que também gastarão suas baterias mais rapidamente. Finalmente, qualquer dado ainda adquirido por algum sensor não será capaz de alcançar o sorvedouro e a rede em si é considerada morta. O intervalo de tempo entre o início das operações da rede e ela ser considerada morta é chamado de *tempo de vida* da mesma. Para aumentar este tempo, os protocolos de rede precisam ser cuidadosamente projetados para conservar energia.

Energia é o fator de maior importância em RSSFs e deve ser economizada em cada componente de hardware e software que compõe a arquitetura da rede, que ainda deve se ocupar da manutenção da topologia no campo de sensores, uma vez que nós morrem e outros são adicionados. A seção 2.3 discutirá algumas propostas para a arquitetura de RSSFs.

2.3 Arquitetura

O primeiro aspecto a ser considerado na arquitetura da rede é a arquitetura de hardware de um nó sensor. Este é constituído basicamente de quatro componentes, apresentados a seguir e visualizados na figura 2.2:

Unidade de Sensoriamento: composta de dispositivos analógicos de sensoriamento e de um conversor analógico-digital. A tecnologia de sensoriamento não tem acompanhado o mesmo ritmo de desenvolvimento dos semicondutores usados nas unidades de processamento e nas memórias [17]. Os dispositivos sensores podem ser sísmicos, acústicos, infravermelhos ou magnéticos. A sensibilidade à vibração atingida pelos sensores sísmicos é igual ao ruído local de fundo dos ambientes mais quietos. A sensibilidade dos sensores acústicos é menor do que 35 dB (A) SPL (*sound pressure level*), também comparável ao ruído de fundo. A sensibilidade do infravermelho é limitada pelo ruído térmico fundamental até um valor de $5 \times 10^{-10} W/(Hz)^{1/2}$[18].

Unidade de Processamento: composta de processador e memória. Uma vez que as aplicações, tais como monitoração médica e segurança, requerem detecção de eventos, esta unidade recebe os sinais digitais (formas de onda) da unidade de sensoriamento e os transforma em pacotes de informação contendo o evento de interesse (modificação do estado de um paciente, por exemplo). Este pequeno pacote de dados é então transmitido para a rede. Esta tarefa pode ser feita colaborativamente com outros nós e parâmetros como limiar de detecção e largura de banda podem ser remotamente programados. Ainda com o objetivo de minimizar comunicações entre nós para conservar energia, aumentam os requisitos de armazenamento local, que pode ter sua capacidade aumentada consideravelmente com avanços recentes em nanoeletrônica, principalmente em MRAM (*Magnetoresistive Random Access Memory*).

Unidade de Comunicações: responsável por conectar o nó à rede. Esta conexão pode ser de rádio, infravermelho, ou óptica. Na maior parte dos sistemas sem fio, o transceptor domina a equação de dispêndio de energia, quando comparado aos outros componentes. É por isso que se busca aumentar o processamento para diminuir a necessidade de comunicações. Para distâncias maiores do que 10 metros, o transmissor torna-se o principal responsável pelo consumo de energia. Uma vez que a latência é raramente crítica, transmissão empregando múltiplos saltos a curtas distâncias é normalmente adotada. Como o receptor também consome uma quantidade considerável de energia, a monitoração do canal também custa caro. As freqüências de operação para as RSSFs são as não licenciadas, nas faixas 902-928 MHz e 2.4 GHz, que proporcionam uma boa relação entre o custo de transmissão e as dimensões das antenas necessárias.

Unidade de Potência: responsável por uma das mais difíceis tarefas de um sensor: fornecer-lhe energia, que é essencial para as outras três unidades. Em alguns casos, os nós são equipados com unidades geradoras de energia, tais como células solares, para aumentar o seu tempo de vida. Como comparação, usando 100 mW, $1cm^3$ de bateria não recarregável de lítio dura menos de 6 meses. Esta bateria pode ser substituída por $10cm^2$ de um painel solar, que em condições de luminosidade típicas atende a estas especificações, assim como $3cm^2$ de material piezo-elétrico. Atingir esta dissipação ultrabaixa de potência requer reduções da arquitetura do sistema até mesmo no nível da tecnologia de circuitos [19].

Para atingir estes fortes requisitos em termos de tamanho, custo, dissipação de potência e capacidade de processamento, o nó sensor deve ser projetado para que a maior parte de suas funções seja implementada em CMOS (*complementary metal oxide semiconductor*). Isto vale para praticamente todos os seus componentes com exceção dos passivos de RF, antenas e o dispositivo analógico de sensoriamento, que necessitam de hardware específico.

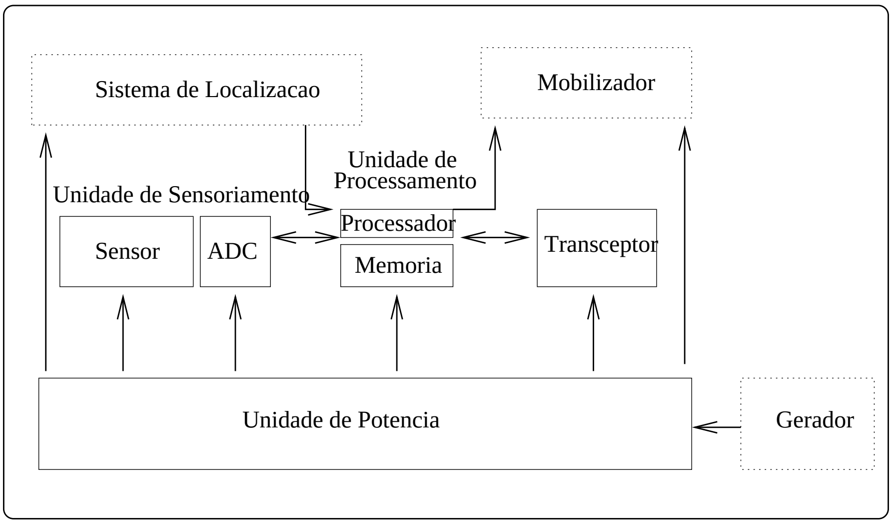


Figura 2.2: Arquitetura de um Sensor

Existem ainda alguns componentes de hardware opcionais, tais como mobilizadores ou atuadores (nos casos de sensores móveis ou ativos) e subsistemas de localização (como GPS). Os mobilizadores podem ser requeridos para mover os nós no campo de sensoriamento com o objetivo de desempenhar alguma tarefa específica. Isto pode se tornar real com a integração da tecnologia das RSSFs com a robótica para a construção de sensores capazes de se locomover autonomamente. Os atuadores são usados quando a RSSF está integrada a um sistema de controle que precisa interferir no fenômeno monitorado. O subsistema de localização é utilizado para prover o nó de informações geográficas, que são de grande importância para a maioria das aplicações.

Em termos de software, a ênfase é no sistema operacional, que precisa gerenciar e operar o nó sensor da forma mais eficiente possível. Os principais requisitos são confiabilidade, concorrência, modularidade e baixo consumo de potência. Um exemplo de um sistema operacional desenvolvido especialmente para rodar em sensores, e utilizado em grande parte do hardware hoje existente, é o TinyOS [20].

Ele é um sistema operacional baseado em eventos e possui um conjunto de serviços divididos em vários componentes. Inclui também uma linguagem de programação, chamada NesC [21] (linguagem C estilizada), para que programas sejam construídos a partir dos módulos (componentes) existentes. O comportamento destes componentes é especificado por meio de interfaces. Os componentes e suas interfaces implementam os principais serviços, tais como: comunicações (a pilha de protocolos); interface com a unidade de sensoriamento; e gerenciamento de energia, entre outros.

O TinyOS funciona baseado em comandos, eventos e tarefas, que são chamadas de função entre os componentes. Para facilitar a modularidade, cada componente declara os seus comandos e os eventos que sinaliza, onde os componentes de alto nível emitem comandos aos componentes de baixo nível e os componentes de baixo nível sinalizam eventos aos componentes de alto nível. A parte física de hardware representa o nível mais baixo dos componentes. O processamento ocorre dentro do contexto do estado dos componentes, indo para as camadas superiores do Sistema Operacional por meio de eventos e para as inferiores (em direção ao hardware) por meio de comandos.

As tarefas executam os trabalhos da rede, sendo atômicas com respeito a outras tarefas e executadas completamente. Elas podem chamar comandos na direção do hardware, sinalizar eventos para camadas superiores e programar outras tarefas dentro de um componente. O TinyOS implementa um escalonador de tarefas, que é uma fila (*FIFO - First In First Out*), utilizando uma estrutura de dados de tamanho limitado. A execução completa das tarefas torna possível alocar uma única pilha, que é atribuída a tarefa processada no momento. O processador entra no modo dormente (*sleep*) toda vez que a fila de tarefas estiver vazia, mas os periféricos continuam em operação, de modo que alguns deles podem reativar o sistema. Este comportamento permite o uso eficiente da energia.

Existem atualmente diversos projetos de pesquisa tratando de arquitetura de redes de sensores, cobrindo aspectos de hardware, software, e middleware. A seguir seguem alguns exemplos:

mAMPS [22]: um microsensor integra sensoriamento, processamento e comunicações com componentes comerciais de baixa potência em placas de 55 x 55 mm. Seu rádio (usando tecnologia Bluetooth) tem dois modos de operação: *idle*, onde o consumo de potência é o menor possível, e ativo, onde este consumo escala de acordo com a diversidade operacional do nó. Estes modos de operação são possíveis devido às características voltadas a eventos do nó, o que significa longos períodos inativos (sem eventos) e baixas taxas de transmissão.

WINS [23]: começou com a pesquisa e desenvolvimento de LWIM (*Low power Integrated Microsensors*) e MEMS (*Micro Electro Mechanical Systems*). Além do sensor, a arquitetura de um nó WINS inclui um conversor de dados, processamento de sinais e funções de controle. A lógica digital que controla o sensor e a função de processamento de sinais é baseada em tecnologia CMOS. Rádios de micro-potência possibilitam comunicações bidirecionais para acesso à rede com baixas taxas e pequenas distâncias (múltiplos saltos). Os componentes operam continuamente para reconhecimento de eventos, mas a interface de rede atua com baixo ciclo de serviço.

PicoRadio [24]: busca o desenvolvimento de uma RSSF com nós de ultra-baixa potência, chamados *piconodes*. A rede não é homogênea, pois possui três diferentes tipos de nós: sensores (para adquirir dados), monitores (para processar dados) e atuadores (para executar ações de controle). O rádio usa dois canais na faixa de 1.9 GHz, com uma taxa de 10 kbps por canal (um para dados, outro para gerenciamento). A rede usa um receptor cíclico, colocando o nó no modo de dormência (*sleep*) periodicamente. Existe também um subsistema de localização para descobrir a posição geográfica do nó e um algoritmo para ajustar a potência de transmissão dos nós para informar uma boa topologia.

SmartDust [25]: consiste no desenvolvimento de plataformas de sensoriamento e comunicações em escala milimétrica, chamadas de *dust motes*, que são os nós elementares de uma rede de milhares. Dependendo do tamanho da fonte de energia, o nó varia de $1mm^3$ a $100mm^3$. Cada nó consiste de um ou mais dispositivos de sensoriamento (temperatura, vibração, aceleração, ou pressão do ar), um micro-controlador para processar estas leituras e armazenar dados na memória e um transceptor óptico. O receptor está a maior parte do tempo em modo dormente (*sleep*) e periodicamente é ligado para verificar se outro nó está tentando estabelecer comunicações. Existem implementações funcionais dos *motes* utilizando o sistema operacional TinyOS.

As arquiteturas acima mencionadas, bem como qualquer projeto de RSSF, devem lidar com um conceito integrado de hardware, software, e dos protocolos de comunicação da rede, bem como estabelecer mecanismos para a auto-organização da rede, em termos de topologia, comunicações, fusão de dados de múltiplos sensores, filtragem, estabelecimento de perfis e até mesmo gerenciamento. Hardware e software estão fora do escopo deste trabalho. Os protocolos de comunicação serão discutidos na seção 2.4.

2.4 Pilha de Protocolos

As características peculiares das RSSFs levam à pesquisa e ao desenvolvimento de soluções para as três camadas básicas da arquitetura da rede: física, enlace com controle de acesso ao meio (MAC) e camada de rede. Devido às grandes restrições em termos de energia, idealmente o projeto dessas três camadas básicas deve ser integrado, com as características de cada uma explorada pelas demais, bem como pelo software de aplicação.

2.4.1 Camada Física

É responsável pelas comunicações dos transmissores e receptores no meio sem fio, modulando o sinal e codificando os dados. Em [7], é apresentado o modelo de rádio de primeira ordem para uma rede de sensores, onde se dissipa $E_{elec} = 50nJ/bit$ nos componentes eletrônicos do transceptor e $E_{amp} = 100pJ/bit/m^2$ no amplificador de transmissão, para uma relação sinal/ruído aceitável. Dessa forma, a equação de consumo de energia para a transmissão de k bits a uma distância d é:

$$E_{Tx}(k,d) = E_{elec} * k + E_{amp} * k * d^2 \tag{2.1}$$

A energia gasta no receptor é dada por:

$$E_{Rx}(k) = E_{elec} * k \tag{2.2}$$

Com base nas equações 2.1 e 2.2, pode ser inferido que receber uma mensagem também consome uma considerável quantidade de energia, de forma que os protocolos de comunicação devem minimizar o número de operações de transmissão e recepção na rede. Esta seria uma tarefa da camada de enlace de dados, que também resolve questões como controle de acesso, tendo em vista que o meio sem fio é compartilhado.

2.4.2 Camada de Enlace

É responsável pela multiplexação dos fluxos de dados produzidos pela camada física, criando (e detectando) quadros (*frames*) para facilitar o controle de erros e o acesso ao meio (MAC - *Medium Access Control*). O objetivo final de todas estas tarefas é garantir comunicações confiáveis, seja ponto-a-ponto, *broadcast* ou *multicast*. Devido às características de auto-organização de uma RSSF, a camada de enlace deve ser distribuída e não deve requerer sincronização global. Devido às restrições de potência, ela também deve ser reativa, baseando-se em eventos. Dados devem ser transmitidos apenas quando há um evento de interesse. Do contrário, a rede deve hibernar para economizar energia. Fatores que influenciam a qualidade de serviço, como retardo e vazão, também são importantes no projeto, mas em RSSFs o requisito mais importante é o consumo de energia.

O trabalho apresentado em [26] fornece uma descrição funcional da camada de enlace para RSSFs, dividindo-a em 10 subsistemas:

- Controle de Acesso ao Meio: determina quando (em qual *time slot*) cada nó pode transmitir e em que canal;
- Controle de Erros: codifica e decodifica os dados usando uma técnica específica de correção ou detecção de erros;
- Transmissão: responsável pelo envio de dados à camada física;
- Endereço local: designa um endereço local único para garantir escalabilidade. Este endereço é único no conjunto formado pelos vizinhos do nó e pelos nós

adjacentes aos vizinhos;

- Localização: calcula ou refina a localização geográfica do nó;
- Processamento de dados: processa os quadros vindos da camada física;
- Lista de vizinhos: cria e mantém uma lista com o endereço local, e uma medida da qualidade do enlace para cada vizinho;
- Mobilidade: suporte para nós móveis;
- Qualidade do enlace: calcula uma métrica (normalmente energia gasta) para cada enlace. Pode ser usada pela camada de rede para configurar a probabilidade de escolher um caminho. Também armazena o estado do enlace para ser usado pelo subsistema de Controle de Acesso ao Meio e a potência do sinal recebido para o subsistema de localização;
- Controle de Potência: mantém registro do nível de energia remanescente.

De todas as tarefas da camada de enlace, a de Controle de Acesso ao Meio é uma das mais críticas. De acordo com [27], técnicas tradicionais de multiplexação possuem características problemáticas para RSSFs. O TDMA (*Time Division Multiple Access*) requer controle preciso da freqüência do oscilador do rádio, enquanto o CDMA (*Code Division Multiple Access*) requer a operação em alta velocidade dos circuitos digitais por um período relativamente longo de tempo, potencialmente consumindo energia em excesso. Para fazer frente a estes problemas no projeto de camadas de Controle de Acesso ao Meio, uma combinação dessas técnicas de multiplexação deveria ser utilizada, ou características deveriam ser adicionadas para aprimorar estas técnicas para redes de sensores.

O trabalho apresentado em [28] busca minimizar o gasto de energia baseando-se na topologia da rede e no emprego de uma camada MAC com origem no TDMA. Nesta abordagem, o tráfego de dados entre sensores é arbitrado no tempo para evitar colisões (TDMA) e permite o desligamento de sensores não necessários. A topologia é importante porque a rede precisa ser organizada em hierarquia de *clusters*. Assim, os *cluster heads* ou *gateways* são os responsáveis pelo gerenciamento dos protocolos

MAC e de roteamento. Eles coordenam a designação de quando (quais os *slots*) os nós escutariam pela transmissão de outros nós e quais *slots* poderiam usar para sua própria transmissão. Ele também é responsável por controlar o nível de energia restante de todos os sensores do seu *cluster*.

Em outro trabalho [29], é usada uma combinação de CDMA com desligamento periódico do receptor. A camada de enlace também inclui capacidade de localização, designação de endereços locais, controle de potência e gerenciamento de lista de vizinhos.

Um outro protocolo é o SMAC (Sensor MAC), descrito em [30]. Trata-se de uma camada MAC baseada em contenção, nos termos de que o tráfego é coordenado entre os sensores para evitar colisões. O problema nos protocolos baseados em contenção é a necessidade de que os nós continuamente escutem canais vazios, o que provoca desperdício de energia, como pode ser visto na equação 2.2. O SMAC aprimora isto configurando os nós para que periodicamente escutem o canal e depois adormeçam. Nós sensores em proximidade geográfica formam *clusters* virtuais para se auto-sincronizar no modo adormecido. Cada nó adormece por algum tempo, acordando para escutar o canal e verificar se algum *vizinho* deseja se comunicar. Enquanto adormece, o nó desliga seu rádio e configura um temporizador para ser posteriormente acordado.

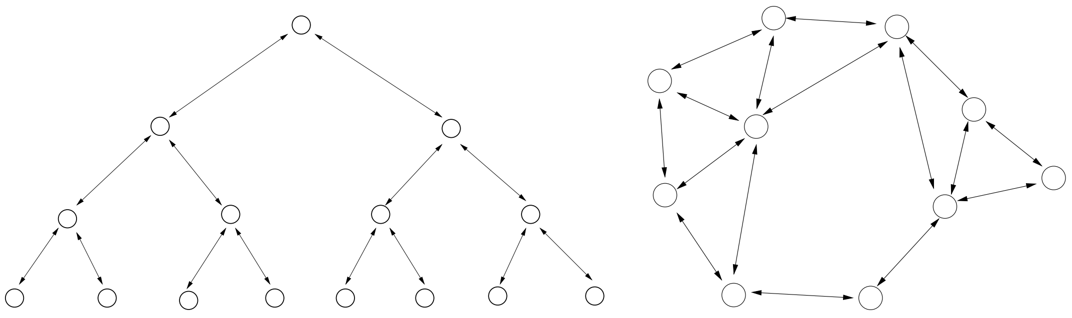
As questões referentes ao projeto de protocolos da camada de enlace para RSSFs são na sua maior parte relacionadas à economia de energia e controle de acesso ao meio. A implementação de tais protocolos pode ser independente do projeto da camada de rede, mas um projeto conjunto aprimora o desempenho da RSSF, nos termos em que a camada de rede pode beneficiar-se de características como localização dos nós, métricas de enlace, listas de vizinhos e controle de potência.

2.4.3 Camada de Rede

Uma vez que as comunicações em uma RSSF devem ser a curtas distâncias, os protocolos da camada de rede devem permitir comunicações por múltiplos saltos entre fontes e sorvedouros. Estes protocolos devem ser eficientes no dispêndio de energia, tolerantes a falhas e centrados em dados, além de estabelecer colaborações locais e permitir endereçamento baseado em atributos. Devem também prover acesso a redes externas, como outra RSSF, a Internet ou um sistema de comando e controle.

A responsabilidade primária desta camada é o roteamento de dados e o endereçamento dos nós. Em RSSFs, os protocolos de roteamento de dados devem descobrir o caminho com mínimo dispêndio de energia para as comunicações com o nó de interesse. O endereçamento é crítico devido às características da topologia, escala (milhares de nós) e mobilidade. O roteamento em RSSFs, diferentemente das redes tradicionais, deve ser centrado em dados, e não em endereços. O endereçamento, por sua vez, não pode ser baseado nos nós, mas sim em atributos que sejam relevantes para a aplicação. Devido à sua relevância para este trabalho, o roteamento centrado em dados e o endereçamento baseado em atributos serão discutidos à parte, na seção 2.5.

Existem duas famílias distintas de protocolos de roteamento aplicáveis para RSSFs: hierárquicas ou planas, dependendo da topologia desejada. A figura 2.3 ilustra as diferenças.



(a) Topologia Hierárquica (b) topologia Plana

Figura 2.3: Famílias de Protocolos de Roteamento em RSSFs.

Em 2.3(a), é observada a topologia hierárquica. Os nós são colocados em níveis diferentes e grupos (*clusters*) são formados. Em cada grupo, os nós podem se comunicar apenas com um nó especial, chamado de *líder* (*cluster head* ou *gateway*), que coordena todas as comunicações no seu grupo. Os líderes podem também se agrupar formando *clusters* de ordem mais elevada, fazendo com que a hierarquia cresça verticalmente. Em cada nível, o líder recebe informações dos nós abaixo dele, a integra com alguma função de agregação e reporta verticalmente os resultados. O nó de mais alto nível é chamado de estação base, tornando-se o recipiente final dos dados, responsável pela entrega de informação ao usuário ou para redes externas. Neste tipo de topologia, surgem problemas em como selecionar os líderes e configurar a hierarquia de forma que o consumo de energia seja distribuído uniformemente entre todos os nós da rede.

Em 2.3(b), está a topologia plana. Não existem nós especiais (*cluster heads*) e cada um é autônomo e conectado a alguns ou a todos os outros nós, dependendo do alcance do seu transmissor. Também não existem estações base, de forma que todos os nós estão engajados no mesmo nível de atividades de roteamento, e qualquer um pode ser o sorvedouro dos dados e o ponto de acesso ao usuário. Esta abordagem pode acarretar em um custo elevado para as comunicações porque n^2 interconexões são necessárias para redes de n nós. Para evitar isso, os protocolos devem identificar e privilegiar melhores caminhos das fontes ao sorvedouro e limitar o número de conexões.

O mais antigo e simples protocolo para roteamento de dados em redes de sensores é o da inundação (*flooding*), que é freqüentemente utilizado como parâmetro de comparação com outros protocolos. O seu princípio operacional é simples: cada nó recebendo um pacote de dados o retransmite por difusão até que o destino final seja alcançado ou um número máximo de saltos seja contado. Aspectos negativos deste protocolo são:

Implosão: caracterizada por mensagens duplicadas serem enviadas ao mesmo nó e acontece quando os nós transmissores compartilham os mesmos vizinhos;

Sobreposição: ocorre quando dois vizinhos próximos sensoriam a mesma coisa ao mesmo tempo e também originam mensagens duplicadas;

Inaptidão para controle de recursos (energia): o protocolo não leva em consideração o dispêndio de energia no roteamento.

Na topologia hierárquica, um dos protocolos de maior destaque é o LEACH (*Low Energy Adaptative Clustering Hierarchy*), proposto em [7]. Ele usa uma rotação aleatória dos líderes para balancear o dispêndio de energia entre os nós da rede. Em abordagens hierárquicas convencionais, os líderes consomem mais energia que outros. Isto é inviável em RSSFs porque tais nós *depletariam* rapidamente seus estoques de energia, diminuindo também o tempo de vida do *cluster*. O LEACH também estabelece computação local para agregar ou comprimir dados e reduzir a quantidade de pacotes a serem transmitidos verticalmente até ao sorvedouro.

Os sensores se auto-elegem probabilisticamente como líderes e difundem este *status*. Cada nó determina em qual grupo (*cluster*) deve pertencer tendo por parâmetro a energia para se comunicar com o líder. Este é o processo de configuração da topologia da rede e é repetido periodicamente para garantir a rotação de líderes e a criação de novos grupos.

Uma vez criada a topologia, cada líder cria um cronograma de transmissão, permitindo aos demais nós desligarem seus rádios, exceto nos seus intervalos de transmissão. Ao completar o ciclo de mensagens recebidas de todos os nós do grupamento, o líder agrega as informações disponíveis e envia os dados comprimidos para a estação base. A economia de energia é devida a dois fatores: compressão com perdas nos líderes de grupos e no esquema de roteamento de dados. Uma vantagem é que esta economia é obtida com computações e interações locais, não havendo necessidade de interação global na rede, que é um requisito vital para RSSFs.

Outro exemplo de protocolo hierárquico é o PEGASIS, proposto em [8], no qual cada nó se comunica apenas com um vizinho próximo e tira turnos para transmitir à estação base. Este esquema forma uma cadeia de nós, na qual os dados colhidos movem-se de nó para nó, são agregados, até que o nó designado transmita à estação base. Todos os nós, exceto o último da cadeia, processam a fusão de dados. Os autores afirmam que esta abordagem seria um aprimoramento do LEACH, pois não é necessária a formação de grupos. No LEACH, 5% dos nós são líderes e devem transmitir por difusão para todos os nós. No PEGASIS, apenas um nó por rodada transmite para a estação base. Nas simulações realizadas, foi verificada uma melhoria por um fator de dois em relação a LEACH. Entretanto, a abordagem da cadeia pode reduzir a sensibilidade da rede ao fenômeno de interesse, uma vez que apenas um nó em cada rodada vai agregar os dados de todo o campo de sensores e enviar à estação base, provocando também aumento do retardo.

Na topologia plana, um exemplo de protocolo é o SPIN (*Sensor Protocols for Information via Negotiation*), descrito em [31], que incorpora negociação e adaptação de recursos. A negociação elimina os problemas da implosão e sobreposição, porque certifica que apenas informação relevante será transferida. Os nós usam metadados para descrever a informação coletada e assim anunciar suas capacidades a outros nós. Cada nó também gerencia seus recursos, permitindo-se eliminar certas atividades quando seu nível de energia estiver baixo.

Dessa forma, são usados três tipos de mensagens: ADV, que anuncia os metadados para a rede e é usada antes da transmissão de informação; REQ, que é o pedido de dados enviado por um nó interessado após ter recebido uma mensagem ADV; e DATA, que é a mensagem que transporta a informação propriamente dita. Os autores compararam o desempenho com a inundação simples e encontraram uma melhoria por um fator de 4, indicando também que a latência não aumentou devido à fase de negociação.

Outro exemplo de protocolo plano é o EAR (*Energy Aware Routing*), desenvolvido sob o projeto PicoRadio e descrito em [9]. É um protocolo reativo e busca encontrar caminhos de mínimo dispêndio de energia. Ao invés de identificar o melhor caminho, ele mantém um conjunto de bons caminhos, escolhendo um deles probabilisticamente a cada rodada de transmissão de dados. Isto evita que o melhor caminho tenha sua energia esgotada rapidamente, aumentando a sobrevivência da rede.

Ao manter diversos caminhos, o protocolo também melhora a tolerância para nós móveis. O EAR é reativo porque quem inicia o estabelecimento e a manutenção das rotas é o consumidor dos dados, e possui três fases: configuração ou propagação de *interesses*, quando inundação localizada ocorre para que se encontrem todas as rotas da fonte ao destino e seus custos em energia, configurando assim tabelas de interesses;

propagação de dados, usando as tabelas de roteamento já estabelecidas, selecionando probabilisticamente qual caminho usar; e manutenção das rotas, quando inundação localizada é realizada para manter os caminhos vivos.

Outro exemplo de protocolo muito similar ao EAR é o da Difusão Direcionada (*Directed Diffusion*), proposto em [5]. É centrado em dados e consegue economia no gasto de energia ao selecionar empiricamente bons caminhos, estabelecendo cache local em todos os nós e processando dados de forma distribuída na rede. Além disso, ele criou um sistema de nomes que auxilia na identificação das fontes de dados e interesses. Como estas características são exploradas na proposta deste trabalho, a difusão direcionada será discutida em mais detalhes no capítulo 3.

2.5 Processamento Distribuído com Agregação de Dados

RSSFs são específicas para cada tarefa e devem ser econômicas no consumo de energia. O trabalho descrito em [4] conclui que o custo da comunicação excede por grande margem o custo do processamento em cada nó. Os autores assumem um modelo de comunicações com características tais como: portadora de 1 GHz, elevação da antena de 1/2 comprimento de onda, modulação BPSK com Probabilidade de erro de 10^{-6}, desvanecimento de Rayleigh e um receptor ideal. Neste modelo, o custo de se transmitir 1kb na distância de 100m é de aproximadamente de 3 joules. Em contraste, um processador genérico com 100MIPS/W pode executar 3 milhões de instruções com a mesma quantidade de energia.

Sendo assim, o projeto deve minimizar as comunicações na rede, enquanto mantém em níveis aceitáveis o desempenho da tarefa de interesse. Portanto, para economizar energia é melhor aumentar o processamento dos dados em cada nó de forma a minimizar suas transmissões. De acordo com [32], o desempenho das comunicações pode ser quantificado por três parâmetros: alcance, confiabilidade, e latência.

O alcance representa a distância ao recipiente; a confiabilidade trata da probabilidade de que os dados transmitidos sejam propriamente recebidos; e a latência mede o tempo requerido para a comunicação das fontes ao sorvedouro. Como em RSSFs é desejável a comunicação em curtas distâncias, o alcance deve ser mantido curto, e a transmissão ao sorvedouro deve seguir rotas de múltiplos saltos. A confiabilidade e a latência dependem dos requisitos da aplicação, que podem ser relaxados, permitindo à unidade de comunicações do nó trocar o desempenho por economia de energia. Além disso, a confiabilidade pode ser aprimorada com o uso adequado de técnicas de fusão de dados nos nós intermediários.

Estes aspectos tornam o processamento distribuído na rede essencial para o seu tempo de vida. A redundância dos dados coletados e a associação muitos-para-um entre fontes e sorvedouros podem ser explorados por um esquema de roteamento centrado em dados para agregação em nós intermediários, minimizando o número de transmissões e, portanto, economizando energia, o que será discutido em 2.5.1. Para que esta *agregação-no-interior-da-rede* (*in-network-aggregation*) se torne efetiva, um esquema apropriado para o endereçamento e encaminhamento dos dados das fontes até o sorvedouro é essencial, e será discutido em 2.5.2. O código do software de aplicação pode ser distribuído pela rede e ativado quando necessário para agregar ou comprimir dados. Técnicas de agregação de dados serão discutidas em 2.5.3

2.5.1 Roteamento Centrado em Dados

A abordagem tradicional para roteamento em redes *ad hoc*, bem como em outros tipos de redes móveis, é centrada em endereços. Neste caso, as fontes enviam dados de forma independente para o destino. Em RSSFs, o foco do roteamento é deslocado dos nós da rede para os dados nela trafegados. Dessa forma, surge um novo paradigma, centrado em dados, que permite processamento distribuído na rede, com consolidação de dados redundantes enquanto eles trafegam, reduzindo a comunicação necessária para enviar leituras de um conjunto de sensores para um destino comum e, conseqüentemente, aumentando o tempo de vida da rede.

Um modelo para este esquema foi discutido em [6]. Seu objetivo é a construção de uma árvore de agregação com várias fontes convergindo para o sorvedouro de forma a obter uma agregação ótima, e seu desempenho comparado com esquemas de roteamento fim-a-fim. Ao ler o conteúdo das mensagens, os nós intermediários processam algum tipo de agregação, que reduz o número de transmissões, mas aumenta o retardo, uma vez que os dados de fontes próximas precisam esperar para ser combinados com os de fontes distantes, segundo o método *store-and-forward* (armazena-e-encaminha).

Os autores analisam três esquemas quase ótimos para obtenção dessa árvore em função de dois modelos de posicionamento e de um número crescente de fontes. Descobriram que os maiores ganhos em termos de economia de energia ocorrem quando as fontes estão todas próximas umas das outras e afastadas do sorvedouro. Isto minimizaria o retardo e maximizaria a redução nas transmissões.

Um exemplo de roteamento centrado em dados é o algoritmo chamado *Data Funnelling*, desenvolvido no projeto PicoRadio e descrito em [33]. Neste algoritmo, um determinado nó intermediário recebe os dados de um grupo de fontes, e então envia um fluxo comum de dados até o destino. São usados dois métodos para aumentar o tempo de vida da rede: agregação de pacotes e compressão de dados. O campo de sensores é heterogêneo, e seus nós consistem de alguns poucos controladores (ou atuadores), com maior capacidade de processamento, e muitos sensores.

O controlador divide o campo em regiões distintas e envia interesses para cada uma, usando inundação direcional. Ao atingir a *borda* da região alvo, o interesse é modificado para que os nós internos se organizem de forma sincronizada para reportar-se ao nó da borda, de forma a permitir neste ponto a agregação e a compressão de dados, antes do seu envio ao sorvedouro. Dessa forma, a agregação é realizada próxima do fenômeno de interesse, o que é interessante, mas existe um alto custo em estabelecer, sincronizar e manter a árvore de agregação dentro de cada região alvo.

Outros protocolos buscam estabelecer árvores de agregação. A árvore NP-complexa de Steiner fornece o caminho ótimo para agregação em termos de número de transmissões [34]. Existem também esquemas sub-ótimos, que foram examinados em [35]. Um esquema é do tipo *Centro na Fonte mais Próxima* (*CNS - Center at Nearest Source*), no qual a fonte mais próxima do sorvedouro age como ponto de agregação. Todas as outras fontes enviam os dados diretamente para esta, que envia a informação agregada ao sorvedouro.

Outro esquema é do tipo *Árvore dos caminhos mais curtos* (SPT - Shortest Paths Tree), onde cada fonte envia seus dados ao sorvedouro pelo caminho mais curto entre os dois. Nos pontos onde os fluxos de dados coincidem para fontes diferentes, eles são combinados para formar uma árvore de agregação. Nesta dissertação, este esquema será chamado de *Agregação Oportunista*. Existe também o esquema baseado em *Árvore Incremental* (*Greedy Incremental Tree*), no qual a árvore é construída seqüencialmente. No primeiro passo, apenas a fonte mais próxima ao sorvedouro é incluída. Em cada passo seguinte, a fonte mais próxima da árvore é conectada e assim por diante. Nestes esquemas, os maiores ganhos são obtidos quanto maior for o número de fontes, mais próximas umas das outras e afastadas do sorvedouro elas estiverem.

O roteamento baseado em agrupamentos hierárquicos (*clusters*) favorece a agregação nos *cluster heads* e é analisado em [7]. Outra questão importante é o balanço de dispêndio de energia entre os nós intermediários. Quanto maior é o número de nós agregadores na rota até o sorvedouro, maior é o retardo, mas melhor é o balanço de energia. Todas estas propostas não levam em conta o gasto em energia para estabelecer e manter árvores de agregação em topologias dinâmicas devido a falhas de nós. Na abordagem hierárquica, há ainda eleição dos *cluster heads* e sua permuta, que também envolve troca de mensagens e consome uma quantidade não desprezível de energia.

2.5.2 Endereçamento Baseado em Atributos

RSSF são exemplos de sistemas distribuídos onde as comunicações de baixo nível não são baseadas em endereços de nós, mas em atributos de mensagens que sejam relevantes para a aplicação. Com uma grande população de nós sensores, torna-se inviável o estabelecimento de identificadores únicos (como um endereço IP) para todos os nós, de forma que um sistema de nomes baseado em atributos deve ser utilizado, em benefício do software de aplicação. Os nomes podem ser baseados no tipo de sensor (temperatura, pressão, etc.) ou na sua localização geográfica (caso fornecida pela camada de enlace).

Este tipo de endereçamento elimina os custos relacionados com a resolução de nomes, além de permitir redução ou agregação de dados perto de onde são gerados, pois as aplicações precisam descrever *qual* dado precisam, e não *onde* encontrá-lo.

Por exemplo, seja a situação descrita a seguir: a aplicação precisa identificar qual região do campo de sensores tem temperatura maior do que um determinado limiar (p. ex., 40^0C). Se os nós foram identificados por um conjunto de atributos [*Tipo*, *Localização*, *Temperatura*], os nós com os atributos [*Tipo* = temperatura, *Localização* = N-E, *Temperatura* = 42] responderiam à consulta, indicando que o quadrante nordeste do campo é a resposta correta.

2.5.3 Técnicas de Agregação de Dados

O processamento distribuído implica os nós intermediários armazenarem as mensagens na chegada, esperar por mensagens de outras fontes e então combiná-las em novas mensagens comprimidas, e só então encaminhá-las ao destino. A grande variedade de aplicações e a diversidade de técnicas tornam o estudo da agregação de dados muito complexa [16]. Algumas dessas técnicas são baseadas em métodos matemáticos e heurísticos envolvendo diversas disciplinas como a estatística, a inteligência artificial, o processamento de sinais e a teoria de decisão [36].

A função mais trivial de agregação de dados é supressão de mensagens duplicadas. Outras funções básicas incluem calcular o valor máximo, mínimo ou médio das mensagens como discutido em [37]. A compressão de dados também é sugerida em [33] como uma forma de agregação. Esta pode ocorrer somente uma vez (*instantânea*) ou em intervalos regulares (*periódica*) [37]. As funções básicas de agregação são adequadas para agregar múltiplas informações de um mesmo tipo de sensor (ex.

temperatura), enquanto a fusão de dados de diferentes tipos de sensores (ex. temperatura, pressão atmosférica, umidade, etc.) exige métodos mais complexos para ocorrer.

Para se extrair o máximo das técnicas de agregação de dados distribuídas na rede e atingir a maior economia no dispêndio de energia, integrando o hardware, os protocolos de comunicação e a aplicação, é importante que exista uma API (*Application Programming Interface*) que possibilite ao software de aplicação utilizar-se das características das camadas MAC e de rede para seu processamento, bem como para especificar suas restrições em latência, confiabilidade e acurácia.

Uma API deste tipo foi desenvolvida pelos criadores do *Directed Diffusion* e foi utilizada neste trabalho, conforme será apresentado no capítulo 3. Maiores informações sobre técnicas de agregação de dados empregadas de forma distribuída em RSSF podem ser encontradas em [38].

Capítulo 3

Mecanismo de Roteamento para Agregação de Dados em RSSF

NESTE capítulo, é proposto um mecanismo para favorecer a agregação durante o roteamento de dados em RSSFs, aproveitando-se dos três diferentes aspectos, que são *onde*, *como* e *quando* agregar, implicando o uso de protocolos de roteamento e mecanismos de filtragem nos nós sensores. Estes mecanismos são implementados de forma a explorar seus impactos na rede, em particular, a economia de energia. Neste trabalho, propõe-se uma extensão a um protocolo de roteamento centrado em dados, de forma a aumentar a possibilidade de agregação na rede sem incorrer em altos custos relacionados com o estabelecimento e manutenção de uma árvore de agregação.

O protocolo de roteamento centrado em dados utilizado é o *Directed Diffusion* [5], ou Difusão Direcionada, que estabelece comunicações eficientes por vários caminhos entre uma ou mais fontes e sorvedouros, e que permite processamento enquanto dados fluem na rede. O sorvedouro dissemina *interesses*, que constituem as tarefas de sensoriamento. Estes interesses são armazenados e interpretados pelos nós intermediários, que criam *gradientes* provocando o estabelecimento de rotas para os dados cujo conteúdo case com o do interesse. Dessa forma, dados fluem das fontes até os originadores de interesse por múltiplos caminhos. O sorvedouro reforça um vizinho em particular, aquele do qual ele recebeu a primeira mensagem de dados.

Os nós intermediários usam o mesmo critério para reforçar seus respectivos vizinhos, criando uma rota até a fonte, o que privilegia caminhos de pequeno retardo, mas não necessariamente leva ao estabelecimento dos melhores caminhos para agregação.

Estes caminhos, descobertos por interações locais entre nós vizinhos, são independentes para fluxos de fontes diferentes. Proponho um novo esquema para descoberta de caminhos, reforçando nós intermediários com maior potencial para combinar dados de diversas fontes. Após a descoberta destes nós agregadores, devem ser empregados algoritmos distribuídos de estimação para criar fluxos comuns, agregando de forma eficiente o tráfego da rede em um número reduzido de mensagens de dados.

A idéia é usar as mesmas interações locais características da difusão direcionada para estabelecer uma árvore de agregação empírica na rede. Para tal, foi utilizada a sua arquitetura baseada em filtros, descrita em [39], que permite que um módulo de software externo, chamado *filtro*, interaja com o núcleo do protocolo e modifique suas capacidades de roteamento, influenciando assim como os dados são transferidos pela rede. Dois tipos de filtros são usados, um para descoberta de rotas com nós agregadores (Filtro PAR) e outro para agregação de dados, que foge ao escopo deste trabalho, mas foi implementado no mesmo grupo de pesquisa. Uma descrição detalhada do filtro de agregação de dados pode ser encontrada em [38]. Outras informações sobre o trabalho desta dissertação podem ser obtidas em [40, 41].

3.1 Arquitetura Distribuída Baseada em Filtros

A arquitetura de comunicações utilizada é baseada em três componentes: o núcleo do protocolo da difusão direcionada, *regras de casamento* (ou *matching rules*) e filtros. O primeiro é usado para disseminar a informação no sistema distribuído. As mensagens trafegadas são identificadas por meio de trincas *atributo-valor-operação*. As regras de casamento atuam sobre estas trincas e identificam quando o dado chegou no seu destino (sorvedouro) ou se deve ser processado em filtros (no caso deste trabalho o filtro PAR) distribuídos nos nós intermediários. Os filtros são os módulos de software incorporados ao protocolo que estabelecerão o processamento distribuído na rede. Uma descrição mais detalhada desses componentes pode ser obtida em [42].

3.1.1 Difusão Direcionada

O núcleo da arquitetura é a difusão direcionada. Como já mencionado, ela se baseia na disseminação de interesses pelos sorvedouros e no armazenamento e interpretação destes em nós intermediários. Estes interesses constituem as tarefas da rede e são descritos pela trinca *atributo-valor-operação* (p. ex. $temperatura \geq 40^0C$). A interpretação ocorre por meio de *gradientes*. Um gradiente representa tanto a direção para onde os dados devem fluir (para qual vizinho), caso casem com um determinado interesse, como o status dessa demanda (se o fluxo está ativo ou não e, possivelmente, qual a taxa de transmissão).

Quando um nó sensor que *case* com o interesse é encontrado, a aplicação local ativa a sua unidade de sensoriamento para o início da coleta de dados, até então desligada para economia de energia. Este nó é então chamado de fonte e começa a enviar mensagens de dados, também na forma *atributo-valor-operação* (p. ex. $temperatura = 45^0C$). Estes dados são propagados na direção do sorvedouro por meio dos gradientes previamente configurados e armazenados nos nós intermediários em um cache local. Este servirá como base de dados para atuação do filtro PAR no estabelecimento das rotas e para funções de agregação utilizadas para a combinação das informações sobre a detecção de um único objeto por vários sensores em uma resposta única ao sorvedouro, com supressão das demais.

A primeira mensagem de dados enviada por uma fonte para a rede é marcada pelo *Diffusion* como *exploratória* e é enviada para todos os vizinhos com gradientes. Ela é usada para a configuração dos melhores caminhos, sendo propagada até atingir o sorvedouro. Se este tiver múltiplos vizinhos, cada um enviando uma mensagem exploratória, ele escolhe o vizinho de que recebeu primeiro para enviar um reforço positivo, que por sua vez também reforça o primeiro de seus vizinhos (na direção das fontes) de quem recebeu a mensagem exploratória, até que eventualmente a fonte receba o reforço positivo e passe a transmitir mensagens de dados.

Dessa forma, é formado um caminho preferido, com gradientes reforçados, por onde as mensagens de Dados serão propagadas até o sorvedouro. Caso este caminho falhe, os nós intermediários podem localmente repará-lo, pois mantém, na sua tabela de roteamento, gradientes com alternativas. Estes nós enviam então, reforços negativos para o vizinho do caminho que falhou e reforço positivo para um novo vizinho, estabelecendo dessa forma um novo caminho.

Estes caminhos preferidos são seguidos pelos fluxos de dados fontes/sorvedouro até que um novo fluxo de mensagens exploratórias inunde a rede para propiciar a descoberta de novas rotas. Durante um certo tempo, as mensagens de dados são propagadas por ambas as rotas até que uma delas (a que apresente maior retardo) é abandonada.

3.1.2 Regras de Casamento

As regras de casamento controlam quais filtros serão ativados e quando, e como fontes e sorvedouros se relacionam. Como já mencionado, as mensagens (dados ou interesses) são compostas de trincas atributo-valor-operação. Os atributos são formados por chaves únicas que identificam o seu tipo, podendo suportar dados de diversos formatos (integer, float, string, blob). O valor é a sua quantidade. As operações definem como as mensagens de dados e interesses interagem. São as operações binárias de comparação usuais (EQ=igualdade, NE= diferença, LE= menor igual, etc). Existe o operador IS, que permite definir um valor literal.

Por exemplo, temperatura IS 45, significa que a unidade de sensoriamento do nó em questão leu o valor 45. Dessa forma, uma mensagem de Interesse usaria o operador GE (maior igual) associado ao valor 40. Uma mensagem de dados usaria o operador IS com valor 45 para responder a este interesse. As regras da Difusão Direcionada casam sempre um operador IS com os demais operadores.

Dessa forma, através da escolha adequada de nomes para os atributos das mensagens de dados e pela utilização de operações de comparação simples, é possível estabelecer regras de casamento que sejam de pequeno custo em termos de processamento, e rápidas na execução. A arquitetura foi estabelecida de forma que novas regras podem ser criadas, que foram utilizadas no presente trabalho para seleção dos nós vizinhos com melhores parâmetros de agregação.

3.1.3 Filtros

Filtros são módulos de software que processam as mensagens do *diffusion* enquanto estas se movem pela rede, constituindo o mecanismo da arquitetura para rodar na rede código específico da aplicação, assistindo de forma conveniente o roteamento e processamento. Os módulos podem ser instalados nos sensores antes da sua disposição no campo, ou podem ser distribuídos como pacotes móveis durante o tempo de execução. Para funcionar, o filtro deve ser registrado junto ao núcleo do protocolo, bem como deve estabelecer sobre que tipo de mensagem atuará através de regras de casamento explícitas.

Qualquer mensagem chegando em um nó onde o filtro está instalado o ativará sempre que seus atributos casem com os atributos do filtro. Após ser invocado pela chegada da mensagem, ele a processa, podendo modificar seus atributos, duplicá-la, suprimi-la, ou qualquer outra ação que seja necessária para a aplicação. Caso mais de um filtro seja usado, é necessário especificar prioridades (que não podem ser iguais), indicando em qual filtro a mensagem deve ser processada antes. O filtro então se encarrega de definir para onde a mensagem irá depois.

Os filtros também têm acesso a informações internas do núcleo, como os gradientes e lista de vizinhos, que foram utilizadas no presente trabalho. É interessante observar que o roteamento básico da arquitetura foi implementado em um filtro, o *Gradiente*. Ele mantém os gradientes representando o estado de qualquer fluxo existente para todos os vizinhos e é responsável pelo envio periódico de mensagens de reforço (tanto positivo, para privilegiar rotas, como negativo, para remover rotas interrompidas) e interesses. Como já mencionado, os fluxos de dados escolhidos pelo filtro Gradiente são independentes, privilegiando rotas de menor retardo. O filtro elaborado no presente trabalho busca modificar este mecanismo sem interferir no processamento do Gradiente.

3.2 Roteamento Privilegiando a Agregação (Filtro PAR)

O filtro PAR foi desenvolvido para modificar o roteamento natural da difusão direcionada, realizado pelo filtro *Gradiente*, escolhendo preferentemente caminhos que levem a nós intermediários capazes de agregar dados, que serão chamados de *agregadores*.

Um nó tem potencial para agregação se recebe, dentro de um pequeno intervalo de tempo e de **fontes diferentes**, dados com os mesmos atributos (*i.e. temperatura*), mas não necessariamente os mesmos valores. A fusão dos dados vai depender da função de agregação utilizada e esta depende da aplicação da rede de sensores. Um determinado caminho é mais eficiente se o número de nós agregadores é maior e se estes estão próximos das fontes.

Para identificar e selecionar estes melhores caminhos, o filtro PAR (*Privileged Aggregation Routing* - Roteamento com Privilégio para Agregação) foi adicionado à implementação da difusão direcionada disponível no simulador ns-2 [43], como mostrado na Figura 3.1.

O filtro foi implementado com uma prioridade maior que a do filtro Gradiente, e suas regras de casamento permitem somente o processamento das mensagens *exploratórias*. Ou seja, tanto os interesses como as mensagens de dados normais não são processadas pelo PAR. Por ter prioridade maior que o Gradiente, quando uma mensagem exploratória é recebida pelo nó, ela é encaminhada pelo núcleo da difusão direcionada para o PAR. Este, ao invés de imediatamente encaminhá-la ao originador do interesse, configura um temporizador associado a esta mensagem e a compara com outras até que o temporizador expire. No processamento original da difusão direcionada, o filtro Gradiente encaminharia a primeira mensagem exploratória que

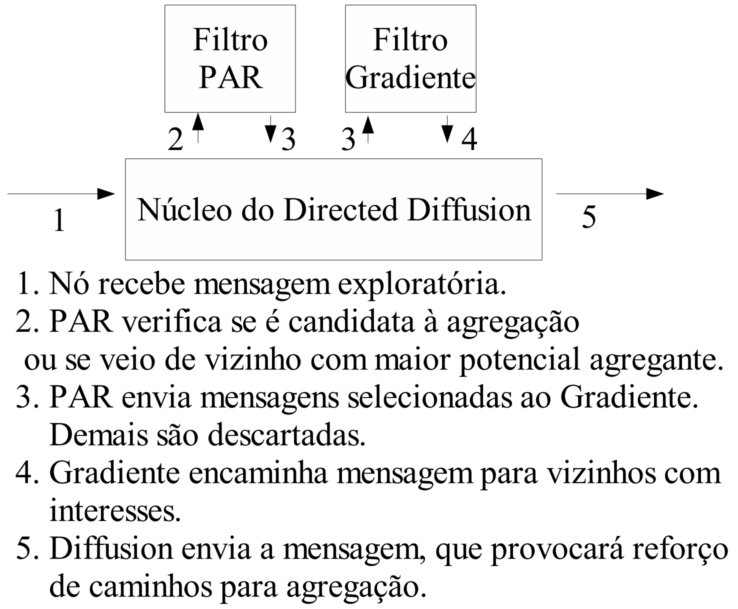


Figura 3.1: Filtro PAR

chegasse ao nó aos vizinhos e o descartaria as subseqüentes.

O objetivo de armazenar todas as mensagens exploratórias em um cache local é criar subsídios para novos critérios de seleção de rotas. Primeiramente, foram definidos quais seriam os parâmetros de comparação, descritos em 3.2.1. O funcionamento do mecanismo se dá em duas fases. A primeira acontece quando a mensagem chega no nó e é comparada com similares previamente recebidas para fins de agregação e será detalhada em 3.2.2. A segunda fase inicia quando o temporizador expira e um novo cheque é realizado para identificar qualquer alteração na fila de mensagens a serem transmitidas, como a chegada de melhor mensagem, e o envio ao filtro Gradiente da mensagem selecionada. Esta fase é descrita em 3.2.3.

3.2.1 Configuração do Filtro

O filtro foi configurado da seguinte forma: Para comparar as mensagens exploratórias, três atributos foram adicionados ao cabeçalho das mensagens originais da difusão direcionada:

- *distance_to_source*, que guarda o número de saltos entre a fonte e o primeiro nó agregador, indicando o quão distante está o ponto de agregação da fonte;
- *aggreg_nodes*, que armazena o número de nós agregadores existentes no caminho seguido até o presente nó; e
- *aggreg_ID*, que mantém uma indicação de quais fontes foram incluídas no processo de agregação.

Os dois primeiros atributos são usados para a comparação de mensagens, como será visto em 3.2.2. Mensagens recebidas de um vizinho em particular com os valores mais altos de *aggreg_nodes* e mais baixos de *distance_to_source* são escolhidas para encaminhamento, mesmo se recebidas depois de outras mensagens, que são descartadas. O terceiro atributo (*aggreg_ID*) é usado para evitar a computação duplicada da mesma mensagem no caminho de agregação, o que provocaria números errados para *aggreg_nodes* e poderia prejudicar o mecanismo de seleção de rotas.

Este último atributo é necessário porque um nó não tem conhecimento global da topologia e não possui um identificador único. Conhece apenas seus vizinhos, sendo transparentes para ele aqueles que estejam um salto além. Além disso, o objetivo deste trabalho é criar condições para suprimir, em nós intermediários, mensagens similares de fontes diferentes, uma vez que a supressão de duplicadas de mesma fonte já é efetuada no protocolo original.

Para tal, a difusão direcionada utiliza-se de dois atributos no cabeçalho da mensagem: *pkt_num* e *rdm_id*, contendo números aleatórios atribuídos internamente pelas fontes e incluídos na mensagem na sua geração. Estes atributos mantêm-se constantes enquanto a mensagem trafega. Assim, ao receber uma mensagem com *pkt_num* e *rdm_id* iguais aos de outra já recebida, o nó intermediário a descarta, pois trata-se de uma duplicata que seguiu outro caminho.

Da mesma forma, para o melhor funcionamento do PAR foi necessário acrescentar o atributo *aggreg_ID* para evitar agregações duplicadas, ou seja, marcar novamente para agregação duas mensagens similares das mesmas fontes F_1 e F_2, já agregadas em outro ponto da rede. A figura 3.2 ilustra melhor o problema.

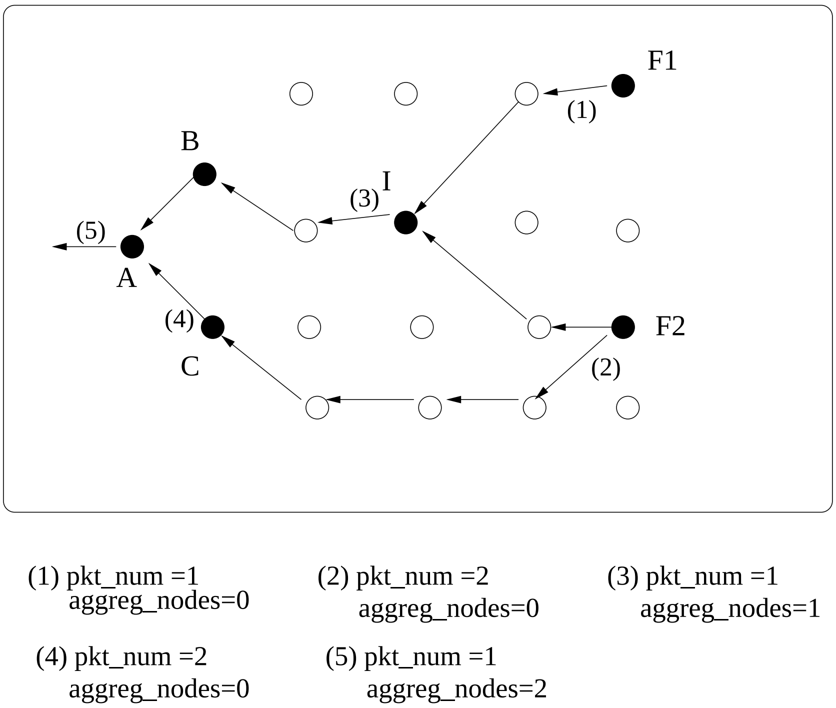


Figura 3.2: Agregação duplicada (***I*** e ***A***) de mensagens originadas em F_1 e F_2.

Ela representa um fragmento de um campo de sensores, onde foram destacados os nós ***A*** (objeto de potencial agregação), os nós ***B*** e ***C***, seus vizinhos, um nó intermediário ***I***, e duas fontes distantes F_1 e F_2. As Fontes geram dados similares, e suas mensagens recebem respectivamente *pkt_ num*=1 e *pkt_ num*=2.

Verifica-se que as duas mensagens são propagadas e eventualmente alcançam o nó *I*, sendo que a mensagem com *pkt_ num*=1 chegou antes (notar que agora a única informação disponível é o *pkt_ num*, o nó *I* desconhece tudo sobre as fontes). *I* então marca a mensagem 1 (incrementando o valor de *aggreg_ nodes*=1) para agregação com a mensagem 2 e a propaga na direção de *A*, passando por *B*. Em um outro fluxo, a mensagem de F_2, (pkt_num=2), chega até o nó *A*, passando por *C*.

Agora vejamos quais informações *A* possui. Este nó recebeu de *B* uma mensagem com *aggreg_ nodes*=1 (uma agregação no caminho) e emphpkt_num=1 e de *C* uma outra com *aggreg_ nodes*=0 (nenhuma agregação no caminho) e *pkt_ num* =2. As duas mensagens são similares e atendem a seus critérios de comparação para agregação, e como evidenciado pelo seu *pkt_ num*, vêm de fontes diferentes. Logo, pelas informações disponíveis (ele não sabe que exatamente estas duas mensagens já foram agregadas em I), A vai marcá-las erroneamente para agregação, incrementando o *aggreg_nodes* de cada uma delas.

A solução adotada no presente trabalho foi a adição do atributo *aggreg_ID*, conforme visualizado na figura 3.3. Este atributo contém o somatório dos *pkt_num* das mensagens que já foram utilizadas para agregação com a presente mensagem e é atualizado em cada nó agregador.

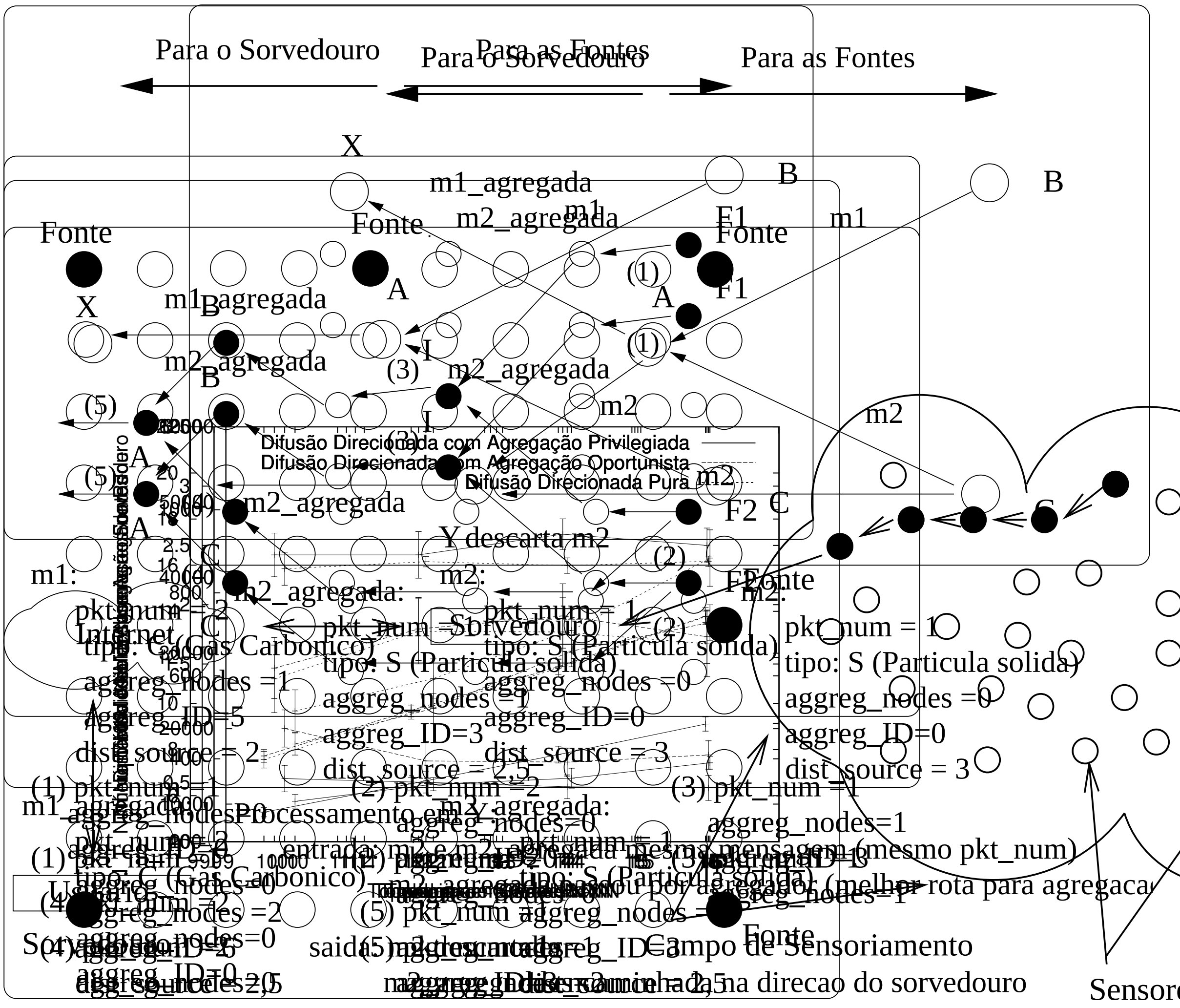


Figura 3.3: Supressão da agregação em ***A*** de duas mensagens já agregadas em ***I***.

Quando o nó I decide agregar a mensagem 1 com a 2, ele altera o *aggreg_ID* de ambas para o valor 3. Dessa forma, A recebe de B a mensagem *pkt_num* $= 1$ com *aggreg_ID* $= 3$ e de C a mensagem *pkt_num* $= 2$ com *aggreg_Id* $= 0$. A verifica que se somar o pkt_num =1 da mensagem vinda de B com o pkt_num = 2 da mensagem vinda de C obtém o valor 3, indicando uma agregação anterior em outro ponto da rede, e decide não agregá-las. Assim, com um custo computacional pequeno e o acréscimo de três atributos nas mensagens, foi possível o estabelecimento de fluxos de dados que forcem a passagem da informação por nós agregadores.

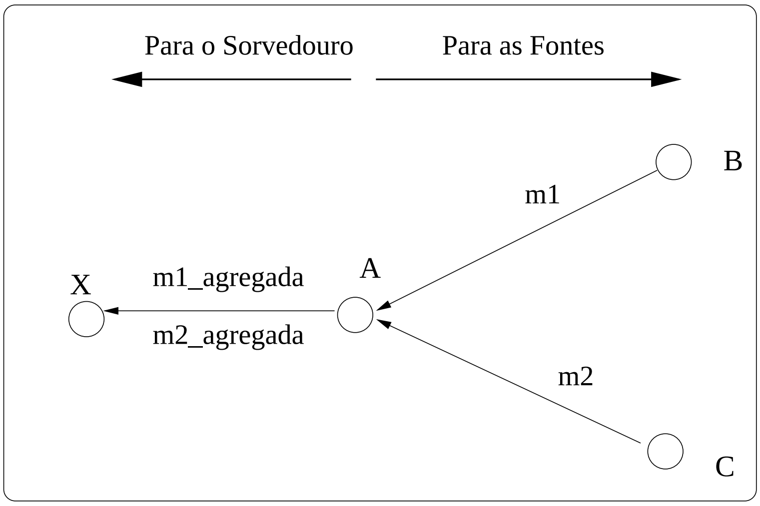
3.2.2 Primeira Fase: Armazenamento e Comparação de Mensagens

Para a primeira fase de processamento do PAR, foi criada uma estrutura de dados, chamada de *message entry*, para o armazenamento das mensagens exploratórias previamente recebidas pelo nó. Quando uma mensagem exploratória chega no filtro PAR, é comparada por meio da função *matchMessageEntry* com outras que já estejam armazenadas na *message entry.* Esta função retorna uma lista com mensagens de atributos similares aos da mensagem que acabou de chegar. Não é necessária a comparação com todos os atributos, mas somente com aqueles que são relevantes para a agregação desejada, que vai depender da aplicação.

Por exemplo, se a aplicação deseja monitorar a qualidade do ar de uma região metropolitana e disparar um alarme a partir de um determinado limiar de poluição, e os sensores ambientais dispostos no terreno fornecem dados sobre temperatura (T), pressão atmosférica (P), umidade relativa do ar (H), presença de fumaça (F), percentual de Oxigênio (O), de gás carbônico (C) e de partículas sólidas (S), potencialmente os atributos de interesse seriam F, O, C e S. Caso o objetivo da aplicação fosse a identificação de fogo em uma zona rural, com os mesmos sensores os atributos de interesse seriam potencialmente T, P, H e F.

Dessa forma, a função matchMessageEntry é a parcela de código do PAR que serve para sintonizá-lo com a aplicação e com outro filtro que processe a fusão de dados. É nela que são implementadas as regras de casamento do filtro. Uma vez definidos os atributos de interesse, quando a função é executada, ela extrai da mensagem estes atributos e compara com a *message entry*, retornando uma lista com as mensagens que parcialmente ou totalmente possuem os atributos de interesse, sendo então o processamento é dividido em dois passos.

O primeiro é ilustrado na figura 3.4 e trata da comparação de mensagens de fontes diferentes, para fins de agregação. É executado quando a mensagem da lista possui um *pkt_num* diferente da recentemente chegada (são de fontes diferentes) e o seu *aggreg_id* indica que elas não foram agregadas em outro ponto da rede, de forma que as mensagens são consideradas candidatas à agregação.



m1:
pkt_num = 2
tipo: C (Gas Carbonico)
aggreg_nodes =1
aggreg_ID=5
dist_source = 2

m2:
pkt_num = 1
tipo: S (Particula solida)
aggreg_nodes =0
aggreg_ID=0
dist_source = 3

m1_agregada:
pkt_num = 2
tipo: C (Gas Carbonico)
aggreg_nodes =2
aggreg_ID=6
dist_source = 2,5

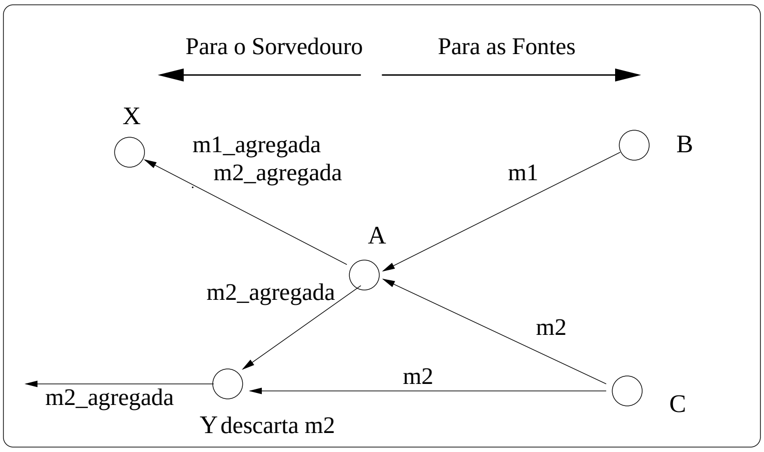
m2_agregada:
pkt_num = 1
tipo: S (Particula solida)
aggreg_nodes =1
aggreg_ID=3
dist_source = 2,5

Figura 3.4: Exemplo de nó agregador

Seus atributos *aggreg_nodes* são então incrementados e *aggreg_ID* atualizados, incluindo as fontes da agregação em curso. O atributo *distance_to_source* passa a guardar a distância média de ambas as mensagens até as respectivas fontes.

No exemplo da figura 3.4, as duas mensagens comparadas são de fontes diferentes, não foram agregadas antes, e possuem os atributos C (percentual de gás carbônico) e S (percentual de partículas sólidas) que foram de encontro aos interesses definidos na função matchMessageEntry. Estes atributos são comparados e modificados conforme ilustrado, o que caracteriza este nó como agregador.

O segundo passo é exemplificado na figura 3.5. Ele compara a mensagem que chegou com outras vindas da mesma fonte (mesmo *pkt_num*) para fins de descarte. Se uma mensagem com maior *aggreg_nodes* ou menor *distance_to_source* (nesta ordem) é encontrada, a atual é descartada e nenhum temporizador é criado para sua transmissão. Caso contrário, os temporizadores para as *piores* entradas existentes são removidos (evitando a sua propagação), elas são apagadas da estrutura de dados e um temporizador é criado para a nova mensagem.



m2_agregada:
pkt_num = 1
tipo: S (Particula solida)
aggreg_nodes =1
aggreg_ID=3
dist_source = 2,5

m2:
pkt_num = 1
tipo: S (Particula solida)
aggreg_nodes =0
aggreg_ID=0
dist_source = 3

Processamento em Y:
entrada: m2 e m2_agregada mesma mensagem (mesmo pkt_num)
m2_agregada passou por agregador (melhor rota para agregaca

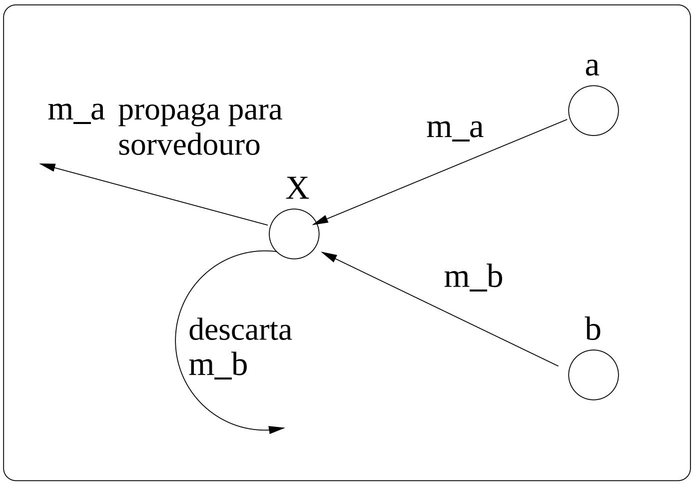
saida: m2 descartada
m2_agregada encaminhada na direcao do sorvedouro

Figura 3.5: Exemplo de Descarte de mensagem

3.2.3 Segunda Fase: Envio ao Filtro Gradiente

A segunda fase começa quando o temporizador expira e uma nova verificação por melhores mensagens é efetuada, antes da transmissão. Caso exista melhor mensagem, esta substitui aquela cujo temporizador expirou e é enviada em seu lugar ao filtro Gradiente, que a encaminhará aos vizinhos que possuam interesses. O filtro Gradiente seguirá agindo como normalmente faz, exceto que agora ele recebe mensagens exploratórias com melhores parâmetros de agregação, e então diferentes vizinhos são reforçados.

Por exemplo, supondo que um nó tenha dois vizinhos, *a* e *b*, dos quais recebe mensagens similares, e que a mensagem de *a* chega antes. Normalmente, o filtro Gradiente a encaminharia (menor latência) e descartaria a mensagem de *b*, sem maiores considerações, conforme ilustrado em 3.6.



1.m_a e m_b mensagens da mesma fonte que chegam em X por caminhos diferentes (por a ou b);

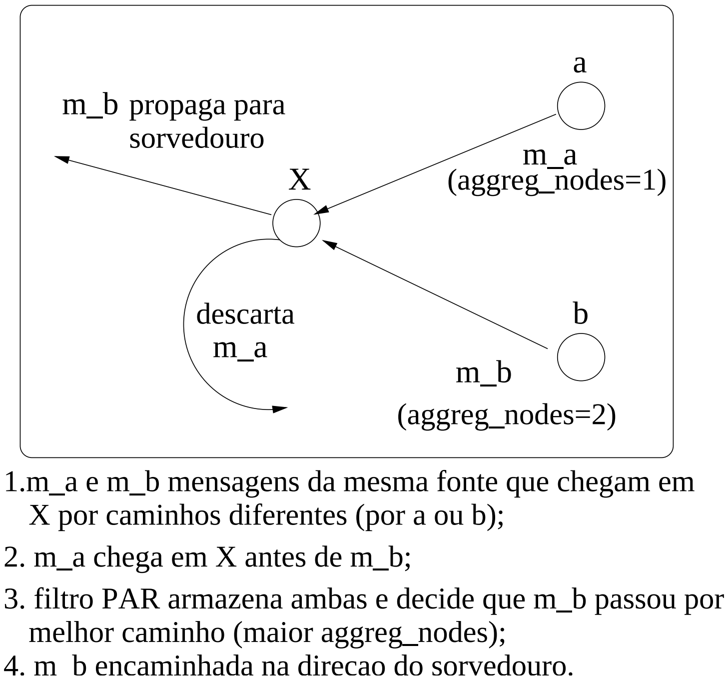
2. m_a chega em X antes de m_b;

3. filtro Gradient considera m_b uma duplicata de m_a e a descarta;

4. m_a encaminhada na direcao do sorvedouro.

Figura 3.6: Propagação pelo Filtro Gradiente

Com o PAR, poderia ser identificado que o valor de *aggreg_nodes* de *a* é 1, significando uma possível agregação de duas fontes distintas, enquanto o valor de *aggreg_nodes* de *b* é 2, indicando agregação de três fontes. O filtro PAR descartaria a mensagem vinda de *a* e encaminharia a mensagem de *b*, conforme ilustrado em 3.7. Para o filtro gradiente, a mensagem de *a* nunca existiu, então apenas a de *b* é encaminhada a frente. Quando o momento de reforçar um caminho chegar, o nó *b* será reforçado. O processo continua em cada nó intermediário até o sorvedouro ser alcançado e as rotas desejadas (agregantes) criadas.



1.m_a e m_b mensagens da mesma fonte que chegam em X por caminhos diferentes (por a ou b);
2. m_a chega em X antes de m_b;
3. filtro PAR armazena ambas e decide que m_b passou por melhor caminho (maior aggreg_nodes);
4. m b encaminhada na direcao do sorvedouro.

Figura 3.7: Propagação pelo Filtro PAR

3.3 Considerações Finais

Desta forma, sem alterações no filtro Gradiente e tampouco no núcleo do protocolo original, rotas contendo nós agregadores são identificadas e reforçadas. Empiricamente, tais rotas possuem mais nós agregadores e estes estão mais próximos das fontes, o que de acordo com [6] proporciona os maiores ganhos em número de transmissões.

É importante salientar que o filtro PAR não processa nenhuma técnica de agregação de dados, apenas preparando o caminho para que esta seja acentuada. Ao agir apenas nas mensagens exploratórias, não provoca influência significativa no tráfego da rede nem no retardo da transmissão de dados, visto que estas correspondem a uma pequena fração das mensagens geradas e somente são transmitidas durante a autoconfiguração da rede. Somente o tempo de descoberta de rotas será aumentado, tendo em vista o temporizador utilizado pelo PAR em cada nó intermediário da rede.

A opção de não modificar o filtro gradiente foi tomada para manter as características de robustez e escalabilidade da difusão direcionada, sem alterações estruturais que provocassem efeitos colaterais imprevistos.

O filtro PAR deve ser usado em conjunto com outro módulo de software que atue nas mensagens de dados (uma vez que o PAR atua nas exploratórias) empregando uma técnica de agregação adequada para agir na árvore de agregação estabelecida e comprimir dados. Para verificar a eficiência do filtro na seleção de rotas, ele foi testado em conjunto com outro filtro para agregação, utilizando inferência bayesiana. Este filtro foi implementado por [38].

Capítulo 4

Simulações e Resultados

O Mecanismo proposto precisa ser avaliado em relação ao seu objetivo pretendido de estabelecer uma árvore de agregação empírica, que se adapte às mudanças topológicas comuns às RSSF. Outro aspecto a ser avaliado é o impacto desta árvore de agregação na redução do consumo de energia na rede. Tendo em vista a grande diversidade de cenários e o grande número de sensores necessários para criá-los, a avaliação foi feita por meio de simulações por software.

Este capítulo descreve a avaliação do mecanismo proposto. A seção 4.1 descreve o ambiente de simulação. A seção 4.2 apresenta a avaliação qualitativa do filtro PAR e sua capacidade de estabelecer rotas que privilegiem agregação de dados de múltiplas fontes. A seção 4.3 quantifica a avaliação da árvore de agregação descoberta pelo PAR, submetendo-o a diversas situações em termos de topologia da rede, temporizador, número de nós e número de fontes. A seção refsec:energia apresenta a avaliação do quanto as novas rotas alteraram parâmetros sobre o consumo de energia na rede. Finalmente, a seção 4.5 descreve as considerações finais do capítulo.

4.1 Ambiente de Simulação

A ferramenta de simulação utilizada foi o *Network Simulator* (NS-2) [43], versão 2.27. Este é um simulador orientado a eventos, escrito em C++ e em outra linguagem orientada a objetos chamada de OTcl (MIT Object Tcl) [44]. Esta última linguagem é interpretada, permitindo que os scripts de simulação sejam modificados sem necessidade de recompilação de todo o simulador.

O NS foi desenvolvido no projeto VINT (*Virtual InterNetwork Testbed*) com o suporte da DARPA (*Defense Advanced Research Projects Agency*), e com a colaboração de diversos pesquisadores e institutos de pesquisas. A versão utilizada no presente trabalho foi obtida junto ao USC/ISI (*University of South California/ Information Sciences Institute*).

Para as simulações das redes de sensores, foi utilizada a extensão *diffusion3*, presente na distribuição do NS utilizada. Esta extensão foi desenvolvida no projeto SCADDS (*Scalable Coordination Architectures for Deeply Distributed Systems*) [45], do USC/ISI. A versão da API do Directed Diffusion utilizada nas simulações foi a 9.0.1 [46].

Como mencionado no capítulo 3, o PAR não implementa nenhuma técnica de agregação de dados. Seu objetivo é a construção da árvore de agregação. Para a avaliação do seu desempenho, ele foi usado em conjunto com um filtro para agregação de dados utilizando inferência bayesiana, chamado de Bayes e desenvolvido por [38]. Esta técnica é um método estatístico baseado na Regra de Bayes [47], pertencendo à classe de algoritmos que utiliza o conhecimento *a priori* sobre o espaço de observações para realizar inferências.

O método utilizado pelo filtro Bayes consiste em atualizar a crença de uma hipótese baseada na crença estimada anteriormente e em evidências adicionais detectadas pelos múltiplos sensores da rede, permitindo um mapeamento direto entre tais evidências e os eventos de interesse, o que é facilitado pelo esquema de endereçamento centrado em dados do *Directed Diffusion*. O mecanismo de agregação tem início após a definição das rotas feita pelo filtro PAR e só atua nas mensagens de dados. Desta forma, a atuação dos dois filtros é complementar: o PAR atua somente nas mensagens exploratórias preparando as rotas e o Bayes atua nas mensagens de dados inferindo sobre a tarefa da rede.

4.1.1 Cenários de Simulação

O cenário de simulação consiste de um campo de sensores na forma de uma grade quadrada, ilustrado na figura 4.1. Este formato para o campo foi escolhido para facilitar a análise dos efeitos das variações nos parâmetros de agregação sobre o número de transmissões na rede, bem como para a melhor visualização da árvore de agregação estabelecida. Ele é apropriado para avaliar as modificações introduzidas na Difusão Direcionada. Alguns cenários com topologia aleatória foram testados com resultados similares e publicados no artigo descrito em [41].

Nas simulações realizadas, existe um único sorvedouro, que será sempre o nó 0 (zero), localizado em um dos vértices do campo. Ele dissemina interesses, correspondentes às tarefas do usuário, e recebe os dados resultantes, originados em fontes dispostas nos quadrantes opostos da grade, afastadas do sorvedouro. Cada rodada de simulação durou 400 segundos, dos quais os 100 primeiros não foram computados, para que as simulações atingissem um estado estacionário, eliminando variações de configuração da rede. Dessa forma, todas as métricas foram computadas nos 300 segundos finais de cada simulação.

O tempo de espera para a computação das métricas é importante porque o próprio protocolo da Difusão Direcionada evidencia variações no tempo de descoberta de rotas ao longo das rodadas de simulação, e que as mensagens de dados somente são transmitidas pelas fontes após as rotas serem estabelecidas. Com a adição do filtro PAR, este tempo de descoberta de rotas é acentuado pelo temporizador implementado em cada nó, fazendo com que se retarde ainda mais o início da transmissão da informação ao sorvedouro. Caso não houvesse um período de aquecimento, os resultados poderiam não refletir a realidade, pois uma determinada configuração de rede poderia apresentar menos mensagens simplesmente por ter começado a transmitir depois. O tempo de 100s foi escolhido com bastante folga, pois o tempo máximo de descoberta de rotas para os campos com mais nós simulados (256) foi bem inferior a este limite.

Para uma melhor avaliação do desempenho da rede, este cenário genérico teve

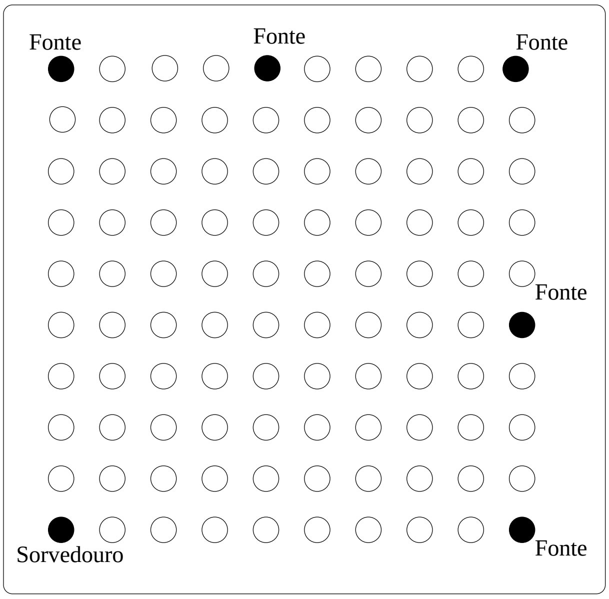


Figura 4.1: Cenário de Simulação

várias condições modificadas ao longo das simulações. Foram avaliadas duas situações de densidade dos nós na grade. Na primeira delas, cada nó está a 200 metros dos seus vizinhos e seu rádio possui alcance de transmissão de 250 metros, evitando assim a comunicação na diagonal entre dois nós. Na segunda, os nós estão distantes 150 metros um do outro, sendo mantido o mesmo alcance rádio e permitindo a comunicação na diagonal.

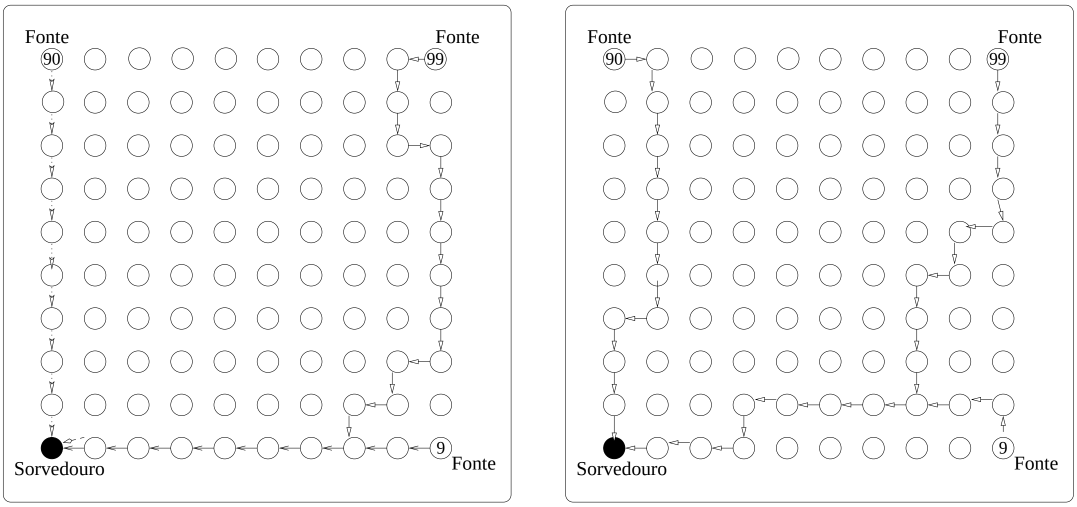
Em outros cenários, variou-se o tamanho da grade (de 100 a 256 nós), a quantidade de fontes (3 a 7), bem como o temporizador utilizado pelo Filtro PAR para a descoberta de Rotas.

Todas as simulações foram feitas para três implementações do protocolo de roteamento. A primeira é a Difusão Direcionada Pura, a segunda, que será chamada de agregação oportunista, a difusão direcionada com a utilização do filtro Bayes. Finalmente, a terceira implementação é o uso conjunto do PAR com o Bayes sobre a Difusão Direcionada, que será chamada de agregação privilegiada.

4.2 Análise da Árvore de Agregação

O primeiro passo nas simulações tratou de avaliar a capacidade do filtro PAR em modificar os fluxos de dados fonte/sorvedouro da difusão direcionada. Para esta avaliação, foi utilizado o cenário genérico com grade de 100 nós, 3 fontes, 1 sorvedouro, e os nós distantes 200 metros uns dos outros. Este cenário foi testado com e sem o filtro PAR.

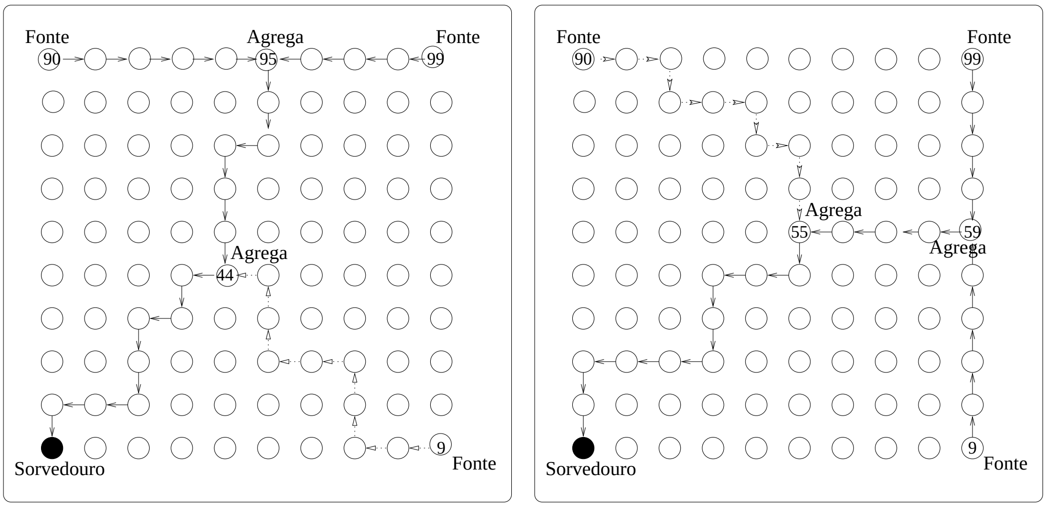
As rotas estabelecidas pela difusão direcionada, sem o PAR, seguiram caminhos independentes, que variavam ao longo das simulações, mas em geral com um padrão de menor retardo, como observado na figura 4.2. Este resultado já era esperado e corresponde ao comportamento natural do protocolo, que privilegia menor latência em detrimento da agregação de dados. Como será visto a seguir, os pontos de agregação também ocorrem nas rotas, produzindo o que se chama de agregação oportunista.



(a) Difusão Pura - Rota Preferida (b) Difusão Pura - Rota Alternativa

Figura 4.2: Rotas Mais Comuns - Difusão Pura.

Em 4.2(a) pode ser observado um fluxo e em 4.2(b) é ilustrado um alternativo. Esta variação freqüente das rotas evita a sobrecarga em nós com uso muito freqüente, mantendo o balanço de energia na rede. Ao adicionar o filtro PAR, rotas não mais independentes foram estabelecidas, e podem ser visualizadas na figura 4.3.



(a) Filtro PAR - Rota Preferida (b) Filtro PAR - Rota Alternativa

Figura 4.3: Rotas mais comuns - Filtro PAR.

Assim, por existirem três fontes, as rotas preferidas alternaram-se entre dois casos: os fluxos das fontes 90 e 99 (próximas uma da outra) agregando-se no nó 95 e posteriormente juntando-se ao fluxo da fonte 9 no nó 44, como observado na figura 4.3(a); e os fluxos das fontes 9 e 99 (também próximas) agrupando-se em 59 e posteriormente juntando-se ao fluxo de 90 no nó 55, como visto na figura 4.3(b). Ambos os casos são similares em termos de potencial economia de energia. Variações em torno destes nós agregantes também foram observadas, pois em algumas rodadas outros nós foram agregadores, como o 45 ou 54, de forma a também manter um certo balanço de energia na rede.

Conforme descrito no capítulo 3, o número de fluxos de PR (*Positive Reinforcement* ou Reforço Positivo) depende da periodicidade das mensagens de dados exploratórias na rede. Na simulação em questão, foram enviadas mensagens exploratórias em três momentos distintos da simulação, e em cada um deles foi reforçado um caminho, não necessariamente o mesmo. Enquanto esta rota estiver "ativa", toda mensagem de dados originada pela fonte a seguirá até o sorvedouro. Após a escolha de um novo caminho (novo ciclo de PR), o antigo e o novo serão utilizados pela rede por um certo tempo, até que finalmente somente o novo seja usado. É desta forma que a Difusão Pura recupera-se de elos quebrados na cadeia fonte-sorvedouro.

98	97	96	95	85	84	74	73	72	62	52	42	32	22	21	11	10	*	*
89	79	69	59	49	48	38	28	18	17	16	15	14	13	3	2	1	*	*
98	88	78	79	69	59	49	39	29	28	18	17	7	6	5	4	3	2	1
8	7	6	5	4	3	2	1	*	*	*	*	*	*	*	*	*	*	*
19	18	17	16	15	14	13	3	2	1	*	*	*	*	*	*	*	*	*
8	7	6	5	4	3	2	1	*	*	*	*	*	*	*	*	*	*	*
91	81	82	72	62	52	42	32	31	21	11	1	*	*	*	*	*	*	*
91	81	71	61	51	41	31	30	20	10	*	*	*	*	*	*	*	*	*
80	70	60	50	40	30	20	10	*	*	*	*	*	*	*	*	*	*	*

Tabela 4.1: Rotas Descobertas - Difusão Pura

A tabela 4.1 apresenta as rotas das fontes 99, 9 e 90 para o sorvedouro 0 (zero) em uma rodada de simulação completa. A tabela apresenta três fluxos por fontes, sendo que cada fluxo representa um ciclo completo de reforço positivo no caminho considerado.

Os resultados em termos de agregação de dados obtidos nesta simulação com Difusão Pura foram resumidos na tabela 4.2. Aqui pode ser observado que a Difusão direcionada pura já tem potencial para uma agregação oportunista na rede. Ou seja, apesar de não forçar a presença de nós agregadores nos caminhos preferidos, eles naturalmente ocorrem.

No fluxo 1, existe o nó 72 agregando dados das fontes 99 e 9, e distando 6 saltos das fontes, enquanto o nó 1 agrega dados das fontes 9 e 90, e está a nove saltos das mesmas. Neste fluxo, não existe nenhum nó agregador para as fontes 99 e 90. Nos outros dois fluxos encontrados nesta simulação, apenas um nó agregador foi encontrado, para as fontes mais próximas uma da outra. Desta forma, percebe-se que o fluxo 1 parece ser melhor para agregação, porém a tabela 4.1 mostra que ele é bem maior que os demais.

As tabelas 4.3 e 4.4 apresentam os resultados quando o filtro PAR é adicionado ao campo de sensores. Na tabela 4.3 pode ser observado que com o PAR o tamanho dos fluxos (número de saltos até o sorvedouro) aumenta, ficando mais parecido

Fluxo	**Fontes**	**Nó Agregador**	**Dist Fonte**
1	99 e 9	72	6
	9 e 90	1	9
2	99 e 9	10	13
3	99 e 90	3	10

Tabela 4.2: Nós Agregadores - Difusão Pura

98	97	96	95	85	75	65	55	54	53	52	51	41	31	21	20	10	*
89	79	69	59	58	57	56	55	54	44	34	24	23	22	12	2	1	*
89	79	69	59	58	57	56	55	54	53	43	42	41	40	30	20	10	*
19	29	39	49	59	58	48	47	46	45	35	34	33	23	22	12	11	1
19	29	39	49	59	58	57	56	46	36	26	25	24	23	22	12	2	1
19	29	39	49	59	58	57	56	55	54	53	43	33	23	13	12	11	10
91	92	93	94	95	85	75	65	55	45	35	25	24	23	22	12	11	10
91	92	93	94	84	74	64	54	44	43	33	23	13	3	2	1	*	*
91	92	93	94	84	74	64	54	44	34	33	32	31	21	11	10	*	*

Tabela 4.3: Rotas Descobertas - Filtro PAR

com o fluxo 1 da tabela 4.1. A tabela 4.4 ilustra que este aumento do tamanho foi compensado com uma maior presença de nós agregadores em cada fluxo, teoricamente melhorando as características para agregação de dados. Estes resultados serão avaliados em termos quantitativos em 4.3.

4.3 Análise dos Parâmetros de Agregação

Os resultados obtidos nas simulações iniciais e a análise qualitativa realizada na seção anterior mostraram que o filtro PAR efetivamente altera as rotas criadas pelo protocolo da Difusão Direcionada, tornando-as mais suscetíveis à agregação de dados, mas também mais longas. Cabe agora uma avaliação quantitativa do efeito das novas rotas sobre os principais parâmetros de agregação. Para tal, foram executadas 20 simulações de cada cenário e todos os resultados foram obtidos com

Fluxo	**Fontes**	**Nó Agregador**	**Dist Fonte**
1	99 e 90	95	4
	9 e 90	45	9
2	99 e 90	59	4
	99 e 9	54	8
	99 e 90	2	15
3	99 e 90	59	4
	99 e 9	10	16
	99 e 90	11	15

Tabela 4.4: Nós Agregadores - Filtro PAR

um intervalo de confiança de 95 %. As simulações foram feitas para que fossem analisadas as quatro métricas da árvore de agregação:

- Número de nós agregadores por fluxo de dados;
- Distância média dos nós agregadores às fontes (em número de saltos);
- Tempo de descoberta de rotas em função do temporizador do filtro PAR;
- Tamanho médio de cada fluxo de dados fonte/sorvedouro (em número de saltos).

Os dois primeiros parâmetros indicam os benefícios que o filtro PAR proporciona e para os quais foi desenvolvido. Os dois últimos indicam os custos relacionados com esta abordagem. Do balanço entre estes indicadores é que sairá a eficiência do filtro.

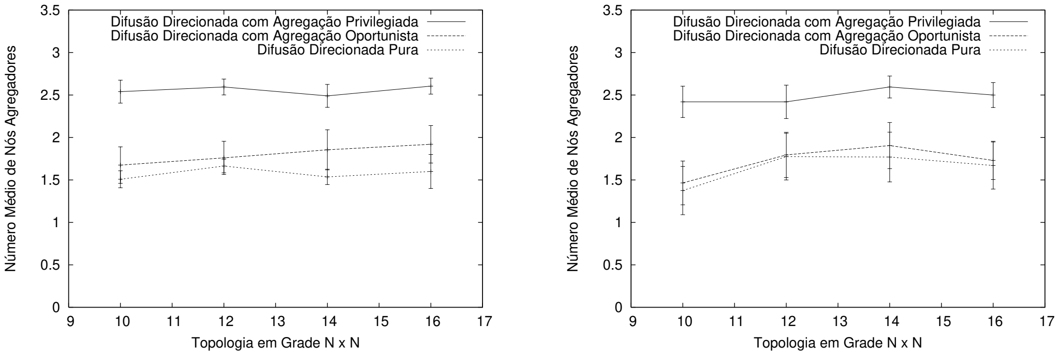
Três diferentes cenários foram testados. O primeiro variou a densidade dos nós na rede para identificar seus efeitos no funcionamento do PAR. O segundo buscou avaliar os efeitos positivos e negativos do aumento do temporizador do filtro PAR na descoberta de rotas. Finalmente, foi testado um cenário onde se variou o número de fontes.

4.3.1 Efeito da Densidade de Nós na Rede

Para a análise do número de nós agregadores e da distância média dos mesmos às fontes, foram feitas simulações com tamanhos variáveis do campo de sensores. Foram exercitados cenários com grades de 10 x 10, 12 x 12, 14 x 14, e 16 X 16 nós. Em todos os cenários foram usadas 03 fontes. Para este número de fontes, existem no máximo 03 (três) nós agregadores por fluxo (combinação de três fontes, duas a duas).

Os resultados da figura 4.4 mostram que a redução da distância entre os nós para 150 metros, possibilitando a comunicação na diagonal, não produziu efeito significativo na quantidade de nós agregadores, que continuou no mesmo patamar do cenário em que a distância foi de 200 metros (menor densidade de nós).

As simulações mostram que nos dois casos o filtro PAR apresentou melhoras significativas neste parâmetro, para todos os tamanhos da grade testados. Os resultados da Difusão Direcionada pura e da agregação oportunista foram equivalentes e inferiores. Enquanto o PAR identificou fluxos com média de 2,6 nós agregadores, as simulações sem o uso do PAR identificaram fluxos com média de 1,7 nós agregadores, com ligeiras variações de acordo com o tamanho da grade. Estes resultados permitem antecipar um efeito positivo na redução do número de mensagens que chega ao sorvedouro, devido à maior quantidade de agregações na rede.

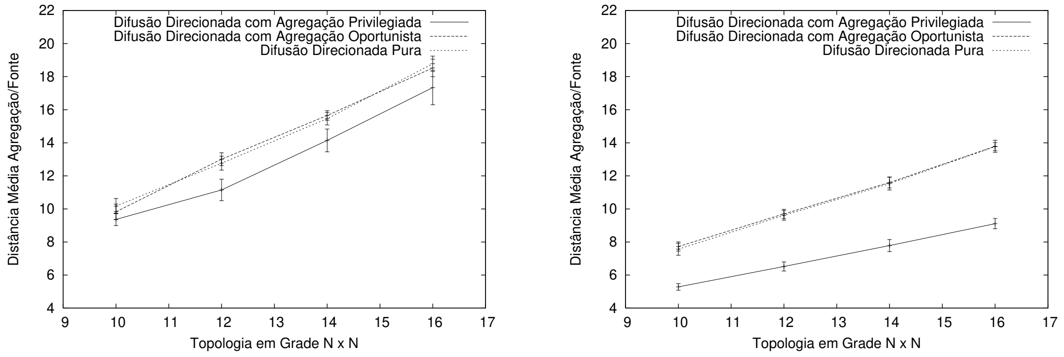


(a) Grades sem comunicações na diagonal

(b) Grades com comunicações na diagonal

Figura 4.4: Efeitos da Densidade dos nós na descoberta de nós agregadores.

Por outro lado, como pode ser visto na figura 4.5, a comunicação na diagonal trouxe as agregações para pontos mais próximos das fontes. Ainda que o Filtro PAR já tenha produzido ganhos em relação à difusão direcionada pura e agregação oportunista em 4.5(a), estes foram acentuados significativamente em 4.5(b). Percebe-se comparando essas duas figuras, que a comunicação na diagonal estabelece "atalhos"ao sorvedouro, que são utilizados pelas três propostas de roteamento, porém a agregação privilegiada conseguiu reduzir ainda mais a distância das agregações às fontes, contribuindo para a diminuição do número de transmissões na rede.



(a) Grades sem comunicações na diagonal

(b) Grades com comunicações na diagonal

Figura 4.5: Efeitos da Densidade dos nós distância das agregações às fontes.

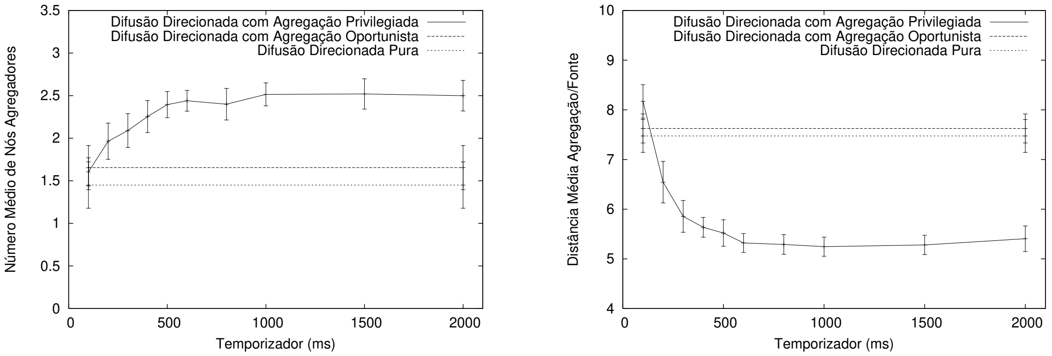
Este resultado pode ser explicado pelo mecanismo pelo qual o PAR escolhe os caminhos, apresentado no capítulo 3. A distância da agregação às fontes só é usada como fator para decisão após a consideração da quantidade de nós agregadores presentes no fluxo. No primeiro cenário (onde não há comunicação na diagonal), as opções de vizinhos são poucas. Quando se permite a comunicação na diagonal, aumentam-se as opções de escolha, e surge um caminho mais curto para se atingir o sorvedouro.

Desta forma, entre dois caminhos com a mesma quantidade de nós agregadores, a decisão vai para o mais curto. Ou seja, o aumento das opções de caminhos manteve o mesmo número de nós agregadores nos fluxos, mas reduziu a sua distância às fontes. Dessa forma, quanto maior a densidade de nós na rede, maiores as opções de caminhos, e assim melhor fica o desempenho do filtro PAR.

4.3.2 Efeito do Temporizador na Construção da Árvore de Agregação

O objetivo deste item é identificar o melhor valor para o temporizador do filtro PAR, no sentido de maximizar os parâmetros para agregação de dados e minimizar os retardos. Foram testados cenários para diversos tamanhos de grade.

A figura 4.6 mostra os efeitos da variação do temporizador para a grade de 10 x 10. Em 4.6(a) estão os resultados em termos de quantidade de nós agregadores. Em 4.6(b) estão as distâncias médias dos mesmos às fontes. Percebe-se que até o valor aproximado de 600 ms existe uma melhora crescente nos dois parâmetros e a partir daí eles ficam constantes, independente do aumento do temporizador. Neste caso, aumentar o temporizador significa aumentar a latência na descoberta de rotas sem nenhuma influência na sua capacidade agregadora.

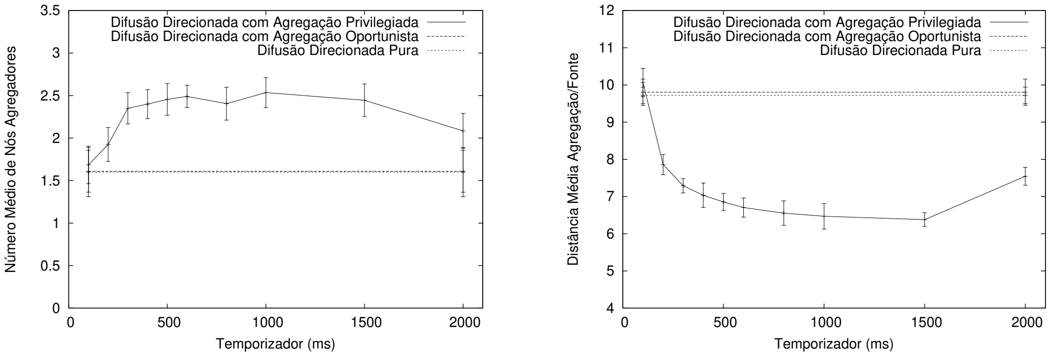


(a) Nós Agregadores presentes nos Fluxos de Dados

(b) Distância Média dos Pontos de Agregação às fontes

Figura 4.6: Efeitos do Temporizador em grade de 10 x 10 nós.

A figura 4.7 mostra os mesmos resultados para uma grade de 12 x 12 nós. Aqui também existe uma melhoria progressiva nos parâmetros de agregação até o temporizador atingir 600 ms, com os mesmos tornando-se constantes a partir daí. Todavia, quando o temporizador atinge 1,5 segundos, percebe-se uma tendência negativa nos parâmetros de agregação. Este fato parece contradizer a hipótese lógica de que o aumento do temporizador só traria benefícios, uma vez que a "espera"maior por dados nos nós intermediários levaria a um maior número de nós agregadores por fluxo, ou pelo menos à manutenção do estado estacionário, uma vez atingido o máximo possível para a topologia. Sendo assim, o aumento do tamanho da grade (10 para 12 nós) teve um efeito não antecipado sobre a influência do temporizador.



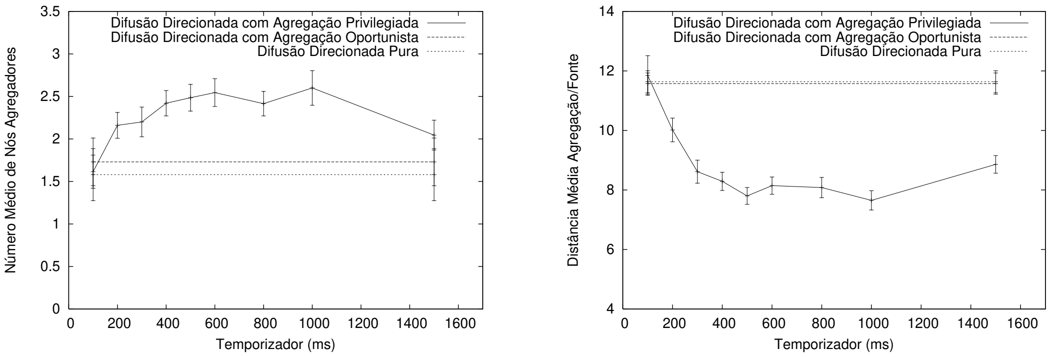
(a) Nós Agregadores presentes nos Fluxos de Dados

(b) Distância Média dos Pontos de Agregação às fontes

Figura 4.7: Efeitos do Temporizador em grade de 12 x 12 nós.

Os resultados para a grade de 14 nós estão na figura 4.8. Eles colaboram com os resultados obtidos na grade de 12 nós, porém a inflexão negativa nas curvas dos parâmetros ocorre para um valor de 1s no temporizador. Foi testada ainda uma grade de 16 nós, e seus resultados estão na figura 4.9. Agora, a tendência de melhoria nas curvas também se mantém até um valor de 600 ms para o temporizador, porém a inflexão negativa ocorre para um temporizador de 800 ms.

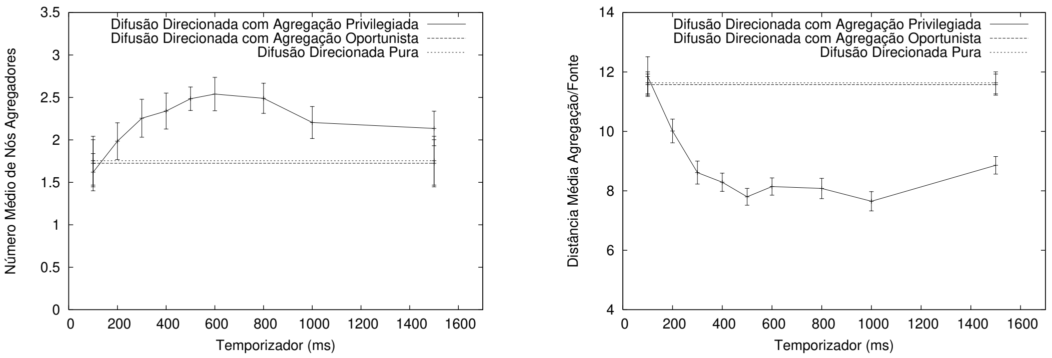
Estes resultados indicam uma faixa ótima de valores para o temporizador. O limite inferior para esta faixa se manteve constante em todos os tamanhos de grade utilizados, girando em torno de 600 ms. Este é o tempo de espera necessário para que as interações locais entre os nós vizinhos produzam os melhores parâmetros de agregação. A partir daí, o aumento do temporizador não influência os parâmetros de agregação, que se mantêm constantes, até um limite superior, variável de acordo com o tamanho da grade, a partir do qual existe uma tendência negativa nos parâmetros. Este limite superior, não antecipado, será discutido posteriormente, quando forem avaliados os efeitos colaterais na rede.



(a) Nós Agregadores presentes nos Fluxos de Dados

(b) Distância Média dos Pontos de Agregação às fontes

Figura 4.8: Efeitos do Temporizador em grade de 14 x 14 nós.



(a) Nós Agregadores presentes nos Fluxos de Dados

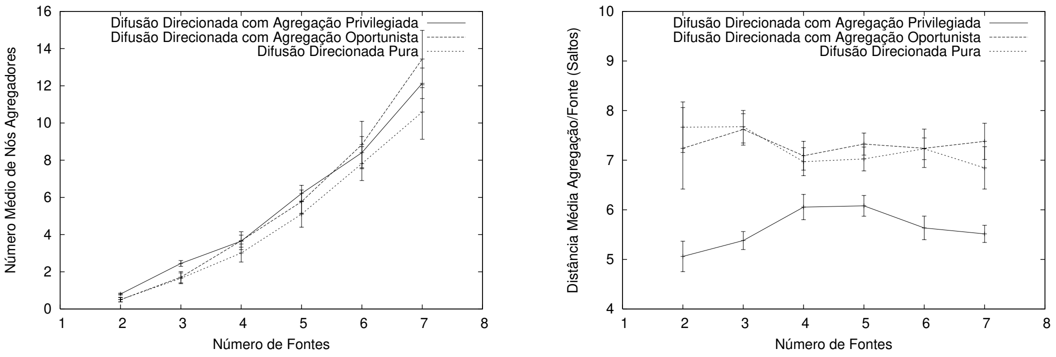
(b) Distância Média dos Pontos de Agregação às fontes

Figura 4.9: Efeitos do Temporizador em grade de 16 x 16 nós.

4.3.3 Efeito da Variação do Número de Fontes

Neste item, é analisada a influência da variação do número de fontes sobre os parâmetros de agregação. Foram realizadas simulações com uma grade de 10 nós e um temporizador de 0,5 segundos, onde se variou o número de fontes de 02 a 07. A figura 4.10 traz os resultados.

Cabe aqui salientar que para duas fontes existe somente um nó agregador, para três fontes existem no máximo 03 agregadores, enquanto 04 fontes podem propiciar no máximo 06, 05 fontes no máximo 10 agregadores, e assim por diante (combinação



(a) Nós Agregadores presentes nos Fluxos de Dados

(b) Distância Média dos Pontos de Agregação às fontes

Figura 4.10: Parâmetros para Agregação em Função do Número de Fontes.

de n fontes, duas a duas). Em 4.10(a) pode ser verificado que o filtro PAR não proporcionou uma melhoria significativa na quantidade de nós agregadores, à medida que se variavam o número de fontes no campo. Em compensação, em 4.10(a), o seu impacto na diminuição da distância das agregações às fontes é significativo.

4.3.4 Efeitos Colaterais

Uma vez analisadas as duas métricas de interesse para agregação de dados, resta analisar os efeitos colaterais, evidenciados pelos outros dois parâmetros de interesse. A figura 4.11 mostra o tempo de descoberta das rotas em função do temporizador para diversos tamanhos de grade. Como observado na figura, o temporizador exerce uma grande influência no tempo necessário para descoberta de rotas. Isto ocorre porque, como o PAR é utilizado em todos os nós da rede, a mensagem exploratória deve esperar em cada nó intermediário antes de ser encaminhada. Com isso, o seu efeito é acentuado à medida que o tamanho da grade é aumentado.

Este maior retardo na descoberta das rotas, acentuado nas grades com mais nós, explica o seu efeito sobre os parâmetros de agregação anteriormente discutidos. Isto acontece porque o temporizador do PAR age em conjunto com um parâmetro interno da Difusão Direcionada, que é o período em que os fluxos de mensagens exploratórias

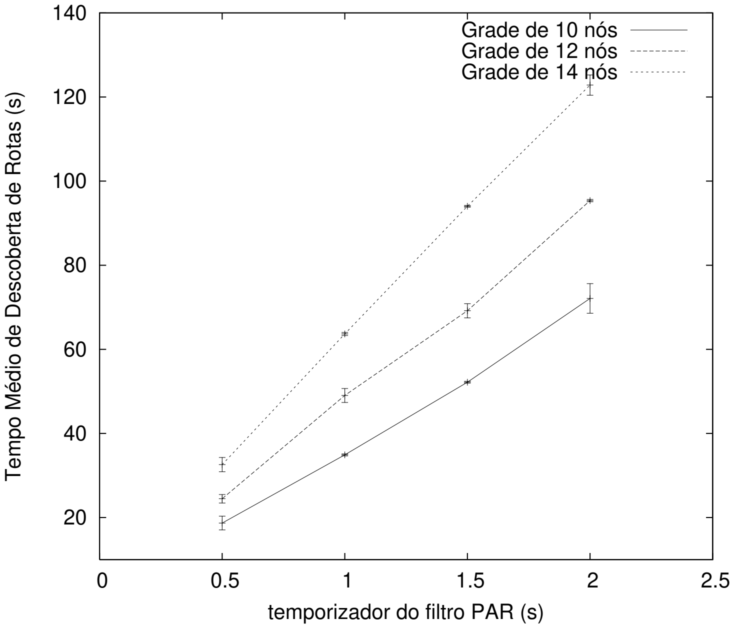


Figura 4.11: Tempo de Descoberta de Rotas.

inundam a rede. Como já mencionado no capítulo 3, os dados gerados pelas fontes são periodicamente transformados em mensagens exploratórias, utilizadas para descoberta de rotas e correção de caminhos quebrados.

O aumento exagerado do temporizador provoca um tempo de descoberta de rotas grande o suficiente para que um novo fluxo de mensagens exploratórias seja gerado antes do reforço positivo referente ao fluxo atual de mensagens chegar à fonte. Com isso, a decisão de caminhos nos nós intermediários fica confusa, pois duas mensagens exploratórias da mesma fonte coexistem no seu cache local. À medida que o temporizador vai crescendo, este fenômeno também atinge nós mais distantes das fontes, até que os resultados em termos de capacidade de agregação descem aos patamares da Difusão Pura.

Nas simulações realizadas, este fenômeno passou a acontecer para um temporizador de 1,5 segundos na grade de 12 nós, de 1 segundo para a grade de 14 nós e 800 ms para a grade de 16 nós. Ele não foi observado na grade de 10 nós, onde todos os fluxos de mensagens exploratórias foram concluídos sem problemas. Estes resultados sugerem que à medida que a grade aumenta de tamanho, o limite máximo do temporizador deve diminuir para que o roteamento não seja prejudicado e ao mesmo tempo reduzindo os impactos na latência da descoberta de rotas, que podem acarretar significativos prejuízos para o fluxo da informação da fonte para o sorvedouro. Significa também que para campos de sensores ainda maiores, a periodicidade do envio de mensagens exploratórias deve ser aumentada, para não prejudicar a descoberta de rotas pelo PAR.

A última métrica a ser analisada na árvore de agregação diz respeito ao tamanho médio dos fluxos de dados. A figura 4.12 mostra os resultados para diversos tamanhos de grade. Como esperado, é verificado que o PAR provoca que os fluxos de dados sejam maiores que na Difusão Pura. Isto ocorre porque para encontrar no meio dos caminhos nós agregadores, o PAR precisa desviar os fluxos do seu caminho de menor retardo (mais curtos). Estes fluxos maiores acarretam um maior número de transmissões na rede, que é justamente o que se pretende diminuir.

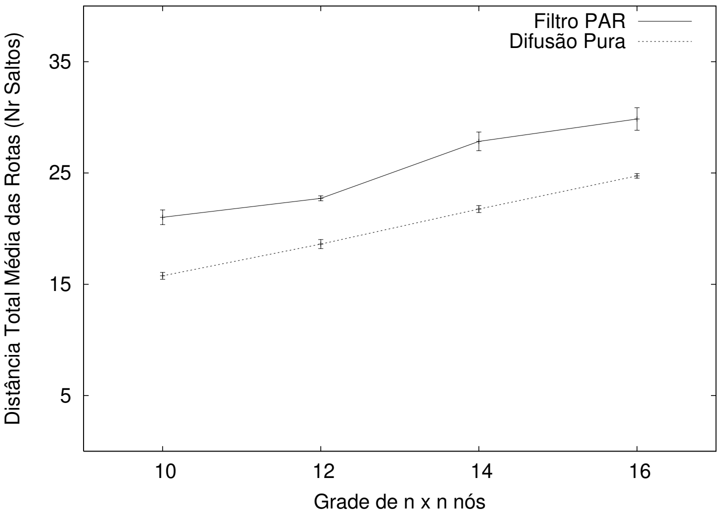


Figura 4.12: Tamanho Médio das Rotas Fonte/Sorvedouro.

Este é um efeito colateral relevante, e a eficiência do filtro no consumo de energia, analisada na seção 4.4, depende da capacidade do PAR em produzir fluxos comuns para várias fontes, com pontos de agregação bastante próximos das mesmas, compensando assim o número de saltos maior em cada fluxo.

4.4 Análise do Consumo de Energia

Na seção 4.3 foi avaliada a capacidade do filtro PAR em influenciar alguns parâmetros que favorecem a agregação de dados na rede. Aqui será avaliado o quanto a modificação destes parâmetros é capaz de reduzir o consumo de energia. Para tal, foram utilizadas duas métricas: O número de mensagens que chega ao sorvedouro, indicando o quanto houve de agregação; e o número total de transmissões efetuadas, indicando efetivamente o quanto o PAR influencia o consumo de energia, tendo em vista que produz fluxos maiores que a Difusão Pura.

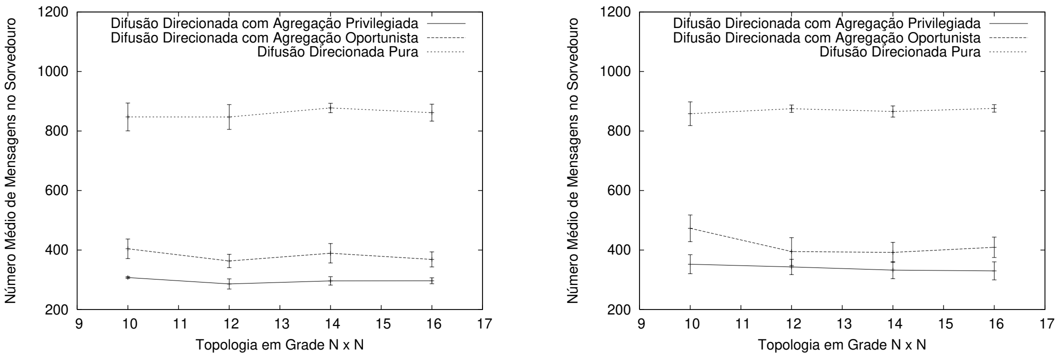
Estas duas métricas foram escolhidas porque mapeiam diretamente os efeitos das modificações propostas neste trabalho. A primeira métrica está diretamente relacionada à quantidade de nós agregadores na rede e a segunda à distância dos mesmos às fontes. Para uma melhor visualização destas relações, nesta seção estão apresentados os gráficos complementares aos da seção anterior, com a variação da densidade de nós, do temporizador e do número de fontes.

4.4.1 Número de Mensagens no Sorvedouro

Com a aplicação distribuída dos filtros PAR e Bayes na rede, foi possível avaliar o desempenho, utilizando-se os mesmos cenários da seção 4.3. O número de mensagens que chega ao sorvedouro foi avaliado pela comparação dos resultados do *Directed Diffusion* puro, do Bayes fazendo agregação oportunista e do Bayes com o PAR. Para todos os cenários, tendo em vista o discutido na seção anterior, foi usado um temporizador de 600 ms para o filtro PAR.

A figura 4.13 mostra os resultados obtidos para diversos tamanhos de grade, com a variação da distância entre os nós da grade discutida em 4.3.1. Nesta figura pode ser observada a redução do número de mensagens fruto da agregação oportunista do filtro Bayes, acentuada significativamente pelo emprego do PAR, em todos os tamanhos de grade simulados.

Isto comprova a hipótese de que a árvore de agregação produzida pelo PAR, ao forçar a presença de nós intermediários capazes de agregação nos fluxos de dados, acentua os efeitos de técnicas de agregação de dados distribuídas na rede. Pode também ser observado que, assim como não houve impacto da densidade dos nós na quantidade de nós agregadores, também não houve na redução do número de mensagens no sorvedouro, comprovando a relação direta entre os dois parâmetros.



(a) Grades sem comunicações na diagonal

(b) Grades com comunicações na diagonal

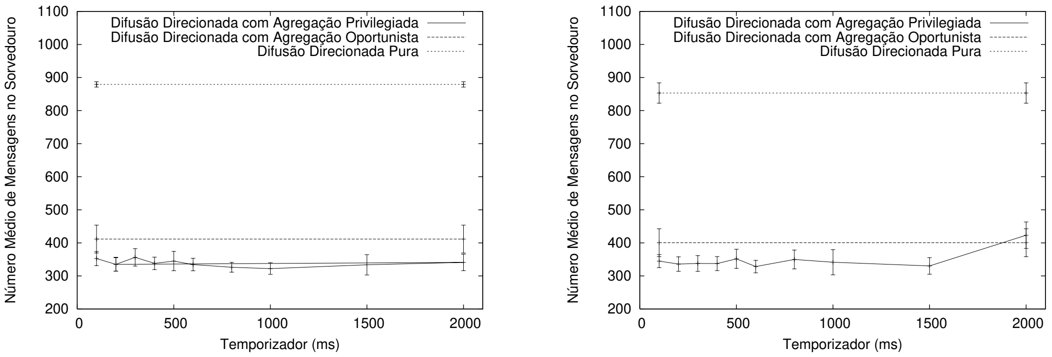
Figura 4.13: Efeitos da Densidade dos nós no número de mensagens no sorvedouro.

Nas figuras 4.14 e 4.15 estão os resultados da variação do temporizador para todos os tamanhos de grade testados. Quando comparados aos resultados obtidos para os parâmetros de agregação, eles também mostram uma relação direta entre o número de nós agregadores nos fluxos de dados e o número de mensagens que chegam no sorvedouro.

Pode, inclusive, ser observada a inflexão das curvas do PAR com o aumento do temporizador (1,5s para a grade de 12, 1s para a de 14 e 0,8s para a de 16.

Na figura 4.16 estão os resultados obtidos com a variação do número de fontes, de duas até sete. Aqui também pode ser observada a influência dos parâmetros de agregação no desempenho da rede.

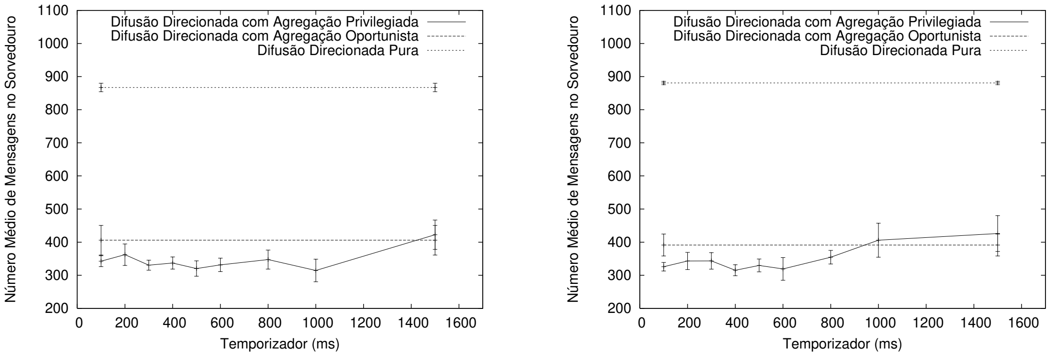
O primeiro aspecto a ser considerado é a redução drástica das mensagens que chegam no sorvedouro, fruto da agregação, seja oportunista ou privilegiada. À medida que se aumenta o número de fontes no campo, os efeitos da agregação são mais acentuados, uma vez que a redundância também aumenta. O segundo aspecto é



(a) Número Msg no Sorvedouro (10 x 10 nós)

(b) Número Msg no Sorvedouro (12 x 12 nós)

Figura 4.14: Efeitos do Temporizador no consumo de energia.



(a) Número Msg no Sorvedouro (14 x 14 nós)

(b) Número Msg no Sorvedouro (16 x 16 nós)

Figura 4.15: Efeitos do Temporizador no consumo de energia.

que, para os campos com mais de 3 fontes, os resultados do PAR foram ligeiramente inferiores ao da agregação oportunista. Ou seja, mais mensagens, ainda que por pequena margem, chegaram com o PAR do que sem ele.

4.4.2 Número Total de Transmissões na Rede

A segunda métrica diz respeito ao número total de transmissões (saltos) envolvendo todos os nós da rede. Esta métrica é importante, pois quanto mais longa é uma rota, maior é o número de saltos, mesmo que se mantenha o mesmo número

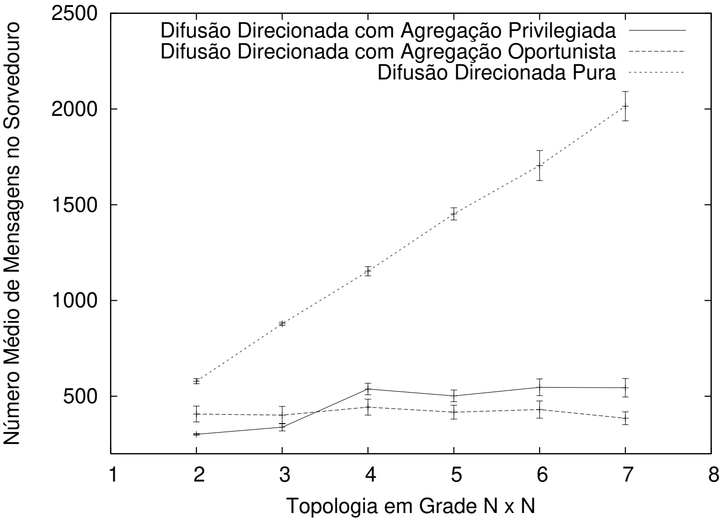
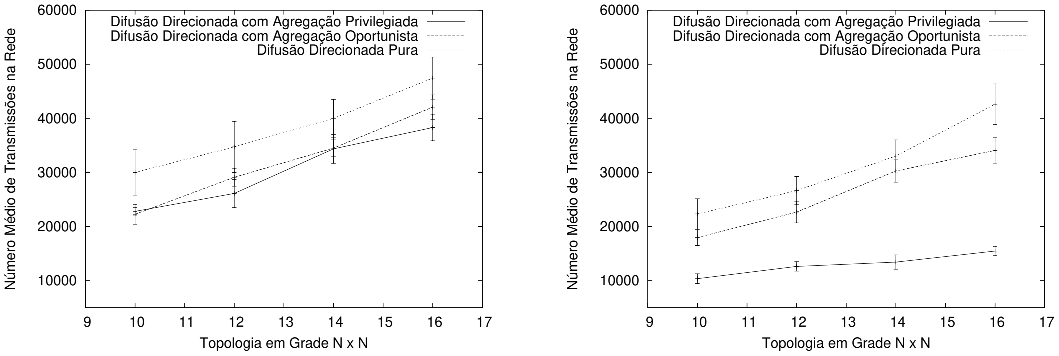


Figura 4.16: Número de mensagens no sorvedouro em função do número de fontes.

de mensagens chegando no sorvedouro. A sua relação com o consumo de energia é mais direta, pois quanto menos transmissões na rede, menor o consumo de energia nos rádios dos nós sensores.

Os resultados para vários tamanhos de grade e com duas distâncias entre os nós (200 e 150 metros) são apresentados na figura 4.17.



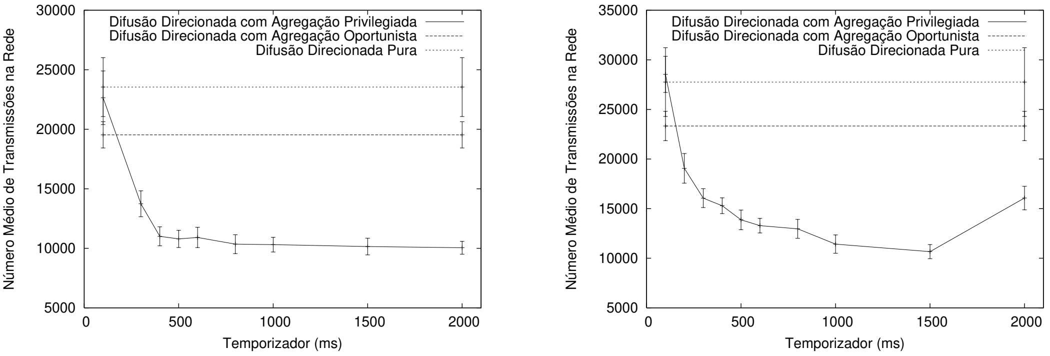
(a) Grades sem comunicações na diagonal)

(b) Grades com comunicações na diagonal

Figura 4.17: Efeitos da Densidade dos nós no número de transmissões na rede.

Aqui pode ser observada uma relação direta entre a distância às fontes e o número de transmissões. Com as transmissões na diagonal e o emprego do PAR, como visto em 4.17(b), estas distâncias foram bem menores, contribuindo para uma redução significativa no número total de transmissões, da ordem de pelo menos 50% para todos os tamanhos de grade.

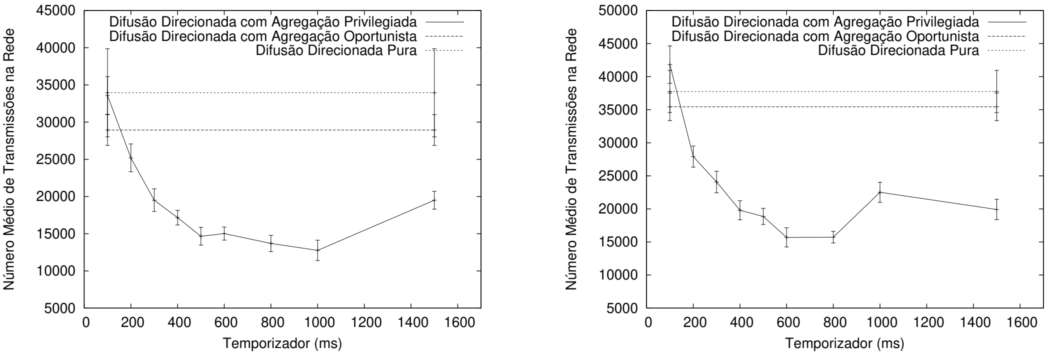
Nas figuras 4.18 e 4.19 estão os resultados da variação do temporizador. Todos os resultados são consistentes com os parâmetros de agregação, no caso a distância das agregações às fontes. As inflexões das curvas acontecem nos mesmos pontos discutidos anteriormente, o que comprova as conclusões obtidas.



(a) Número de Transmissões (10 x 10 nós)

(b) Número de Transmissões (12 x 12 nós)

Figura 4.18: Efeitos do Temporizador no consumo de energia.



(a) Número de Transmissões (14 x 14 nós)

(b) Número de Transmissões (16 x 16 nós)

Figura 4.19: Efeitos do Temporizador no consumo de energia.

A figura 4.20 mostra os resultados quando se varia o número de fontes. Neste caso, diferentemente do que acontece no número de mensagens, o filtro PAR apresenta vantagens em relação à agregação oportunista para todas as simulações, independente do número de fontes. Isto é perfeitamente explicado pelas agregações estarem mais próximas das fontes na árvore de agregação estabelecida.

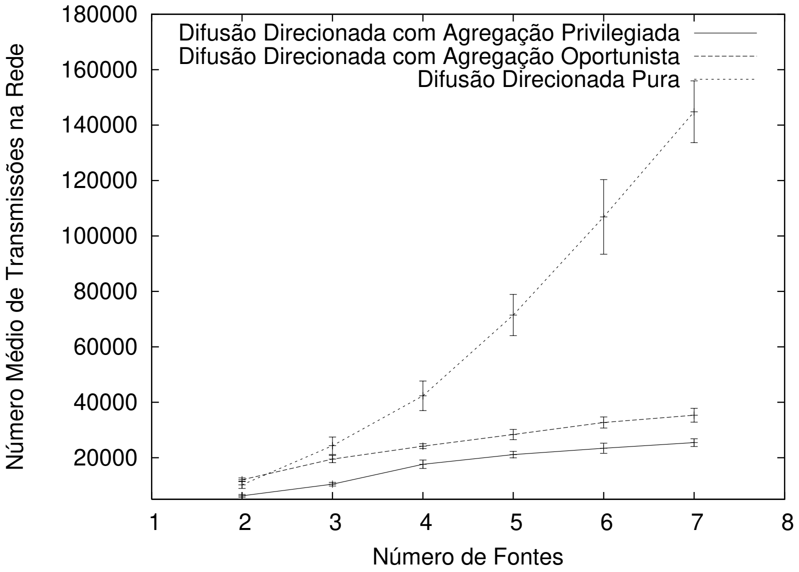


Figura 4.20: Número de transmissões em função do número de fontes.

4.5 Considerações Finais

Os diversos cenários simulados mostraram que a construção de uma árvore de agregação na rede, implementada por meio do filtro PAR sobre o protocolo da Difusão Direcionada, foi realizada com êxito. Os parâmetros ótimos para agregação de dados foram obtidos, seja em termos de quantidade de nós agregadores por fluxo de dados, seja na pequena distância dos mesmos às fontes.

As simulações comprovaram também os efeitos destes parâmetros sobre a economia de energia. Na medida que o número de nós agregadores nos fluxos de dados é incrementado, o número de mensagens que chega ao sorvedouro é reduzido. Da mesma forma, quanto menor a distância destas agregações às fontes, menor é o número total de transmissões na rede.

Entretanto, o PAR também provocou reações adversas, cujos custos devem ser balanceados com as vantagens citadas, de acordo com a aplicação. O primeiro deles é o tempo de descoberta de rotas, que aumenta progressivamente com o temporizador, até atingir valores que diminuem a própria eficiência do PAR em descobrir os caminhos agregantes.

As simulações realizadas apontaram para um temporizador ótimo em torno de 600 ms, a partir do qual o aumento do retardo na descoberta de rotas não traz melhoria nos parâmetros de agregação. Para aplicações que exigem respostas mais ágeis da rede, um temporizador menor pode ser utilizado, ao custo de uma menor capacidade agregadora.

Um outro aspecto importante diz respeito ao tamanho médio das rotas fontes-sorvedouro. Ao desviar-se do caminho mais curto para incluir nós agregadores, o PAR naturalmente produz rotas maiores, aumentando as transmissões por rota. Este aumento nas transmissões por rota deve ser compensado pelo menor número de fluxos agregados mais próximo das fontes.

Os melhores resultados do PAR ocorreram no cenário com maior densidade de nós, pois neste caso o número de transmissões na rede foi da ordem de 1:4 em relação à difusão pura e 1:3 em relação à agregação oportunista. Isto acontece porque, com uma maior densidade de nós, a oferta de caminhos alternativos é maior, e as chances de um nó intermediário possuir mais de um vizinho agregador são também maiores. Assim, ele pode decidir entre vários caminhos agregantes pelo mais curto. Em uma rede escassamente povoada, com poucos nós ao alcance do seu transmissor, isto não é possível.

Nos cenários em que se aumentou a quantidade de fontes, observou-se que o PAR não obteve muito sucesso na descoberta de nós agregadores, mas os trouxe para mais próximo das fontes, o que reduziu significativamente o número de transmissões na rede.

Capítulo 5

Conclusões e Trabalhos Futuros

OO bjetivo do trabalho descrito nesta dissertação foi o desenvolvimento de um mecanismo que identifique rotas ótimas para agregação de dados em redes de sensores, favorecendo desta forma o processamento distribuído e com isso, economizando energia com a redução do número de mensagens dos sensores que chegam ao sorvedouro.

As principais contribuições deste trabalho foram:

- a modificação do protocolo da Difusão Direcionada, por meio da implementação de um módulo de software (filtro PAR) para o estabelecimento de rotas que privilegiem agregação de dados;
- as simulações de diversos cenários de redes de sensores, com a identificação dos benefícios e dos custos de tal abordagem para as aplicações.

Este capítulo descreve as conclusões sobre o desempenho do filtro PAR e a experiência adquirida no seu desenvolvimento e apresenta sugestões para trabalhos futuros. A seção 5.1 descreve as observações finais. A seção 5.2 traz sugestões para trabalhos futuros, motivados pelas questões levantadas na pesquisa desenvolvida.

5.1 Conclusões

O mecanismo proposto para descoberta de rotas buscou otimizar o protocolo da Difusão Direcionada para a agregação de dados de múltiplos sensores na rede, abrindo espaço para o emprego de técnicas mais sofisticadas do que a simples supressão de mensagens duplicadas, que é feita no protocolo original. No caminho de múltiplos saltos seguido pelos dados de uma RSSF das fontes ao sorvedouro, a proposta foi privilegiar a quantidade de nós agregadores em detrimento do tempo de trânsito da mensagem até o sorvedouro.

Para tal, foi implementado o filtro PAR e integrado ao protocolo original. Ele foi instalado em todos os nós e atuou nas mensagens exploratórias, modificando os seus atributos antes de enviá-las ao filtro Gradiente (módulo básico do protocolo original). Este passou a enviar reforços positivos para diferentes vizinhos, colaborando para que as rotas finais fontes/sorvedouro fossem modificadas. Durante este processo, as características essenciais da Difusão Direcionada não foram alteradas, especialmente as interações locais entre um nó e seus vizinhos e o desconhecimento da topologia global da rede pelos nós sensores, de forma que não surgissem custos desnecessários para o estabelecimento e manutenção da árvore de agregação.

Na avaliação da árvore de agregação formada, constatou-se que efetivamente o PAR atingiu seus objetivos ao aumentar o número de nós agregadores na rede. Estes nós possuem um grande potencial para, ao receber dados de diferentes sensores, processar uma técnica de agregação paramétrica, inferir sobre o fenômeno de interesse, e encaminhar na direção do sorvedouro apenas uma mensagem com os resultados desta inferência. Os ganhos no aumento dos nós agregadores e na redução da sua distância às fontes foram consideráveis para os cenários simulados, porém a abordagem também proporcionou custos que, dependendo da aplicação, podem se tornar onerosos.

O principal deles diz respeito à latência na descoberta das rotas, que influi no tempo de espera do sorvedouro até a rede responder pela primeira vez ao seu interesse. Uma vez estabelecido o fluxo de dados, o PAR não acarreta mais latência.

Todavia, para aplicações de monitoração onde o tempo de resposta seja crítico, o temporizador necessário para que o PAR estabeleça a árvore de agregação pode tornar a abordagem inviável.

Foi observado que este temporizador possui um limite superior de utilização que varia de acordo com o tamanho do campo de sensores. Quanto maiores são as rotas entre fontes e sorvedouro, maior é a influência do temporizador, até um momento em que ele prejudica o filtro Gradiente na descoberta de rotas. As simulações realizadas indicaram que para cada tamanho de rede e quantidade de fontes, existe uma faixa ótima de valores para o temporizador.

Os fluxos de dados produzidos pelo PAR também foram maiores que os da difusão direcionada pura. Esta conseqüência é natural e já era antecipada, uma vez que para incluir nós agregadores nos fluxos de cada fonte, foi necessário desviá-los da sua rota mais curta para que em algum momento eles coincidissem com o fluxo de outras fontes. Todavia, fluxos maiores provocam um maior número de transmissões nos nós intermediários até que cada mensagem chegue ao sorvedouro. Este número maior de transmissões por fluxo foi compensado pela aproximação das agregações às fontes, obtidas nos cenários simulados.

Na avaliação dos efeitos dessa árvore de agregação sobre o consumo de energia na rede, foi observado que ela efetivamente acentua os efeitos de uma técnica paramétrica de agregação distribuída, reduzindo significativamente o número de mensagens que chegam no sorvedouro. Dessa forma, foi comprovada uma relação de causa e efeito entre a maior quantidade de nós agregadores e um menor número de mensagens de dados que chegam no sorvedouro. O impacto da agregação de dados sobre o número total de transmissões também foi positivo, indicando que as agregações ocorreram próximas às fontes, como desejado para compensar o maior tamanho dos fluxos.

Como última observação, o desempenho de um esquema deste tipo depende muito da perfeita sintonia entre o estabelecimento da árvore de agregação e a técnica de agregação utilizada. Dessa forma, os parâmetros selecionados para a escolha dos nós agregadores, o temporizador e as regras de seleção devem ser ajustados para extrair o máximo de ganhos da técnica utilizada. A posição relativa das fontes no campo de sensores também contribui para um melhor ou pior desempenho. Quanto mais dispersas forem as fontes, maiores serão os fluxos de dados para a agregação, e maior o custo na transmissão de mensagens em nós intermediários.

5.2 Sugestões para Trabalhos Futuros

Como sugestão para trabalhos futuros, recomenda-se um estudo mais profundo dos efeitos da árvore de agregação no balanço de energia nos nós da rede. Apesar do filtro PAR manter as características básicas da difusão direcionada, e variar constantemente (a cada fluxo de mensagens exploratórias) as rotas escolhidas, os nós com potencial agregador são intrinsecamente poucos na rede e existe uma tendência de que a sua energia seja depletada mais rapidamente que em outros nós. Dessa forma, a energia remanescente em cada nó poderia ser utilizada como parâmetro adicional para a escolha de rotas.

Além disso, o PAR foi instalado em todos os nós da rede, tanto nos próximos às fontes, como nos próximos ao sorvedouro. Na verdade, a seleção de um nó agregador próximo ao sorvedouro não apresenta muitos ganhos no consumo de energia. Dessa forma, para diminuir a latência na descoberta das rotas e seus efeitos adversos apresentados nesta dissertação, pode-se estudar uma forma de somente invocar o filtro até uma determinada distância das fontes, onde os ganhos são mais altos, e não fazê-lo a partir daí. Dessa forma, possivelmente o tempo de descoberta de rotas seria diminuído com reduzidas perdas no consumo de energia.

O mecanismo proposto foi implementado sobre a difusão direcionada, herdando as suas vantagens e limitações. Essa abordagem pode ser estendida e implementada em outros protocolos centrados em dados. Além disso, a proposta pode ser aplicada em cenários mais complexos, explorando redes onde exista mais de um sorvedouro.

Referências Bibliográficas

[1] Akyildiz, I. F. Wireless sensor networks. Palestra realizada na COPPE/UFRJ, Novembro 2003.

[2] Hof, R. D. The quest for the next big thing. Página da World Wide Web. http://www.businessweek.com/magazine/content/03_34/b3846618.htm, Agosto 2003.

[3] Taylor, C. What dust can tell you. Página da World Wide Web. http://www.dust-inc.com/news/articles/Jan12_2004.htm, Janeiro 2004.

[4] Pottie, G. J., and Kaiser, W. J. Wireless Integrated Network Sensors - wins. *Communications of the ACM* (maio 2000), 51–58.

[5] Intanagonwiwat, C., Govindan, R., and Estrin, D. Directed diffusion: A scalable and robust communication paradigm for sensor networks. *Proceedings of ACM MobiCom'00, Boston, MA* (agosto 2000), 56–67.

[6] Krishnamachari, B., Estrin, D., and Wicker, S. Modeling data-centric routing in wireless sensor networks. *IEEE INFOCOM* (junho 2002).

[7] Heinzelman, W. R., Chandrakasan, A., and Balakrishnan, H. Energy-efficient communication protocol for wireless microsensor networks. *Proceedings of the 33rd IEEE Hawaii International Conference on System Sciences (HICSS)* (janeiro 2000), 1–10.

[8] Lindsey, S., and Raghavendra, C. S. Pegasis: Power efficient gathering in sensor information systems. *International Conference on Communications* (2001).

[9] SHAH, R. C., AND RABAEY, J. Energy aware routing for low energy ad hoc sensor networks. *IEEE Wireless Communications and Networking Conference (WCNC)* (2002).

[10] INTANAGONWIWAT, C., ESTRIN, D., GOVINDAN, R., AND HEIDEMANN, J. Impact of network density on data aggregation in wireless sensor networks. *Proceedings of the 22nd International Conference on Distributed Computing Systems* (julho 2002).

[11] KRISHNAMACHARI, B., ESTRIN, D., AND WICKER, S. The impact of data aggregation in wireless sensor networks. *International Workshop on Distributed Event-Based Systems, (DEBS '02)* (julho 2002).

[12] HEIDEMANN, J., SILVA, F., INTANAGONWIWAT, C., GOVINDAN, R., ESTRIN, D., AND GANESAN, D. Buiding efficient wireless sensor networks with low-level naming. *ACM Symposium on Operating Systems Principles* (outubro 2001).

[13] PAHLAVAN, K., KRISHNAMURTHY, P., AND HATAMI, A. Handoff in hybrid mobile data networks. *IEEE Personall Communications* (abril 2000), 34–36.

[14] INTERNET ENGINEERING TASK FORCE. *IETF Mobile Ad Hoc Networks Working Group*, página da world wide web. http://www.ietf.org/html.charters/manet-charter.html ed., março 2004.

[15] RABAEY, J., AMMER, J., DA SILVA JR., J. L., AND PATEL, D. Picoradio: Ad-hoc wireless networking of ubiquitous low-energy sensor/monitor nodes. *WVLSI* (Abril 2000).

[16] AKYILDIZ, I. F., SU, W., SANKARASUBRAMANIAM, Y., AND CAYIRICI, E. A survey on sensor networks. *Communications of the ACM* (agosto 2002), 102–114.

[17] FENG, J., KOUSHANFAR, F., AND POTKONJAK, M. System-architectures for sensor networks issues, alternatives, and directions. *Proceedings of the 2002 IEEE International Conference on Computer Design* (2002).

[18] Yu, T. *Target Identification Processor for Wireless Sensor Networks.* Phd dissertation, UCLA, 1999.

[19] Rabaey, J. M., Ammer, J., Karalar, T., Li, S., Otis, B., Sheets, M., and Tuan, T. Picoradios for wireless sensor networks - the next challenge in ultra-low power design. *Proceedings of the International Solid-State Circuits Conference* (Fevereiro 2002).

[20] Coleri, S., Ergen, M., and Koo, T. Lifetime analysis of a sensor network with hybrid automata modelling. *Wireless Sensor Networks Applications* (Setembro 2002).

[21] Macedo, D., Correia, L. H., Figueiredo, C. M., Nakamura, E., Ruiz, L., Mini, R., Câmara, D., Maia, E., Loureiro, A. A. F., Nogueira, J. M., Vieira, L. F., Vieira, M. A., and da Silva, D. C. Arquiteturas para redes de sensores sem fio. *Anais do XXII Simpósio Brasileiro de redes de Computadores (p.167-218)* (maio 2004).

[22] Massachussets Institute of Technology. *MIT uAMPS Project*, página da world wide web. http://www.mtl.mit.edu/research/icsystems/uamps/ ed., março 2004.

[23] University of California at Los Angeles UCLA. Electrical Engineering Department. *Wireless Integrated Micro Sensors Project*, página da world wide web. http://www.janet.ucla.edu/wins/ ed., março 2004.

[24] University of California at Berkeley. Berkeley Wireless research Center. *PicoRadio Project*, página da world wide web. http://bwrc.eecs.berkeley.edu/research/picoradio/default.htm ed., março 2004.

[25] University of California at Berkeley. Robotics and Intelligent Machines Laboratory. *The Smart Dust Project*, página da world wide web. http://robotics.eecs.berkeley.edu/ pister/smartdust/ ed., março 2004.

[26] Zhong, L. C., Rabaey, J., Guo, C., and Shah, R. Data link layer design for wireless sensor networks. *Proceedings of MILCOM* (2001).

[27] Hsu, V., Kahn, J. M., and Pister, K. S. J. Wireless communications for smart dust. *Electronics Research Laboratory Technical Memorandum Number M98/2* (fevereiro 1998).

[28] Arisha, K., Youssef, M., and Younis, M. Energy-aware tdma-based mac for sensor networks. *Proceedings of the IEEE Integrated Management of Power Aware Communications, Computing and Networking* (2002).

[29] Willig, A., Shah, R., Rabaey, J., and Wolisz, A. Altruists in the picoradio sensor network. *International Workshop on Factory Communication Systems (WFCS)* (2002).

[30] Ye, W., Heidemann, J., and Estrin, D. Medium access control with coordinated, adaptive sleeping for wireless sensor networks. *USC/ISI TECHNICAL REPORT ISI-TR-567* (janeiro 2003).

[31] Kulik, J., Heinzelman, W. R., and Balakrishman, H. Negotiation-based protocols for disseminating information in wireless sensor networks. *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99)* (1999).

[32] Chandrakasan, A., Min, R., Bhardwaj, M., Cho, S.-H., and Wang, A. Power aware wireless microsensor systems. *ESSCIRC* (Setembro 2002).

[33] Petrovic, D., Shah, R. C., Ramchandran, K., and Rabaey, J. Data funneling: Routing with aggregation and compression for wireless sensor networks. *IEEE Sensor Network Protocols and Applications (SNPA)* (maio 2003).

[34] Garey, M. R., and Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, San Francisco, 1979.

[35] Krishnamachari, B., Estrin, D., and Wicker, S. Modeling data-centric routing in wireless sensor networks. *IEEE INFOCOM* (2002).

[36] Watlz, E., and Llinas, J. *Multisensor Data Fusion.* Artech House, 1990.

[37] Boulis, A., Ganeriwal, S., and Srivastava, M. Aggregation in sensor networks: An energy-accuracy trade-off. *IEEE Sensor Network Protocols and Applications (SNPA)* (maio 2003).

[38] gonçalves Pinto, A. J. Mecanismo de agregação de dados baseado em técnicas paramétricas aplicado em redes de sensores. Dissertação de mestrado, COPPE/UFRJ, Junho 2004.

[39] Heidemann, J., Silva, F., Yu, Y., Estrin, D., and Haldar, P. *Diffusion Filters as a Flexible Architecture for Event Notification in Wireless Sensor Networks.* USC/ISI Technical Report 2002-556, 2002.

[40] Pinto, A. J. G., Stochero, J., and Rezende, J. F. Descoberta de rotas e mecanismo de agregação em redes de sensores. *XXII Simpósio Brasileiro de Redes de Computadores - SBRC'2004* (maio 2004), 567–570.

[41] Stochero, J., Pinto, A. J. G., and Rezende, J. F. Proposta e avaliação de um mecanismo de agregação de dados em redes de sensores. *A ser publicado no III Workshop em Desempenho de Sistemas Computacionais e de Comunicação - WPerformance'2004 (XXIV Congresso da Sociedade Brasileira de Computação - SBC 2004)* (agosto 2004).

[42] Heidemann, J., Silva, F., Intanagonwiwat, C., Govindan, R., Estrin, D., and ganesan, D. Building efficient wireless wensor networks with low-level naming. *Proceedings of the Symposium on Operating Systems Principles, pages 146-159* (2001).

[43] The Network Simulator NS-2. Página da World Wide Web. http://www.isi.edu/nsnam/ns, março 2004.

[44] Massachussets Institute of Technology. *Otcl - object tcl: the faq manual*, página da world wide web. url:ftp://ftp.tns.lcs.mit.edu/pub/otcl/readme.html ed., 2004.

[45] University of South California. Information Sciences Institute. *Scalable Coordination Architectures for Deeply Distributed Systems Project*, página da world wide web. http://www.isi.edu/scadds/ ed., março 2004.

[46] Silva, F., Heidemann, J., and Govindan, R. Network routing application's program interface (api) and walk through 9.0.1. Página da World Wide Web. http://www.isi.edu/scadds/papers/nr_api_9_0_1.pdf, Dezembro 2004.

[47] Hall, D. L. *Mathematical Techniques in Multi-Sensor Data Fusion.* Artech House, 1992.