

JULIANA FRANÇA SANTOS AQUINO

**ALGORITMO DE CODIFICAÇÃO DIFERENCIADA PARA  
REDES DE SENSORES SEM FIO**

Belo Horizonte  
09 de outubro de 2007

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

**ALGORITMO DE CODIFICAÇÃO DIFERENCIADA PARA  
REDES DE SENSORES SEM FIO**

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

JULIANA FRANÇA SANTOS AQUINO

Belo Horizonte  
09 de outubro de 2007



UNIVERSIDADE FEDERAL DE MINAS GERAIS

FOLHA DE APROVAÇÃO

Algoritmo de Codificação Diferenciada para Redes de Sensores Sem Fio

JULIANA FRANÇA SANTOS AQUINO

Dissertação defendida e aprovada pela banca examinadora constituída por:

Ph. D. ANTONIO ALFREDO FERREIRA LOUREIRO – Orientador  
Universidade Federal de Minas Gerais (DCC/UFMG)

D. Sc. EDUARDO FREIRE NAKAMURA – Co-orientador  
Fundação Centro de Análise, Pesquisa e Inovação Tecnológica (FUCAPI)

Ph. D. CLAUDIONOR JOSÉ NUNES COELHO JUNIOR  
Universidade Federal de Minas Gerais (DCC/UFMG)

D. Sc. RAQUEL APARECIDA DE FREITAS MINI  
Pontifícia Universidade Católica de Minas Gerais (DCC/PUC-MG)

Belo Horizonte, 09 de outubro de 2007

# Resumo

Rede de sensores sem fio é um tipo de rede *ad hoc* que pode ser usada com o propósito de monitorar uma variedade de características ambientais, tais como som, temperatura, umidade, pressão, níveis de ruído, dentre outros. Um problema típico dessas redes é como coletar e enviar informações históricas de todos os nós sensores da rede para a estação base. Como a energia disponível desses nós é um recurso crítico, é impraticável transmitir todo o conjunto de dados de cada nó sensor para o nó sorvedouro. Há, portanto, a necessidade de minimizar a comunicação entre os nós sensores e o nó sorvedouro, visto que a comunicação no meio sem fio é o consumidor primário de energia nessas redes. Conseqüentemente, é importante aplicar técnicas para redução dos dados para que menos bits possam ser transmitidos pelo meio sem fio. Neste trabalho, é proposto um algoritmo de codificação diferenciada para redes de sensores sem fio, no qual os nós de sensoriamento enviam apenas as diferenças de suas leituras para uma base comum de dados. Os resultados de simulação mostraram que o algoritmo apresentou um bom desempenho em aplicações onde os nós sensores coletam leituras similares ao longo do tempo.

# Abstract

Wireless sensor network is a type of *ad hoc* network that can be used to monitor a variety of environmental characteristics such as sound, temperature, humidity, pressure and noise levels. A typical problem of these networks is how to collect and send historical information from all sensor nodes to the base station. Due to the resource limitations such as energy, it may be impracticable to transmit a full-resolution data feed from each sensor node. Therefore, it is necessary to minimize the communication among sensor nodes and the base station, since that the communication in wireless media is the primary consumer of energy. So, it is important to apply techniques for the reduction of the data so that the fewest bits can be transmitted in the wireless medium. In this master thesis, a differential coding algorithm for wireless sensor networks was considered, where sensor nodes report just the differences among their readings for a common database. The simulation results have shown that the algorithm presented a good performance in applications where sensor nodes collect similars readings over time.

*À minha família, aos meus padrinhos e ao Henrique.*

# Agradecimentos

A Deus, pelo dom da vida e pela oportunidade de realizar este trabalho.

Aos meus amados pais, Aquino e Efigênia; aos meus irmãos, Jeanne e Jean; e ao meu sobrinho Alan, por me apoiarem e por entenderem a minha ausência. Agradeço pelo amor e por sempre estarem presentes na minha vida. Por eles, sempre busco o melhor. Aos meus padrinhos Gisélia e Roberto, pelo carinho e pelo constante incentivo aos meus estudos.

Ao meu orientador, Professor Antônio Alfredo Ferreira Loureiro, pela pessoa gentil e educada, sempre disposta a ajudar os alunos. Agradeço por tudo: pela paciência, pelas orientações dedicadas e pela compreensão em todos os momentos. Ao meu co-orientador, Professor Eduardo Freire Nakamura, pelas grandes idéias e contribuições ao meu trabalho, pela amizade, pelos ensinamentos e pela injeção de ânimo que eu recebia após conversar com ele. A ajuda dele foi fundamental para a realização deste trabalho. Ao Professor Geraldo Robson Mateus, por ter me acolhido tão bem na UFMG e pela preocupação em me acompanhar desde o início do mestrado. Aos membros da banca pelas relevantes sugestões para melhorar este trabalho.

Ao Professor Marcos José Negreiros Gomes (Gênio), pelo exemplo de professor, pela dedicação aos seus alunos e pelo zeloso apoio e confiança a mim dedicados. A ele, minha eterna gratidão e respeito. Aos Professores Ana Lúiza e Joaquim Celestino, pela oportunidade que me deram de ingressar na área científica.

Aos amigos do Laboratório ATM, pelo companheirismo ao longo do mestrado. Aos amigos que fiz na UFMG e que sempre farão parte da minha vida: Alla, Fê, Lu e Raquel. Obrigada pelos momentos de alegria, pelas dificuldades compartilhadas e pela inabalável disposição em me ajudar. Às secretárias da Pós-Graduação do DCC, pela eficiência e pela disponibilidade em atender bem os alunos.

Aos amigos da UECE, por todo o carinho a mim dedicado. Ao meu bravo amigo César, pelo companheirismo, pelos conselhos, pelas colaborações e por sua sincera amizade. Às minhas queridas amigas Ana Lisse, Diana, Luciana, Manú e Mazé, pelo apoio contínuo, apesar da distância.

Ao meu querido amigo Henrique, pela dedicação e pelo apoio que sempre me foram oferecidos nos momentos mais difíceis. Pela confiança e pela força que me fez seguir em frente. Pelo carinho e por cuidar tão bem de mim. Sem ele, tudo seria mais difícil.

Agradeço ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) pelo apoio financeiro durante o mestrado.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Motivação . . . . .	1
1.2	Objetivo do trabalho . . . . .	2
1.3	Contribuições . . . . .	2
1.4	Estrutura desta dissertação . . . . .	3
<b>2</b>	<b>Fundamentos</b>	<b>4</b>
2.1	Rede de sensores sem fio . . . . .	4
2.1.1	Arquitetura . . . . .	5
2.1.2	Infra-estrutura . . . . .	5
2.1.3	Pilha de protocolos . . . . .	6
2.1.4	Aplicação . . . . .	8
2.2	Compressão de dados . . . . .	8
2.2.1	Tipos de compressão de dados . . . . .	8
2.2.2	Métodos de compressão de dados . . . . .	9
2.2.3	Benefícios da compressão . . . . .	10
2.3	Terminologia . . . . .	11
2.4	Conclusão . . . . .	11
<b>3</b>	<b>Trabalhos Relacionados</b>	<b>12</b>
3.1	Compressão de dados em redes de sensores sem fio . . . . .	12
3.2	Métodos de codificação distribuída . . . . .	13
3.3	Métodos de transformada . . . . .	14
3.4	Métodos de dicionário . . . . .	15
3.5	Métodos de codificação diferenciada . . . . .	15
3.6	Métodos de roteamento direcionado à compressão . . . . .	16
3.7	Outros métodos . . . . .	17
3.8	Conclusão . . . . .	18
<b>4</b>	<b>Algoritmo de Codificação Diferenciada</b>	<b>19</b>
4.1	Motivação . . . . .	19
4.2	Papéis dos elementos da rede . . . . .	20
4.3	Protocolo de comunicação de dados . . . . .	20

4.3.1	Fase de configuração . . . . .	21
4.4	Algoritmo de Codificação Diferenciada . . . . .	25
4.4.1	Fase de formação de base comum . . . . .	25
4.4.2	Fase de comunicação dos dados . . . . .	28
4.4.3	Fase de manutenção . . . . .	29
4.5	Síntese do protocolo . . . . .	31
4.6	Conclusão . . . . .	32
<b>5</b>	<b>Resultados de Simulação</b>	<b>33</b>
5.1	Algoritmos para testes . . . . .	33
5.2	Codificação por ordenação . . . . .	33
5.3	Configuração dos experimentos . . . . .	34
5.4	Testes . . . . .	36
5.4.1	Densidade . . . . .	36
5.4.2	Escalabilidade . . . . .	38
5.4.3	Intervalo de disseminação dos dados . . . . .	40
5.4.4	Variação média das leituras . . . . .	42
5.4.5	Imprecisão das leituras . . . . .	43
5.4.6	Mudança de base . . . . .	45
5.5	Conclusão . . . . .	46
<b>6</b>	<b>Considerações Finais</b>	<b>48</b>
6.1	Conclusões . . . . .	48
6.2	Limitações . . . . .	49
6.3	Trabalhos Futuros . . . . .	49
<b>A</b>	<b>Lista de Siglas e Abreviações</b>	<b>50</b>
	<b>Referências Bibliográficas</b>	<b>51</b>

# Lista de Figuras

2.1	Exemplo de uma rede de sensores sem fio. . . . .	5
2.2	Componentes de um nó sensor (Akyildiz et al., 2002). . . . .	6
2.3	Pilha de protocolos de uma RSSF (Akyildiz et al., 2002). . . . .	7
2.4	Compressão lógica de dados. . . . .	9
2.5	Fluxo de Dados. . . . .	11
4.1	Papéis dos elementos da rede (Petrovic et al., 2003). . . . .	22
4.2	Formato do pacote de interesse. . . . .	22
4.3	Formato do pacote de interesse modificado. . . . .	24
4.4	Base comum de bits para cinco leituras de nós sensores. . . . .	26
4.5	Lista de bases comuns no nó borda. . . . .	26
4.6	Codificação diferenciada das leituras dos nós sensores. . . . .	27
4.7	Bits mais significativos são iguais para valores consecutivos. . . . .	27
4.8	Agregação de pacotes dentro da região alvo. . . . .	28
4.9	Concatenação de leituras. . . . .	29
4.10	Descompressão de leituras. . . . .	30
5.1	Densidade - Energia média consumida pela rede. . . . .	37
5.2	Densidade - Taxa de compressão média dos nós bordas. . . . .	38
5.3	Escalabilidade - Energia média consumida pela rede. . . . .	39
5.4	Escalabilidade - Taxa de compressão média dos nós bordas. . . . .	40
5.5	Energia média consumida de acordo com o intervalo de disseminação dos dados. . . . .	41
5.6	Taxa de compressão média dos nós bordas de acordo com o intervalo de disseminação dos dados. . . . .	41
5.7	Energia média consumida variando-se o desvio-padrão das leituras. . . . .	42
5.8	Taxa de compressão média dos nós bordas variando-se o desvio-padrão das leituras. . . . .	43
5.9	Energia média consumida com <i>bits</i> suprimidos. . . . .	44
5.10	Taxa de compressão média dos nós bordas com <i>bits</i> suprimidos. . . . .	45
5.11	Latência média do nó controlador com <i>bits</i> suprimidos. . . . .	45
5.12	Energia média consumida pela rede reenviando uma base. . . . .	46

# Lista de Tabelas

2.1	Alguns métodos de compressão de dados. . . . .	10
5.1	Mapeamento de permutação de três identificadores para inteiros. . . . .	34
5.2	Parâmetros comuns. . . . .	36
5.3	Parâmetros para o experimento de densidade. . . . .	37
5.4	Parâmetros para a escalabilidade horizontal. . . . .	39
5.5	Parâmetros para disseminação das leituras. . . . .	40
5.6	Parâmetros para o tamanho do pacote de dados. . . . .	42
5.7	Parâmetros para erros. . . . .	44
5.8	Parâmetros para mudança de base. . . . .	46

# Capítulo 1

## Introdução

Este capítulo apresenta a motivação, os objetivos e as contribuições da presente dissertação. A motivação é apresentada na seção 1.1. A seção 1.2 define o problema a ser tratado, bem como esboça a abordagem a ser aplicada ao problema. A seção 1.3 apresenta as contribuições. Finalmente, a seção 1.4 apresenta a forma como este trabalho está organizado.

### 1.1 Motivação

O rápido avanço na tecnologia de fabricação de semicondutores tem permitido o desenvolvimento de dispositivos eletrônicos de mais baixo custo, de menor tamanho físico e que requerem um menor consumo de energia para operarem (Akyildiz et al., 2002). Um dos tipos de redes que se tem beneficiado com o avanço dessa tecnologia na área de microeletrônica e de comunicação sem fio é a rede de sensores sem fio (RSSF). Essas redes são compostas por uma grande quantidade de nós, dotados de capacidade de sensoriamento, de processamento de dados e de disseminação de dados, coletados e processados para um ou mais observadores (Ruiz, 2003). RSSFs podem ser usadas com o propósito de coletar dados em áreas como monitoramento ambiental, monitoramento de pacientes, gerenciamento de desastres e aplicações militares (Akyildiz et al., 2002).

RSSFs possuem limitações de recursos como: energia, largura de banda para comunicação, capacidade de processamento e quantidade de memória. Muitas pesquisas têm sido realizadas para otimizar a utilização dos recursos limitados de nós sensores. Uma vez que o processamento dos dados consome menos energia que a transmissão desses dados em um canal de comunicação sem fio, é interessante aplicar técnicas de compressão de dados lógica, também conhecida como supressão de dados, para reduzir o consumo de energia total por nó sensor. Compressão de dados tem como objetivo eliminar informações redundantes dos dados, de forma a diminuir o seu tamanho.

Desta maneira, apresenta-se, no decorrer desta dissertação, uma abordagem de compressão de dados lógica, baseada em codificação diferenciada (ver capítulo 4), para reduzir o consumo de energia na comunicação dos dados coletados pelos nós sensores, e assim, prolongar o tempo de vida útil dessas redes.

## 1.2 Objetivo do trabalho

As redes de sensores sem fio podem ser usadas em diferentes aplicações (ver seção 2.1.4), porém, há pelo menos três características comuns presentes nas aplicações de redes de sensores que motivaram o presente estudo:

1. Necessidade de minimizar a comunicação entre os nós sensores, desde que a comunicação no meio sem fio seja o consumidor de energia primário nessas redes. Enviar um único bit de dados pode consumir energia capaz de executar um milhão de instruções (Asada et al., 1998).
2. Necessidade de projetar algoritmos simples, dado que os nós sensores têm recursos computacionais bastante limitados (Madden et al., 2002).
3. Os dados sensoriados, comunicados através da rede, têm tanto redundância espacial<sup>1</sup> quanto redundância temporal<sup>2</sup>, sendo ativados pelo mesmo fenômeno de interesse (Krishnamachari et al., 2002, apud Arici et al., 2003).

Em virtude dessas características peculiares dos nós sensores, vários pesquisadores têm proposto diversas técnicas para diminuir o consumo de energia durante a comunicação: agregação de dados (Madden et al., 2002), compressão de dados (Kimura e Latifi, 2005), fusão de dados (Nakamura et al., 2005a), diferentes algoritmos de roteamento (Al-karaki e Kamal, 2004), mapas de energia (Mini et al., 2005) e controle de densidade (Nakamura et al., 2005b; Zhang e Hou, 2005).

Neste trabalho, a técnica escolhida para diminuir o consumo de energia durante a comunicação foi a técnica de compressão de dados lógica baseada em codificação diferenciada. Essa é uma técnica adequada tanto para aplicações que requerem o conteúdo histórico dos nós sensores, como também para ser usada em combinação com outras técnicas (Madden et al., 2002), como por exemplo, agregação de dados, que foi utilizada neste trabalho.

## 1.3 Contribuições

A consequência direta da compressão de dados é uma redução significativa na quantidade de bits a serem transmitidos, permitindo a redução do consumo de energia durante a comunicação dos dados coletados pelos nós sensores. Dentre as principais contribuições do algoritmo de compressão de dados proposto, pode-se citar:

1. O algoritmo de codificação diferenciada permite diferentes taxas de compressão para os diferentes nós fontes da rede de sensores sem fio.

---

<sup>1</sup>Mais de um nó sensoriando um mesmo evento ou uma mesma área de interesse, o que acarreta aumento do número de linhas de comunicação.

<sup>2</sup>Em um intervalo de tempo especificado, um mesmo nó reporta mais de uma leitura sobre uma mesma área, o que aumenta o número de bits transmitidos na comunicação.

2. O algoritmo de codificação diferenciada pode ser combinado com outras técnicas para se obter uma maior redução do consumo de energia durante a comunicação dos dados no meio sem fio.
3. Apresenta bom desempenho em regiões de monitoramento onde as leituras dos nós sensores apresentam padrões similares ao longo do tempo, sendo ideal para aplicações que requerem informações históricas de todos os nós sensores.
4. Os nós sensores que recebem leituras diferenciadas são capazes de restaurar essas leituras para a sua leitura original. Isso permite que qualquer nó sensor possa tomar decisões e agir como um atuador baseado no valor das leituras recebidas.

## 1.4 Estrutura desta dissertação

Este trabalho está organizado em seis capítulos.

O capítulo 2 expõe os conceitos básicos para o entendimento da técnica, cujo conhecimento é necessário para melhor compreensão desta dissertação.

O capítulo 3 apresenta um levantamento de trabalhos que usam diferentes técnicas de compressão de dados em redes de sensores sem fio. Os trabalhos estão organizados seguindo uma taxonomia para apresentar as principais características de cada um.

O capítulo 4 define, em detalhes, o algoritmo de codificação diferenciada de dados proposto.

O capítulo 5 apresenta uma série de experimentos realizados para avaliar o desempenho do algoritmo de codificação diferenciada, usando o simulador de eventos discretos para redes *Network Simulator 2* (NS-2, 2004).

O capítulo 6 apresenta conclusões sobre o estudo desenvolvido nesta dissertação, limitações do algoritmo proposto e sugestões para trabalhos futuros.

## Capítulo 2

# Fundamentos

Este capítulo aborda os conceitos básicos necessários para a compreensão deste trabalho. A seção 2.1 apresenta as redes de sensores sem fio. A seção 2.2 aborda os conceitos gerais sobre compressão de dados. A seção 2.3 apresenta a terminologia que será usada ao longo de todo o texto.

### 2.1 Rede de sensores sem fio

O progresso tecnológico, na comunicação sem fio e na eletrônica, tem permitido o desenvolvimento de dispositivos de pequeno tamanho, de baixo custo, porém com energia limitada, capazes de se comunicar a curtas distâncias. Estes dispositivos são os nós sensores: dispositivos autônomos capazes de sensoriar, de processar e de transmitir dados.

Rede de sensores sem fio (RSSF) é um tipo de rede *ad hoc*<sup>1</sup> que consiste de nós sensores responsáveis por realizar tarefas de sensoriamento, de forma distribuída, em benefício de aplicações clientes. Essas redes funcionam como sistemas de aquisição de dados ambientais. Estes dispositivos podem monitorar fenômenos físicos ou ambientais, tais como: temperatura, som, pressão e vibração. Os nós sensores coletam os dados e encaminham-os para um ponto de saída da rede, chamado nó sorvedouro ou estação-base, para serem analisados e processados, como ilustra a Figura 2.1. Esses dados podem ser enviados por satélite ou através da *Internet* para o gerenciador de tarefas. Os nós sensores responsáveis por coletar os dados são chamados de nós fontes, enquanto que os nós que têm como função rotear os dados são chamados de nós roteadores. Um nó sensor também pode acumular mais de uma função. Os dados recebidos pelo nó sorvedouro podem ser enviados por satélite para o gerenciador de tarefas, o usuário.

Em uma RSSF, a posição dos nós sensores não necessita ser pré-determinada. Isso permite a distribuição não-determinística dos nós sensores em terrenos inacessíveis ou operações de resgate em área de desastre. Desta maneira, os protocolos e os algoritmos de redes de sensores devem possuir capacidade de auto-organização (Figueiredo, 2007). Outra característica de uma rede de sensores sem fio é o esforço de cooperação dos nós sensores, que podem processar

---

<sup>1</sup>Rede na qual os dispositivos comunicam-se diretamente uns com os outros sem a intervenção de pontos centrais.

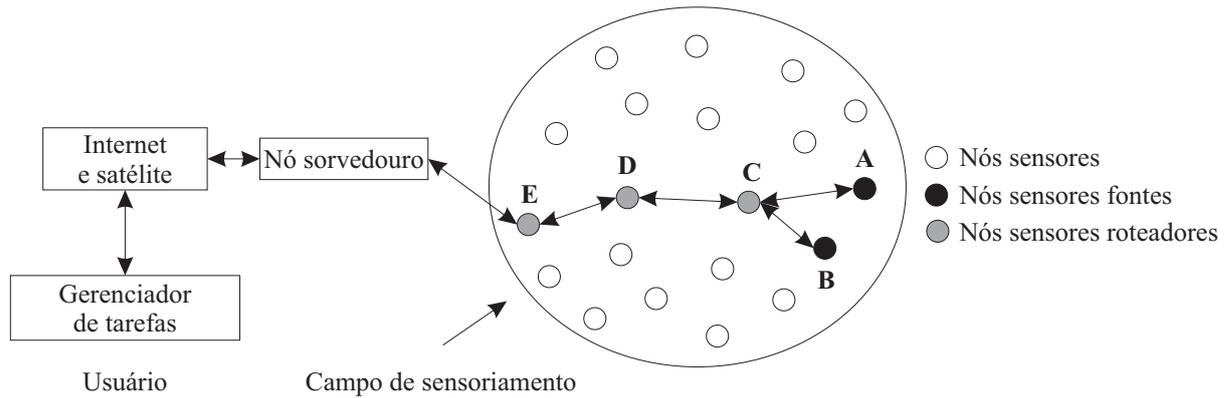


Figura 2.1: Exemplo de uma rede de sensores sem fio.

os dados localmente, realizar simples computações e transmitir somente os dados requisitados e parcialmente processados.

### 2.1.1 Arquitetura

Os três principais componentes da arquitetura de uma RSSF são a infra-estrutura, a pilha de protocolos e a aplicação. Como a economia de energia é um requisito essencial em RSSFs, técnicas devem ser adotadas nos diferentes componentes da rede visando a otimização dos recursos de energia.

Nas seções seguintes, apresenta-se cada um desses componentes.

#### 2.1.2 Infra-estrutura

A infra-estrutura de uma RSSF consiste de nós sensores (suas características físicas e suas capacidades) e de sua estratégia de instalação no ambiente (como e onde eles são instalados). Em geral, uma RSSF possui vários nós sensores e um ou mais nós sorvedouros. Os sorvedouros, em geral, são nós com maior poder computacional e sem restrições de energia.

O *hardware* de um nó sensor típico é composto por quatro unidades principais: a de sensoriamento, a de processamento, a de comunicação e a de energia (ver Figura 2.2). Dependendo da aplicação, podem existir outras unidades adicionais, por exemplo, a de localização (para determinar com precisão a posição de um nó) e a de movimento (para mover o nó para um local que permita a realização de uma tarefa).

A unidade de sensoriamento é usualmente composta por sensores e pelo conversor de sinal analógico para digital (ADC). Os sinais analógicos, produzidos pelos sensores a partir do fenômeno monitorado, são convertidos em sinais digitais pelo ADC e são passados para o processador. O subsistema de processamento, que geralmente está associado a uma pequena unidade de armazenamento, é responsável pela execução dos protocolos de comunicação, pelo processamento dos dados, pelo controle dos sensores e pela gerência dos procedimentos necessários para que os nós sensores atuem de forma colaborativa. O subsistema de comunicação

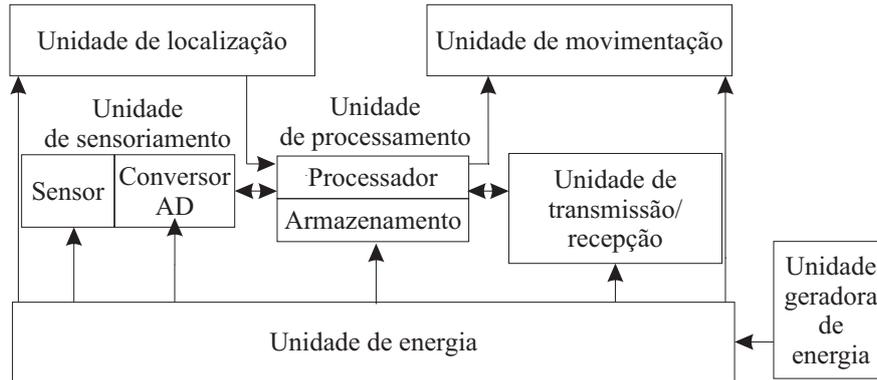


Figura 2.2: Componentes de um nó sensor (Akyildiz et al., 2002).

baseado em rádio-freqüência é composto por um transceptor, uma antena e um conjunto de componentes, para configurar as características da camada física, como sensibilidade e intensidade do canal (Akyildiz et al., 2002). O subsistema de energia é composto por uma bateria e por um conversor DC-DC<sup>2</sup>, responsável por fornecer uma tensão constante para o sensor. O fator de eficiência do conversor DC-DC é um dos elementos que determina o tempo de vida da bateria.

Além dos componentes de *hardware*, o projeto de um nó sensor pode incluir um núcleo de um sistema operacional simples. Quando presente, o sistema operacional (SO) deve gerenciar a operação de um nó sensor da forma mais eficiente possível. Exemplos de SOs para nós sensores são: TinyOS (Hill et al., 2000), MANTIS (Bhatti et al., 2005), SOS (Han et al., 2005) e Contiki (Dunkels et al., 2004).

### 2.1.3 Pilha de protocolos

A pilha de protocolos refere-se ao *software* que implementa as diversas camadas de protocolos existentes em cada nó da rede (nó sensor e nó sorvedouro). O conjunto de protocolos de um nó pode ser observado na Figura 2.3. A pilha de protocolos consiste de cinco camadas horizontais e de três planos verticais (Akyildiz et al., 2002).

As camadas horizontais são:

1. Aplicação: Refere-se a protocolos que são de uso comum para diferentes tipos de aplicação. Dependendo da tarefa de sensoriamento, diferentes tipos de *software* de aplicação podem ser construídos e usados para interagir com a RSSF. O componente de aplicação emite um conjunto de consultas, que descrevem as características dos fenômenos que o usuário deseja analisar. Essas consultas são traduzidas em tarefas de sensoriamento a serem executadas pela rede. A aplicação deve indicar os tipos de dados desejados, a necessidade dos dados serem processados, a freqüência com que esses dados devem ser coletados ou eventos que podem disparar algum comportamento particular da rede.

<sup>2</sup>Em eletrônica, um conversor DC-DC é um dispositivo que converte uma tensão de corrente contínua de entrada em uma tensão diferente de corrente contínua como saída.

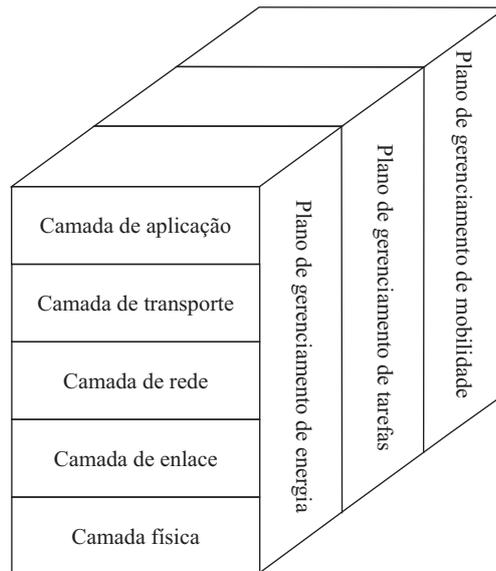


Figura 2.3: Pilha de protocolos de uma RSSF (Akyildiz et al., 2002).

2. Transporte: Responsável por manter o fluxo de dados entre a origem e o destino, se a aplicação assim necessitar. Exemplos de protocolo de transporte em RSSF podem ser vistos em Wan et al. (2002) e Sankarasubramaniam et al. (2003).
3. Rede: Trata do roteamento dos dados. Alguns dos principais protocolos de roteamento podem ser encontrados em Al-karaki e Kamal (2004).
4. Enlace de dados: Assegura uma conexão confiável em uma rede, através da multiplexação dos fluxos de dados, detecção dos quadros, acesso ao meio e controle de erros. O trabalho de Woo e Culler (2001) implementa um protocolo para essa camada.
5. Física: Abrange as técnicas de transmissão, de recepção e de modulação usadas na rede. A comunicação entre os nós (transmissão e recepção) é responsável pela maior parte do consumo de energia em RSSFs, portanto, a eficiência em termos de gasto de energia é importante no projeto da camada física.

Os planos verticais são planos de gerenciamento necessários para que os nós sensores possam trabalhar juntos de forma eficiente, em termos de energia, roteando dados em redes com sensores móveis e compartilhando recursos entre si. São os planos de:

1. Gerenciamento de energia: Permite gerenciar a forma como os nós sensores utilizam a sua energia. Um nó sensor pode desligar seu receptor ao receber uma mensagem de seus vizinhos, evitando a recepção de mensagens duplicadas. Um sensor com baixo nível de energia pode decidir não participar do roteamento de mensagens, indicando aos vizinhos a sua decisão. Todas essas decisões são de responsabilidade do plano de gerenciamento de energia.

2. Gerenciamento de tarefas: Permite balancear e gerenciar as tarefas de sensoriamento de uma região específica.
3. Gerenciamento de mobilidade: Permite detectar e registrar a disponibilidade de movimento dos nós sensores, de modo a sempre manter uma rota de cada nó para o nó sorvedouro. Esse plano não é comum nas implementações existentes dos nós sensores.

#### 2.1.4 Aplicação

As RSSFs podem ser empregadas em diversas áreas, como as descritas a seguir:

- Aplicações ambientais: Podem ser usadas para monitorar variáveis ambientais em prédios, em florestas, em oceanos. Podem rastrear o movimento de animais, monitorar as condições ambientais que afetam a colheita e o gado, o controle de irrigação e a agricultura de precisão.
- Aplicações domésticas: Nós sensores e atuadores inteligentes podem ser embutidos em eletrodomésticos. Os usuários podem gerenciar dispositivos domésticos remotamente de forma fácil e integrada (Garlan et al., 2002).
- Aplicações de saúde: As RSSFs podem ser usadas na monitoração de pacientes, na realização de diagnósticos, na administração de drogas em hospitais e no telemonitoramento de dados fisiológicos humanos (Noury et al., 2000).
- Aplicações comerciais: Podem ser usadas no gerenciamento de estoque de fábricas, monitoramento da qualidade de produtos, controle de robôs, brinquedos e museus interativos e detecção de roubos de automóveis (Priyantha et al., 2000; Rabaey et al., 2000).
- Aplicações militares: Devido à rápida instalação, a auto-organização e a tolerância a falhas, as RSSFs são de grande utilidade para sistemas militares de comando, de controle, de comunicação, de inteligência, de vigilância e de reconhecimento (Akyildiz et al., 2002).

## 2.2 Compressão de dados

É comum associar a compressão de dados à economia de espaço que, na realidade, é um dos seus principais objetivos, todavia esta abordagem deve ir além. O escopo da compressão de dados não abrange apenas dados armazenados, mas incorpora também a agilização da transmissão dos dados através de uma rede. As seções 2.2.1, 2.2.2 e 2.2.3 apresentam os tipos, os métodos e os benefícios da técnica de compressão de dados, respectivamente.

### 2.2.1 Tipos de compressão de dados

As duas técnicas apresentadas a seguir podem resultar em uma representação de dados codificada de forma mais eficiente. São elas: compressão lógica e compressão física.

### 2.2.1.1 Compressão lógica

Esta técnica refere-se ao projeto de representação otimizada de dados. Por exemplo, quando um banco de dados é projetado, um dos primeiros passos da análise é obter toda redução possível. Esta redução de dados é oriunda da eliminação de campos redundantes de informação que pode ser representada por poucos indicadores lógicos. Ao invés de seqüências de caracteres ou de inteiros, utilizam-se bits, reduzindo significativamente o espaço de utilização do banco de dados. Este tipo de compressão é possível de ser efetivado em campos projetados para representar dados constantes, como datas e códigos (ver Figura 2.4).

Embora a compressão lógica possa ser uma ferramenta efetiva na minimização de um banco de dados, ela somente reduz o tempo de transmissão quando é empregada também durante a transmissão desses dados. Similarmente, a ocorrência de padrões e de grupos de caracteres repetidos não é afetada pela compressão lógica. Para essas situações, uma redução no tempo de transmissão depende também de uma compressão física aplicada sobre esses dados (Held e Marshall, 1996).

	DIA	MÊS	ANO
Extensão	1	Junho	1981
Compressão lógica usando representação numérica	01	06	81
Compressão lógica usando representação binária	00001	0100	1010001

Figura 2.4: Compressão lógica de dados.

### 2.2.1.2 Compressão física

A compressão física pode ser vista como um processo de redução da quantidade de dados antes desses dados serem enviados por um meio de transmissão, com a garantia de sua expansão para o formato original, após a recepção desses dados comprimidos no destino. Essa técnica é realizada sobre dados existentes, a partir dos quais é verificada a repetição de caracteres para efetivar a redução do número de elementos de dados através de uma representação otimizada de caracteres ou de grupos de caracteres. Caracteres de maior freqüência de utilização são representados por códigos binários pequenos, e os de menor freqüência são representados por códigos proporcionalmente maiores. Não é necessário saber qual caractere vai ser comprimido, mas sim conhecer a probabilidade de ocorrência de todos os caracteres sujeitos à compressão. Caso não seja possível a tabulação de todos os caracteres sujeitos à compressão, utiliza-se uma técnica para levantamento estatístico dos dados a comprimir, formando tabelas de probabilidades.

## 2.2.2 Métodos de compressão de dados

Em geral as técnicas de compressão de dados classificam-se em: compressão de dados sem perdas e com perdas.

Quando a descompressão de um fluxo de dados comprimido resulta na exata reconstrução do fluxo de dados original, tem-se a categoria de compressão de dados sem perdas, também denominada compressão reversível ou não-destrutiva.

Como exemplos de métodos de compressão reversível, tem-se: codificação *run-length*, codificação de dicionário (LZ77, LZ78, LZW), transformada Burrows-Wheeler e codificação de entropia (Huffman, Aritmética). Para citar exemplos práticos de implementação de compressão sem perdas tem-se o DEFLATE (combinação de LZ77 e codificação de Huffman), usado pelos arquivos zip, gzip e png; bzip2 (combinação da transformada de Burrows-Wheeler e codificação Huffman).

Técnicas de compressão com perdas de dados podem ou não ser totalmente revertidas a seus dados originais e são primariamente restritas a operações sobre imagens, vídeo e áudio. Embora o resultado da descompressão não deva fornecer uma duplicação exata dos dados originais, as diferenças entre os dados originais e os dados reconstruídos são mínimas (Held e Marshall, 1996).

Transformada de cosseno discreta (TCD), compressão fractal, compressão de *wavelet* e codificação preditiva linear (CPL) são exemplos de métodos de compressão com perdas. Exemplos de implementação desses métodos são: MPEG (transformação de cossenos discretos), JPEG (transformação de cossenos discretos e codificação de Huffman) e JPEG 2000 (compressão de *wavelet*).

Um resumo dos principais métodos de compressão de dados pode ser visto na Tabela 2.1.

Compressão de dados		
Sem perdas	Codificação de entropia	Aritmética (Shannon-Fano), Huffman, Huffman Adaptativo
	Codificação de dicionário	LZ77/78, LZW, LZO, DEFLATE
Com perdas	Áudio	Transformada de Fourier, TCD Modificado, CPL
	Imagem	Fractal, Wavelet, TCD
	Vídeo	Quantização, TCD

Tabela 2.1: Alguns métodos de compressão de dados.

### 2.2.3 Benefícios da compressão

Ao aplicar compressão sobre os dados, tem-se a economia de espaço em dispositivos de armazenamento, como discos rígidos. Com respeito à transmissão de dados, a compressão pode reduzir a probabilidade de colisões e, conseqüentemente, a probabilidade de erros durante a transmissão. A transferência de dados comprimidos sobre o meio resulta em um aumento da taxa efetiva de informação transferida, conseqüentemente, tem-se um ganho de desempenho nas transmissões.

## 2.3 Terminologia

Como mostrado na Figura 2.5, um fluxo de dados é processado por um algoritmo para produzir um fluxo comprimido. Essa compressão é referenciada como um processo de codificação, cujo resultado é chamado de fluxo de dados codificado ou comprimido. Desde que o processo de descompressão resulta na decodificação do fluxo de dados comprimido, o resultado é referenciado como fluxo de dados decodificado ou descomprimido. Os termos fluxos de dados codificados/comprimidos serão usados ao longo desta dissertação sem distinção, bem como fluxos de dados decodificados/descomprimidos.

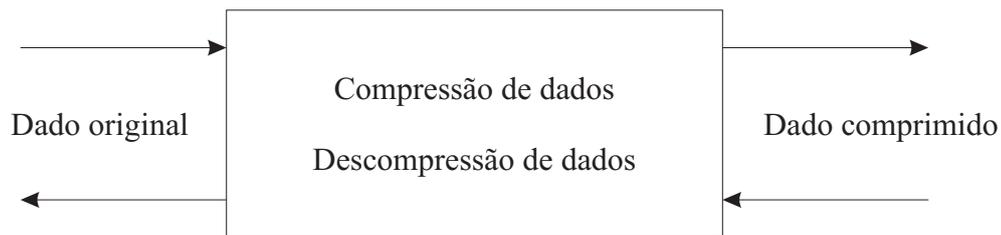


Figura 2.5: Fluxo de Dados.

O grau de redução de dados obtido como resultado de um processo de compressão é conhecido como taxa de compressão de dados. Esta taxa é usada para determinar a quantidade de dados comprimidos em relação à quantidade de dados original (ver equação 2.1).

$$\text{Taxa de compressão} = \frac{\text{Tamanho do dado comprimido}}{\text{Tamanho do dado original}} \quad (2.1)$$

## 2.4 Conclusão

Este capítulo explanou os conceitos básicos sobre redes de sensores sem fio e apresentou os principais componentes da arquitetura de RSSF: infra-estrutura, pilha de protocolos e aplicação. Fez ainda uma explanação sobre compressão de dados, bem como os seus tipos: compressão lógica e compressão física. Foram expostos os benefícios da compressão de dados durante a comunicação desses dados no meio de transmissão sem fio. Dividiram-se os métodos de compressão sem fio em dois tipos: métodos de compressão de dados sem perdas, também chamado de reversível ou não-destrutivo, e métodos de compressão de dados com perdas. Finalmente, apresentou-se a terminologia adotada ao longo desta dissertação.

## Capítulo 3

# Trabalhos Relacionados

Tendo em vista os fundamentos apresentados no capítulo 2, o presente capítulo tem como objetivo explicar diversos trabalhos na literatura que utilizam diferentes técnicas de compressão de dados em redes de sensores sem fio. A seção 3.1 expõe os trabalhos divididos segundo uma taxonomia, que será adotada para classificar os trabalhos ao longo de toda a dissertação. As seções seguintes apresentam em detalhes cada método identificado na seção 3.1.

### 3.1 Compressão de dados em redes de sensores sem fio

Atualmente, maneiras de comunicação, de processamento, de sensoriamento e de computação estão se afastando cada vez mais das arquiteturas tradicionais centralizadas. A tendência é afastar-se de uma plataforma centralizada e super-confiável para uma maneira distribuída de componentes baratos, de pequenos tamanhos e individualmente não-confiáveis que, como um grupo, são capazes de realizar inferências e tarefas mais completas do que um super-nó (Pradhan et al., 2002).

Um exemplo clássico desse comportamento distribuído é o sensoriamento distribuído, onde é desejável ter alta densidade de sensores para garantir confiabilidade, acurácia e baixo custo de desenvolvimento.

Muitos cenários envolvendo rede de sensores sem fio necessitam ter alta densidade espacial para assegurar operações confiáveis em face das falhas dos nós sensores, bem como para facilitar a alta localização espacial de eventos de interesse. Isto induz um alto nível de redundância de dados na rede, onde as leituras de sensores com proximidade espacial estão altamente correlacionadas.

A necessidade de retirar esta redundância é motivada por vários fatores, tais como:

- Há somente um canal de rádio disponível para os nós sensores se comunicarem, tornando a utilização eficiente da banda de comunicação um fator crítico. Uma vez que a compressão de dados possa ser aplicada às leituras coletadas pelos nós de sensoriamento, reduz-se a probabilidade de colisões, pois, menos dados estarão trafegando pelo meio sem fio durante menos tempo, reduzindo-se, conseqüentemente, a probabilidade de erros durante a transmissão e a retransmissão dos dados.

- Em uma rede de múltiplos saltos, os benefícios da compressão de dados são ampliados à medida que se obtém economia de energia em cada transmissão e recepção ao longo da rota até o destino final (nó sorvedouro).

Os trabalhos sobre compressão de dados em redes de sensores sem fio serão classificados segundo a taxonomia apresentada abaixo:

1. Métodos de codificação distribuída
2. Métodos de transformada
3. Métodos de dicionário
4. Métodos de codificação diferenciada

Nas seções seguintes, apresenta-se a definição de cada um dos métodos apresentados.

### 3.2 Métodos de codificação distribuída

Os métodos de compressão tradicionais tratam cada origem como uma identidade independente, enquanto que em redes de sensores reais e, particularmente, redes de sensores densas que realizam monitoramento ambiental, as leituras dos nós sensores tendem a ser altamente correlacionadas (Petrovic e Ramachandran, 2003).

Após os resultados publicados por Slepian e Wolf (1973) e por Wyner e Ziv (1976), houve um maior interesse sobre técnicas de codificação distribuída. Essa técnica envolve compressão separada de múltiplas origens estatisticamente correlacionadas e descompressão junta em um destino comum (como uma estação base). A principal vantagem de processamento de sinal colaborativo e distribuído é que comprimindo a redundância dentro de cada origem, elimina-se a correlação entre as origens de maneira distribuída sem o custo caro e explícito da comunicação entre os nós sensores. A codificação distribuída de origem adiciona uma vantagem às técnicas de conservação de energia existentes, podendo ser usada em combinação com outras técnicas para uma conservação máxima de energia.

No trabalho apresentado por Pradhan et al. (2002), também foi proposta uma forma de remover a redundância de uma maneira completamente distribuída, sem os nós sensores comunicarem-se uns com os outros. Isso é alcançado através de um *framework* construtivo chamado DISCUS (*Distributed Source Coding Using Syndromes*), inspirado pelos conceitos da Teoria de Informação. Em DISCUS, ao invés de se transmitir a palavra-código representando  $X$ , a síndrome do *coset* da palavra-código é enviada e o receptor o decodifica pela escolha da palavra código no dado *coset* que é mais próximo de  $Y$ . DISCUS foi usado para codificar origens gaussianas correlacionadas, entretanto, esse *framework* não mostra claramente que os seus limites teóricos aproximam-se do trabalho de Slepian e Wolf (1973).

Petrovic e Ramachandran (2003) propuseram um *framework* para explorar correlações existentes baseadas em processamento de sinal adaptativo, seguindo os princípios de codificação de origem distribuída. Um filtro adaptativo é usado para aprender as estruturas de

correlação de dados relevantes nas leituras dos sensores. Os nós de sensoriamento, entretanto, não precisam conhecer a estrutura de correlação. Eles precisam conhecer somente o número de bits que deverão usar para codificar as suas medidas. O *framework* propõe uma maneira de projetar um codificador barato, em termos de facilidade de implementação, que pode suportar múltiplas taxas de compressão.

O trabalho de Alqamzi e Li (2005) investigaram como compressão distribuída e adaptativa pode ser eficientemente integrada com uma arquitetura de redes de sensores baseada em agrupamentos, em especial, com o protocolo de roteamento amplamente difundido, o LEACH (Heinzelman et al., 2000). Nessa proposta, é usado um esquema adaptativo baseado em códigos convolucionais perfurados, onde se mostrou uma redução de 20 a 41% comparado à compressão convencional sem modificar o protocolo de roteamento LEACH. Para maximizar os benefícios da codificação de origem distribuída, um esquema adaptativo foi proposto, que permite aos nós sensores ajustarem-se, continuamente, para o tempo de variação da correlação da origem e comprimir a uma taxa correta. A dissipação de energia foi comparada usando o LEACH com e sem codificação de origem distribuída.

Hua e Chen (2005) propuseram um algoritmo de compressão de dados distribuído baseado em códigos convolucionais e códigos turbo. O papel deste último é tentar recuperar a informação perdida no processo de compressão. O código de controle de erro tem a habilidade de corrigir sinais corrompidos transmitidos pelo ruído do canal. Como principais vantagens, tem-se a diminuição da probabilidade de decodificação com erro e da complexidade computacional.

### 3.3 Métodos de transformada

Os métodos de transformada baseiam-se na análise espectral do sinal para aplicar a transformada de Fourier, de Laplace e de *wavelets*. Para os algoritmos de *wavelets* distribuídos, a quantidade de troca de informações necessárias e o grau de correlação alcançados estão diretamente relacionados ao número de níveis de decomposição usados. Dependendo das propriedades da rede, como correlação de dados e distância entre os nós sensores, um número específico de decomposições se deve mostrar mais adequado para minimizar o custo da transmissão. São eliminadas transmissões desnecessárias pelo cálculo das aproximações parciais dos coeficientes de *wavelets*, baseados nos dados disponíveis em cada nó sensor.

Em Ciancio e Ortega (2005) foi proposto um algoritmo de compressão distribuído para redes de sensores com múltiplos saltos baseado na fatorização *lifting* da transformada de *wavelet*. Esta fatorização explora o fluxo natural de dados na rede para agregar dados pela computação parcial dos coeficientes de *wavelet*, que são refinados como fluxos de dados em direção ao nó central. O resultado desse trabalho é que a realização das computações parciais reduz transmissões desnecessárias, otimizando significativamente o consumo de energia. O trabalho resolve o problema de compressão em rede de sensores sem fio, tendo como objetivo reconstruir a versão do campo amostrado para o nó central consumindo o mínimo de energia possível.

### 3.4 Métodos de dicionário

O método de dicionário baseia-se no fato de que as informações históricas coletadas pelos nós sensores exibem um padrão similar ao longo do tempo. Sendo assim, um *codebook* (livro de código) pode ser criado pelos nós sensores de acordo com os dados coletados e enviado para a estação base. Os nós sensores continuam a coletar dados. Após um período de tempo coletando dados, o nó sensor computa a atualização do seu *codebook*. Com o *codebook* atualizado, os nós sensores utilizam-o para comprimir os dados coletados. Em seguida, os nós sensores transmitem a atualização do *codebook* e os dados comprimidos. O livro de código enviado é usado pelo nó sorvedouro para realizar a descompressão dos dados comprimidos.

Baseados no fato de que as informações coletadas por nós sensores exibem padrões similares ao longo do tempo, Lin et al. (2005) propuseram o algoritmo LVQ (*Learning Vector Quantization*) para construir o *codebook*, aliado à aplicação da regressão de dois níveis para aumentar a compressão do *codebook*. Sabendo-se que em RSSFs, os nós sensores, usualmente, coletam diferentes dados, a qualidade da compressão de diferentes sensores é usualmente diferente, se for dada uma mesma banda para transmissão. O método apresenta ainda um algoritmo de alocação de banda dinâmica para balancear a qualidade de compressão de diferentes nós sensores. A técnica tenta utilizar eficientemente a banda de transmissão, onde os resultados mostram uma melhora na precisão da compressão.

### 3.5 Métodos de codificação diferenciada

Métodos de codificação diferenciada usam as leituras enviadas anteriormente para codificarem novas leituras baseadas na diferença entre elas ou baseadas na diferença de uma base comum definida por um nó central.

Em Ju e Cui (2005), foi proposto um mecanismo de compressão de pacotes, chamado de EasiPC, como uma nova técnica de gerenciamento de energia para redes de sensores sem fio. Com este mecanismo, os tamanhos dos pacotes de dados são diminuídos significativamente devido à redução efetiva de informação redundante. A técnica de codificação diferenciada não é aplicada somente às leituras dos nós sensores, mas também ao cabeçalho dos pacotes. Campos de cabeçalho que não se modificam ou que se modificam raramente durante um fluxo de dados só são enviados no início do fluxo. Os nós sensores intermediários mantêm informações de contexto sobre esses campos, bem como sobre os campos cujos valores são codificados através de diferença. Ao transmitir novos pacotes, os valores dos campos de cabeçalho e de dados são enviados baseados na diferença dos valores enviados anteriormente. Como consequência, a energia requerida para transportar pacotes é economizada também, bem como a redução de atraso das transmissões e a redução de colisões para acessar o canal. Testes realizados em aplicações reais usando o TinyOS (Hill et al., 2000) mostraram a viabilidade do trabalho proposto, porém, como a computação das diferenças é realizada sempre em relação às últimas informações enviadas, a perda de pacotes intermediários pode comprometer a correta codificação dos pacotes subsequentes.

Hoang e Motani (2005) propuseram uma nova abordagem que explora a característica natural de *broadcast* do canal sem fio para realizar compressão de dados. Em uma rede de sensores baseada em *clusters*, quando um nó particular envia seus dados para o *cluster-head*, os outros sensores podem receber esses dados e utilizá-los para comprimir os seus próprios dados através de codificação diferenciada. Foi proposto um problema de otimização, como um problema de programação linear inteira, em que os nós sensores em cada *cluster* colaboram entre si transmitindo, recebendo e comprimindo dados para maximizar o tempo de vida útil da rede de sensores sem fio. Uma heurística foi apresentada para solucionar esse problema com um custo próximo do valor ótimo.

### 3.6 Métodos de roteamento direcionado à compressão

Os métodos de roteamento direcionado à compressão são métodos que roteiam os pacotes de forma a favorecer a compressão de dados nos pontos de agregação. À medida que as leituras passam por mais pontos de agregação, uma maior compressão será efetuada, diminuindo a quantidade de mensagens a serem enviadas ao nó sorvedouro.

O protocolo de roteamento proposto por Petrovic et al. (2003), *Data Funneling*, considera o problema de minimizar a quantidade de comunicação necessária para enviar leituras de um grupo de nós sensores para um único destino, chamado de nó controlador, usando compressão de dados ao longo do roteamento. O algoritmo também usa a técnica de agregação de dados, desde que permite aos nós sensores comprimirem os seus dados com dados de nós vizinhos antes de enviá-los para o nó controlador.

Considerando-se que os cabeçalhos dos pacotes têm tamanho elevado em relação ao tamanho do pacote total e que a quantidade de dados em cada leitura é baixa, grande economia de energia pode ser alcançada se diferentes leituras de nós sensores puderem ser combinadas em um único pacote, reduzindo-se também a probabilidade de colisões. Assim, um único nó por *round* que está na borda da região a ser monitorada realiza também a compressão das leituras recebidas de todos os nós sensores da sua região, mesmo leituras já comprimidas, unindo todos os dados dessas leituras em um super-pacote. Esse método simples é denominado *coding by ordering*.

No *coding by ordering*, a principal idéia é que quando se transmite muitos pedaços de dados e a ordem desses pedaços de dados não é importante para a aplicação, isto é, o transmissor é quem escolhe a ordem em que esses pedaços de dados devem ser enviados, então a escolha da ordem em que esses pedaços de dados deverão ser enviados pode ser usada para conduzir informações adicionais ao receptor. A quantidade de pacotes que pode ser suprimida depende apenas do intervalo de valores em que os dados podem ser gerados, mas não depende do valor do dado gerado no pacote. Assim, muitas leituras com dados iguais podem ser enviadas, sem que nenhum método de compressão mais eficiente seja usado. Mais detalhes sobre o *Data Funneling* são descritos no capítulo 4.

Beyens et al. (2005) apresentam uma variação do algoritmo Q-learning (Watkins e Dayan, 1992), chamado Q-RC (*Q-Routing with Compression*), um algoritmo que aprende a melhor

estratégia de roteamento com compressão. Esse trabalho utiliza técnicas de agregação e de compressão de dados. A principal contribuição é que o algoritmo Q-RC escolhe a melhor estratégia de roteamento dependendo dos dados correlacionados, ou seja, os nós sensores aprendem como rotear os dados para o nó sorvedouro quando compressão de dados é realizada nos nós sensores intermediários. Esse trabalho considera a rede de sensores sem fio como um sistema multi-agente, em que os agentes são os nós sensores e as ações são os encaminhamentos de pacotes. Baseado no *Q-learning*, cada agente tenta aprender quais são as melhores ações a serem tomadas. Os resultados mostraram que o Q-RC pode adaptar sua estratégia de roteamento dependendo dos dados correlacionados sem uso excessivo de pacotes de *feedback* e com uma velocidade de convergência aceitável.

### 3.7 Outros métodos

Em Scaglione e Servetto (2005), foi considerado o problema de comunicação *broadcast* em uma rede de sensores de múltiplos saltos, em que amostras de um campo aleatório são coletadas em cada nó da rede, onde o objetivo é que todos os nós obtenham uma estimativa do campo de entrada dentro de um valor de distorção prescrito. A principal idéia explorada neste trabalho é que a junção de compressão de dados gerados por diferentes nós e de como esta informação trafega através de múltiplos saltos é necessária para eliminar correlações na representação de um campo amostrado. Como resultado, a proposta construiu uma grande classe de modelos estocásticos motivados fisicamente para os dados dos nós sensores.

Em Vasudevan e Ortega (2002) foi investigado o problema de compressão de sinais aplicada ao problema de localização em redes de sensores sem fio, ou seja, os sinais são processados para o propósito de localização da origem do evento. Nesse cenário, compressão deve ser otimizada para prover a exatidão da localização da origem, ao invés de se utilizar reprodução de sinais com desejável fidelidade. O artigo apresenta técnicas de projeto que apresentam vantagens sobre as abordagens de projeto de quantizadores padrões. Os testes das novas abordagens são realizados em uma aplicação onde o objetivo é otimizar a exatidão da estimativa de localização.

O trabalho proposto por Gao (2002) mostrou um algoritmo adaptativo de roteamento usado em processamento de sinal cooperativo baseado em evento, que destaca duas técnicas: combinação coerente e não-coerente. O núcleo do algoritmo é um procedimento de eleição distribuída que produz um ou mais vencedores baseados na métrica de eleição dependente do contexto. Por exemplo, em cenários onde técnicas de processamento de sinal não-coerente são aplicadas, um nó de processamento central é selecionado para processar os dados comprimidos dos outros nós sensores, o que pode reduzir significativamente o custo de comunicação. Os termos coerente e não-coerente estão relacionados com o grau em que as informações temporais são removidas dos dados.

O trabalho proposto por Maniezzo et al. (2002) tratou de uma heurística visando minimizar o consumo de energia em redes de sensores sem fio para aplicações multimídia, usando processadores *Strong Arm RISC* para realizar compressão de dados. Para esse propósito, foi

considerado um esquema de compressão de taxa de bit variável. O trabalho ainda apresenta um modelo analítico para estudar o consumo de energia ótimo na rede de sensores multimídia. O modelo mostra que existe um número ótimo de nós envolvidos na estratégia de compressão de dados que minimiza o consumo total de energia. São analisados o *tradeoff* de energia entre os recursos de computação e de comunicação, em uma rede de sensores sem fio, para minimizar o consumo total de energia.

O trabalho proposto por Pattem et al. (2004) visa a analisar o desempenho de roteamento com compressão em rede de sensores sem fio usando uma medida de compressão de dados para quantificar o tamanho da informação comprimida e a métrica *bit-hop* para quantificar o custo total de unir roteamento e compressão. Sabendo-se que a natureza de roteamento ótimo com compressão depende do nível de correlação, existe um esquema de clusterização que pode fornecer um desempenho próximo ao ótimo para uma grande variedade de intervalos de correlações espaciais. O trabalho mostra ainda que o tamanho ótimo de um *cluster* depende do número de origens, da posição do *sink* e da quantidade de correlação entre as origens.

### 3.8 Conclusão

Este capítulo apresentou diversos trabalhos de compressão de dados para redes de sensores sem fio. Tentou-se adotar uma taxonomia para dividir os trabalhos relacionados focando nas principais características de cada um. Verificou-se que algumas das técnicas apresentadas podem ser usadas em conjunto para reduzir a perda de energia durante a transmissão dos dados no meio de comunicação sem fio. Em particular, o método de codificação de origem distribuída pode ser utilizado como um modelo ideal para eliminar a redundância das leituras coletadas pelos nós sensores, tendo em vista que a redundância é eliminada ainda na fonte, porém, as leituras dos nós sensores estão correlacionadas estatisticamente e não leva em consideração a variação das leituras em um ambiente real, o que pode fugir a uma correlação estatística. A redução da energia decorrente da aplicação da técnica de codificação de origem distribuída é o limite inferior para os outros métodos de compressão de dados. O método de dicionário também leva em consideração o padrão que as leituras dos nós exibem ao longo do tempo para a construção de livros de códigos. Já os métodos de codificação diferenciada aqui expostos e os trabalhos que focam no roteamento a fim de favorecer a compressão dos dados nos pontos de agregação apresentam uma maior similaridade com o algoritmo proposto no capítulo 4.

## Capítulo 4

# Algoritmo de Codificação Diferenciada

Este capítulo apresenta em detalhes o algoritmo de compressão de dados lógica, também chamado de supressão de dados, baseada em codificação diferenciada. Esse algoritmo trabalha em conjunto com a técnica de agregação de dados para aumentar o tempo de vida útil de uma RSSF. A seção 4.1 apresenta as motivações para o uso das técnicas adotadas neste trabalho. Os papéis dos elementos da RSSF são apresentados na seção 4.2. A seção 4.3 apresenta todas as fases do protocolo de comunicação de dados que une um protocolo de roteamento às técnicas de agregação de pacotes e compressão de dados. A seção 4.4 apresenta o algoritmo de codificação diferenciada em detalhes, bem como apresenta as fases modificadas do protocolo de comunicação de dados usado. Um sumário das principais fases do protocolo de comunicação de dados é apresentado na seção 4.5. A seção 4.6 apresenta as conclusões da utilização conjunta dos métodos apresentados.

### 4.1 Motivação

A motivação para a adoção da técnica de agregação de pacotes é a de que os cabeçalhos dos pacotes compõem uma grande parte dos pacotes de dados dos nós sensores. Esses cabeçalhos são necessários para tornar possível a comunicação em uma rede de sensores sem fio, mas eles não carregam qualquer informação sobre o fenômeno que está sendo sensoriado. Desde que muitos sensores reportam suas leituras periodicamente para o nó sorvedouro aproximadamente ao mesmo tempo e com cabeçalhos de pacotes similares, consideráveis economias de energia podem ser alcançadas combinando diferentes pacotes em um pacote com um único cabeçalho.

O segundo esquema que permite economizar energia é um algoritmo baseado em codificação diferenciada das leituras dos nós sensores. Nos pontos de agregação, há o encontro de dois ou mais fluxos de dados passando por um mesmo nó. Em redes com grande quantidade de nós sensoriando o mesmo ambiente, há muitas leituras iguais ou bastante semelhantes. Aproveitando-se do fato da alta redundância de dados, pode-se tentar eliminar essa redundância através do envio de leituras diferenciadas para uma base comum de dados.

O uso das duas técnicas descritas acima também diminui a probabilidade de colisões de pacotes porque elas permitem que a mesma quantidade de informação de leituras seja enviada

dentro de um único pacote, reduzindo a quantidade de vezes em que os nós disputam pelo acesso ao meio de comunicação sem fio.

O algoritmo de codificação diferenciada é usado para reduzir a energia consumida pelos nós sensores, de modo a aumentar o tempo de vida útil da RSSF em conjunto com um protocolo de roteamento baseado em árvore, detalhado na seção 4.3.

## 4.2 Papéis dos elementos da rede

Os elementos pertencentes à rede de sensores sem fio, utilizados neste estudo, podem ser categorizados de acordo com o seu papel desempenhado na rede. São os seguintes nós:

- **Sensores:** Formam a maioria dos nós da rede. Sensoriam fenômenos físicos no ambiente e geram leituras baseadas nessas medidas.
- **Sorvedouros:** Observam as leituras de múltiplos sensores e, baseados nessas leituras, comandam os nós atuadores. Neste trabalho, há apenas um nó sorvedouro, também chamado de estação base ou nó controlador.
- **Atuadores:** Executam determinadas ações para controlar o ambiente. Como será visto nas seções seguintes, esse papel é desempenhado pelos nós bordas.

Qualquer componente da rede pode acumular mais de uma função. Um nó borda, por exemplo, pode funcionar tanto como nó atuador quanto como nó sensor simultaneamente, conforme visto na seção 4.3.1.

## 4.3 Protocolo de comunicação de dados

Um protocolo de comunicação de dados compreende um protocolo de roteamento e, opcionalmente, algoritmos para agregar e comprimir os dados sensorizados. Pesquisadores têm proposto diferentes protocolos de roteamento para redes de sensores sem fio (Al-karaki e Kamal, 2004). O algoritmo de codificação diferenciada proposto neste trabalho requer um protocolo de roteamento para (1) fornecer e distribuir o resultado da consulta para todos os nós da rede e (2) encontrar pelo menos uma rota dos nós sensores de volta para o nó controlador.

*Data Funneling* é um protocolo de comunicação de dados proposto por Petrovic et al., (2003). Esse protocolo possui um protocolo de roteamento e um algoritmo de codificação por ordenação. O protocolo *Data Funneling*, por sua vez, incorporou e adicionou características próprias às fases de configuração e de comunicação dos dados ao protocolo de roteamento eficiente de energia proposto por Shah e Rabaey (2002), chamado *Energy Aware Routing*.

A idéia básica do algoritmo *Energy Aware Routing* é aumentar a sobrevivência da rede através do uso de caminhos sub-ótimos, ocasionalmente, para assegurar que os caminhos ótimos não sejam extintos tão rapidamente. O uso freqüente de poucos caminhos de baixo custo de energia pode conduzir os nós desse caminho a esgotarem suas energias mais rapidamente que outros nós, o que pode acarretar uma partição da rede. Para isso, múltiplos caminhos

são encontrados entre a origem e o destino e cada caminho é associado a uma probabilidade de ser escolhido. Toda vez que um dado é enviado da origem ao destino, um dos caminhos é escolhido aleatoriamente dependendo das probabilidades. Isso significa que nenhum dos caminhos é usado durante toda a fase de reporte das leituras dos nós sensores.

O protocolo *Data Funneling* original possui três fases principais:

1. **Fase de configuração:** Nesta fase, ocorre uma etapa de inundação localizada para encontrar todas as rotas entre os nós sensores e o nó controlador, bem como para calcular os custos associados a essas rotas. Os custos das rotas são calculados baseados na energia remanescente dos nós sensores.
2. **Fase de comunicação dos dados:** Na fase de comunicação dos dados, os dados são enviados dos nós sensores em direção ao nó controlador. As rotas para a propagação dos dados são escolhidas probabilisticamente de acordo com os custos de energia calculados na fase de configuração. O algoritmo *coding by ordering* é usado nesta fase para realizar a codificação dos dados.
3. **Fase de manutenção:** Na fase de manutenção, inundação localizada é realizada do destino às origens para manter todas as rotas consistentes.

O *Data Funneling* foi modificado para incorporar o algoritmo de codificação diferenciada proposto neste trabalho. As fases modificadas e adicionadas ao protocolo são:

1. **Fase de configuração:** O funcionamento dessa fase continua o mesmo.
2. **Fase de formação de base comum:** Fase adicionada após a fase de configuração para formação da base comum de dados das leituras dos nós sensores.
3. **Fase de comunicação dos dados:** O algoritmo *coding by ordering* foi substituído pelo algoritmo de codificação diferenciada para comprimir os dados sensorizados.
4. **Fase de manutenção:** Esta fase foi modificada para manter a base comum dos dados consistente.

A fase de configuração será explicada na subseção 4.3.1. As fases de formação de base comum, de comunicação de dados e de manutenção serão detalhadas dentro da seção sobre o algoritmo de codificação diferenciada (ver seção 4.4). Essa divisão se dá porque após a fase de configuração, mudanças foram realizadas no protocolo de comunicação de dados original do *Data Funneling* para a incorporação do algoritmo de codificação diferenciada.

### 4.3.1 Fase de configuração

Um nó controlador é responsável por dividir o espaço a ser monitorado em diferentes regiões quadradas e enviar pacotes de controle, chamados pacotes de interesse, para cada região a ser monitorada, como mostrado na Figura 4.1. Se há regiões dentro da área total que não serão monitoradas, então o controlador não enviará pacotes de interesse para essas regiões.

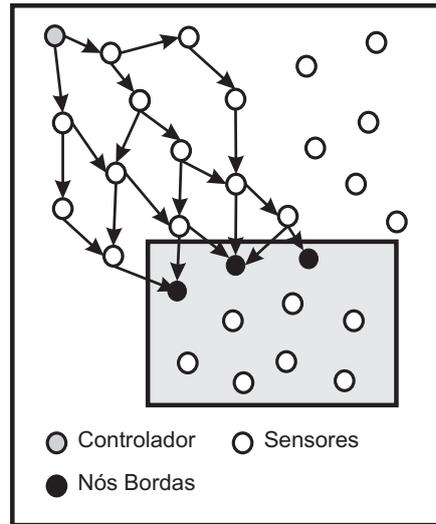


Figura 4.1: Papéis dos elementos da rede (Petrovic et al., 2003).

O nó controlador inicia essa fase inundando as regiões alvo, ou seja, as regiões a serem monitoradas, com pacotes de interesse. Observa-se na Figura 4.1 que nem todos os nós sensores propagam o pacote em direção à região alvo. Isso caracteriza a fase de inundação direcional, também chamada de fase de propagação de interesses. Os campos do pacote de interesse podem ser vistos na Figura 4.2. O campo de custo de comunicação para o nó controlador do pacote de interesse é setado para zero antes do envio desse pacote em direção à região a ser monitorada (ver equação 4.1).

$$Cost(Controlador) = 0 \quad (4.1)$$

Custo para o controlador	Energia residual	Se pertence à região alvo
Coordenadas do nó controlador		Coordenadas da região alvo

Figura 4.2: Formato do pacote de interesse.

Cada nó que está na região de interesse (região alvo), ao receber o pacote de interesse, checka se ele está na região a ser monitorada através das coordenadas da região alvo enviadas nesse pacote. Se ele não está, ele computa seu custo para a comunicação com o nó controlador (ver equação 4.7), atualiza esse custo dentro do pacote de interesse e o envia em direção à região especificada.

Todos os nós intermediários encaminham o pacote de interesse somente para os vizinhos que estão mais perto da região alvo. Um nó  $i$  somente encaminha um pacote recebido para um nó  $j$  se  $i$  estiver mais longe da região alvo que  $j$  e se  $i$  estiver mais perto do nó controlador

que  $j$ , como mostra a equação 4.2:

$$d(N_i, RegiaoAlvo) \geq d(N_j, RegiaoAlvo) \quad e \quad d(N_i, Controlador) \leq d(N_j, Controlador). \quad (4.2)$$

Na recepção de um pacote de interesse, calcula-se o custo para se enviar um pacote de volta ao vizinho que enviou o pacote de interesse. Esse custo é computado e é adicionado ao custo total da rota para se alcançar o nó controlador. Se um pacote é recebido por um nó  $N_j$  de um nó  $N_i$ ,  $N_j$  calcula o custo do caminho como:

$$C_{N_j, N_i} = Cost(N_i) + Metric(N_j, N_i), \quad (4.3)$$

onde  $Cost(N_i)$  é o custo do nó  $N_i$  para se alcançar o nó controlador. A métrica  $Metric(N_j, N_i)$  é calculada baseada no campo de energia residual preenchido pelo nó que reencaminhou o pacote de interesse. Essa métrica é calculada como:

$$C_{ij} = e_{ij}^\alpha R_i^\beta, \quad (4.4)$$

onde  $C_{ij}$  é a métrica de custo entre os nós  $i$  e  $j$ ,  $e_{ij}$  é a energia usada para transmitir e receber no *link* de comunicação, enquanto que  $R_i$  é a energia residual no nó  $i$  normalizada segundo a energia do nó inicial. Os fatores de peso  $\alpha$  e  $\beta$  podem ser escolhidos para encontrar o caminho de custo de energia mínimo, o caminho com nós que têm mais energia ou a combinação de ambos (Shah e Rabaey, 2002).

Caminhos que têm um alto custo são descartados e não são adicionados à tabela de roteamento. Somente os vizinhos de  $N_j$  com caminhos de baixo custo são adicionados à tabela de roteamento, que mantêm múltiplos caminhos dos nós sensores até o nó controlador. Novas rotas são inseridas na tabela de roteamento  $FT_j$  segundo a regra abaixo:

$$FT_j = \{i | C_{N_j, N_i} \leq \alpha \times (\min C_{N_j, N_k})\}. \quad (4.5)$$

Cada nó  $N_j$  determina a probabilidade de escolher uma rota de encaminhamento de dados via o nó  $N_i$  nas tabelas de encaminhamento  $FT_j$ . A probabilidade de escolher uma rota é inversamente proporcional ao custo e pode ser vista como:

$$P_{N_j, N_i} = \frac{\frac{1}{C_{N_j, N_i}}}{\sum_{k \in FT_j} \frac{1}{C_{N_j, N_k}}} \quad (4.6)$$

Cada nó  $N_j$  tem um número de vizinhos através do qual ele pode rotear pacotes para o destino.  $N_j$  pode então calcular o custo médio para alcançar o destino usando os vizinhos da tabela de encaminhamento:

$$Cost(N_j) = \sum_{i \in FT_j} P_{N_j, N_i} C_{N_j, N_i}. \quad (4.7)$$

Este custo médio,  $Cost(N_j)$  é definido no campo custo para o nó controlador do pacote

de interesse e no campo custo para o nó borda no pacote de interesse modificado (ver Figuras 4.2 e 4.3).

Quando um nó, que está na região alvo, recebe um pacote de interesse de um nó vizinho, que está fora da região alvo, então a fase de inundação direcional é concluída. O nó percebe que ele está na borda da região e se auto-designa para ser um nó borda, como é mostrado na Figura 4.1.

Cada nó borda computa seu custo de comunicação com o controlador da mesma maneira como foi feita pelos nós que estão fora da região alvo durante a fase de inundação. Um nó borda inunda a região de interesse agora com uma versão modificada do pacote de interesse. O custo para alcançar o nó controlador assume o valor zero e torna-se o custo para alcançar o nó borda. Dentro da região, cada nó mantém uma tabela com o custo de comunicação para se alcançar o nó borda e não para se alcançar o nó controlador. Intuitivamente, é como se o nó borda se tornasse o controlador da região alvo.

Dois novos campos são adicionados ao pacote de interesse modificado (ver Figura 4.3). Um dos campos mantém o número de saltos entre o nó borda e o nó corrente que está processando o pacote. O outro campo especifica o custo do nó borda para comunicação com o nó controlador, e, este campo, uma vez definido pelo nó borda, não se modifica à medida que o pacote é encaminhado de um nó para outro dentro da região alvo.

Custo para o nó controlador	Custo para o nó borda	Energia residual
Saltos para o nó borda		Se pertence à região alvo
Coordenadas do nó controlador		Coordenadas da região alvo

Figura 4.3: Formato do pacote de interesse modificado.

Uma vez que os nós dentro da região alvo recebem o pacote de interesse modificado dos nós bordas, eles passam a rotear suas leituras para o nó controlador via um dos nós bordas da região. Desde que haja muitos nós bordas dentro da região, a máxima agregação das leituras dos nós sensores requer que todos os nós dentro da região concordem em rotear seus dados via o mesmo nó borda durante cada rodada de reporte de leituras para o nó controlador. Isto é alcançado pelos nós sensores da região alvo, pois eles aplicam a mesma função determinística para os custos de todos os nós bordas para se alcançar o nó controlador. Como todos os nós aplicam a mesma função para as mesmas entradas, todos eles irão computar o mesmo escalonamento, elegendo o mesmo nó borda. O nó borda eleito coleta todos os dados recebidos durante cada rodada de reporte. Isso permite que os nós bordas com baixo custo de comunicação para o nó controlador sejam usados mais frequentemente em relação àqueles nós com alto custo.

O protocolo de roteamento cria *clusters* (agrupamentos) com hierarquia dinâmica dentro da rede de sensores. Não há um único *cluster-head*, cuja falha possa ser crítica para o funcionamento da rede. Em vez disso, os nós bordas podem agir de modo a equilibrar a carga

entre eles. O nó controlador pode redefinir as regiões, baseado nas suas áreas de interesse, forçando os nós a dividirem-se em novos *clusters* e a elegerem novos conjuntos de nós bordas. O controlador pode redefinir as regiões baseado nos dados que ele recebe dos nós e/ou na energia remanescente dos nós que não têm grandes reservas de energia para atuarem como nós bordas.

O protocolo de roteamento é baseado no método de roteamento probabilístico (Shah e Rabaey, 2002) e é usado de duas maneiras: (i) para se encaminhar as leituras dos nós sensores em direção ao nó borda escolhido e (ii) para os nós bordas encaminharem o pacote comprimido em direção ao nó controlador. Toda vez que dados são enviados da origem ao destino, um dos caminhos é aleatoriamente escolhido dependendo das probabilidades de cada vizinho na tabela de rotas como descrito na equação 4.6. Isto significa que nenhum dos caminhos é usado durante todo o tempo, evitando o uso do mesmo caminho durante todas as rodadas.

## 4.4 Algoritmo de Codificação Diferenciada

Esta seção detalha todas as fases do protocolo de roteamento do *Data Funneling* que foram modificadas para a inserção do algoritmo de codificação diferenciada. Correspondem as fases de: formação de base comum, comunicação dos dados e manutenção.

### 4.4.1 Fase de formação de base comum

Após a fase de inundação direcional, pode-se dar início à fase de comunicação dos dados dos nós sensores para o nó controlador, via nós bordas, porém, os nós sensores de cada região ainda não possuem uma base comum para poder enviar suas leituras através de codificação diferenciada para uma base comum.

O nó controlador envia uma mensagem em direção à região alvo para que os nós bordas possam dar início à fase de formação de base comum. Os nós bordas da região alvo encaminham esse pedido para os nós sensores da região. Quando um nó sensor recebe esse pedido, ele começa a enviar leituras, em um intervalo de tempo determinado no pacote recebido, para o nó borda eleito para cada intervalo de tempo.

A quantidade de rodadas necessárias para que o nó borda consiga formar uma base comum vai depender da variação das leituras coletadas pelos nós sensores. Se cada sensor está enviando leituras muito diferentes de uma rodada para outra, mais rodadas serão necessárias para se escolher uma base comum com um nível de confiança aceitável. Quando o nó borda encontra uma base comum de bits aceitável, ele envia a base comum escolhida para os nós da sua região. Porém, um intervalo máximo de tempo pode ser informado pelo nó controlador para os nós bordas processarem a formação da base comum.

Na Figura 4.4, mostra-se como uma base comum de bits é formada a partir de várias leituras de nós sensores. Para cada leitura, verifica-se qual é o valor do bit em cada posição do dado. Por exemplo, para o bit mais significativo, o bit zero apresentou-se mais presente. Para o segundo bit mais significativo, o bit zero também é o mais freqüente. E assim, sucessivamente,

faz-se a varredura em cada posição dos bits das leituras dos nós sensores para compor a base comum.

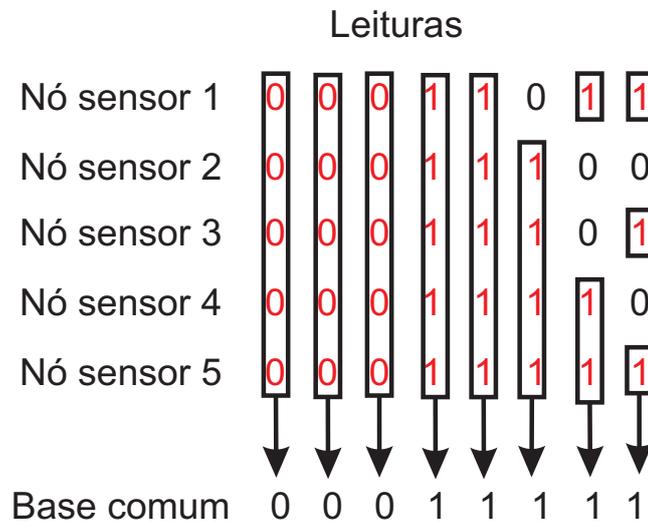


Figura 4.4: Base comum de bits para cinco leituras de nós sensores.

Para cada rodada de reporte de dados, o nó borda computa uma base comum. Após  $n$  rounds, o nó borda escolhe a base comum mais freqüente da lista de bases formadas até então para enviá-la aos nós sensores (ver Figura 4.5).

	Freqüência	
Base 1	0 0 0 1 0 0 1 0	2
Base 2	0 0 0 1 0 0 0 1	2
Base 3	0 0 0 1 0 0 0 0	1
Base 4	0 0 0 1 1 1 1 1	4
Base 5	0 0 0 1 0 0 1 0	1

→ Base escolhida

Figura 4.5: Lista de bases comuns no nó borda.

Os nós sensores da região, ao receberem a base comum, armazenam-na. A partir de então, os sensores irão reportar as suas leituras de acordo com a diferença para a base comum, como pode ser observado na Figura 4.6.

A motivação para a escolha de uma base comum de bits advém do fato de que em uma rede densa, os nós sensores de uma mesma região sensoriam leituras iguais ou muito próximas. Assim, os bits mais significativos, geralmente, permanecem os mesmos para leituras subsequentes, como pode ser visto na Figura 4.7. Os nós sensores tendem a reportar leituras iguais ou muito semelhantes ao longo de um grande intervalo de tempo. Essas leituras diferem em poucos bits para a base comum. Assim, somente os bits diferentes são enviados para o nó

	Valor decimal	Binário	Diferença
Base		0 0 0 1 1 1 1 1	
Leitura 1	27	0 0 0 1 1 0 1 1	0 1 1
Leitura 2	28	0 0 0 1 1 1 0 0	0 0
Leitura 3	29	0 0 0 1 1 1 0 1	0 1
Leitura 4	30	0 0 0 1 1 1 1 0	0
Leitura 5	31	0 0 0 1 1 1 1 1	-

Figura 4.6: Codificação diferenciada das leituras dos nós sensores.

controlador. Se a leitura de um sensor é igual à base, o campo de dados presente no pacote de dados a ser enviado irá vazio.

	Valor decimal	Binário
Base		0 0 0 1 1 0 1 0
Leitura 1	27	0 0 0 1 1 0 1 1
Leitura 2	28	0 0 0 1 1 1 0 0
Leitura 3	29	0 0 0 1 1 1 0 1
Leitura 4	30	0 0 0 1 1 1 1 0
Leitura 5	31	0 0 0 1 1 1 1 1

↓  
Bits significativos iguais

Figura 4.7: Bits mais significativos são iguais para valores consecutivos.

Um maior ganho de energia também pode ser obtido se a aplicação não exigir um nível de 100% de precisão. Se uma margem de erro na precisão da leitura a ser reportada puder ser aceita, alguns bits de leitura podem ser suprimidos. Por exemplo, se a aplicação permite uma leitura com erros de até uma unidade, o bit menos significativo de cada leitura pode ser suprimido. Se a aplicação permite erros de até três unidades, os dois bits menos significativos de cada leitura podem ser suprimidos. E assim, sucessivamente. A possibilidade de aceitação de margem de erro de leituras é indicada no pacote de formação de base, que é enviado pelo controlador em direção à região alvo.

#### 4.4.2 Fase de comunicação dos dados

Os fluxos de dados dentro da região dos sensores para os nós bordas sofrem agregação de pacotes ao longo do caminho, como mostra a Figura 4.8. Quando se inicia uma nova rodada para o envio de leituras para o nó controlador, os nós sensores não enviam seus dados diretamente. Eles esperam uma quantidade de tempo inversamente proporcional à sua distância em número de saltos para o nó borda que será usado na rodada corrente. Isso permite que os nós mais distantes do nó borda enviem seus dados primeiro para os nós que estão mais próximos. Desta maneira, nós mais próximos ao nó borda irão receber primeiro as leituras de nós descendentes na árvore de roteamento para processá-las junto com suas próprias leituras. No final, todos os dados enviados pelos nós sensores da região serão coletados em um único nó borda e enviados em um único pacote para o nó controlador.

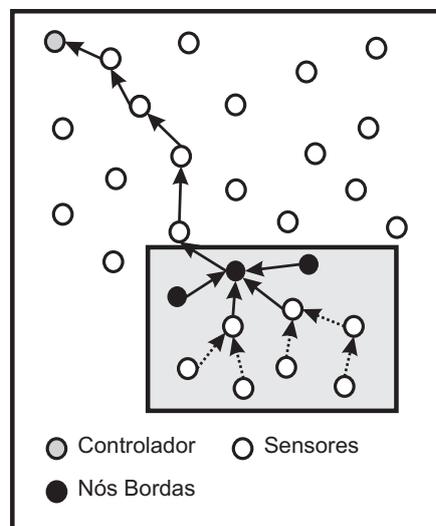


Figura 4.8: Agregação de pacotes dentro da região alvo.

Seja  $\alpha$  a razão entre o número de bits em um cabeçalho de pacotes e o número total de bits em um pacote, que contém o cabeçalho e uma única leitura de um sensor. Seja  $r$  o número médio de leituras por pacote transmitido quando agregação de pacotes é empregada. Então, a energia total dispendida pela rede na comunicação é reduzida por  $\alpha \frac{(r-1)}{r} 100\%$  devido à agregação feita nos pontos de agregação.

Nos pontos de agregação, além da própria agregação de pacotes, aplica-se a técnica de codificação diferenciada nas leituras dos sensores. Ao receber um pacote de dados de um sensor diferente, o nó receptor armazena essa leitura diferenciada recebida em um *buffer*. O intervalo de reporte de leituras para o nó borda é determinado no pacote de formação de base. Um sensor envia suas leituras comprimidas para o nó borda quando esse intervalo expira ou quando recebe uma leitura de um sensor que já enviou sua leitura na rodada corrente. Neste momento, todas as leituras diferenciadas armazenadas no *buffer* são agregadas. Os bits diferenciados são concatenados como mostra a Figura 4.9. Há um *byte* a mais para indicar o

término das leituras diferenciadas chamado *Posição das leituras*, onde cada bit 1 significa o término de uma leitura diferenciada.

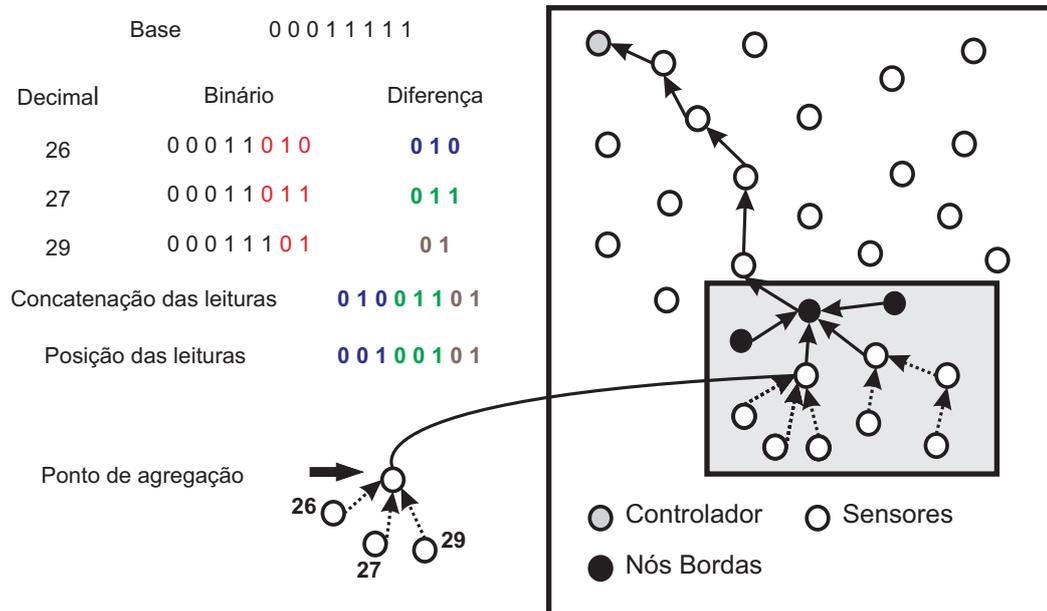


Figura 4.9: Concatenação de leituras.

Se os bits das leituras diferenciadas ultrapassarem um *byte*, um novo *byte* deve ser enviado para conter os bits restantes. Além disso, para cada *byte* enviado de dados, um *byte* a mais será necessário para identificar a posição final da leitura de cada sensor.

Nem sempre codificação diferenciada pode ser empregada. Antes, o nó sensor verifica se a aplicação dessa técnica é vantajosa. Por exemplo, um nó folha possui apenas uma leitura para enviar. Não faz sentido aplicar a técnica de codificação diferenciada em apenas uma leitura, pois para cada *byte* de leitura a ser enviada, precisa-se também enviar um *byte* que conterà as posições das leituras. Como será apresentado no capítulo 5, observa-se que, se redundância temporal for permitida pela aplicação, a técnica de codificação diferenciada poderá ser aplicada nos nós folhas e em nós intermediários que recebam apenas um fluxo de dados além de suas próprias leituras.

A Figura 4.10 apresenta como a descompressão de leituras diferenciadas é feita. No *byte* que indica as posições das leituras dos sensores, cada bit um indica a posição final de uma leitura diferenciada. As seqüências 010, 011 e 01 representam leituras diferenciadas, que podem ser revertidas às leituras originais através da base comum de dados.

#### 4.4.3 Fase de manutenção

Nesta fase, realiza-se inundação localizada para manter atualizada a tabela de rotas dos nós sensores que fora construída durante a fase de propagação de interesses. Este processo pode ser desencadeado por vários elementos da rede de acordo com certos critérios:

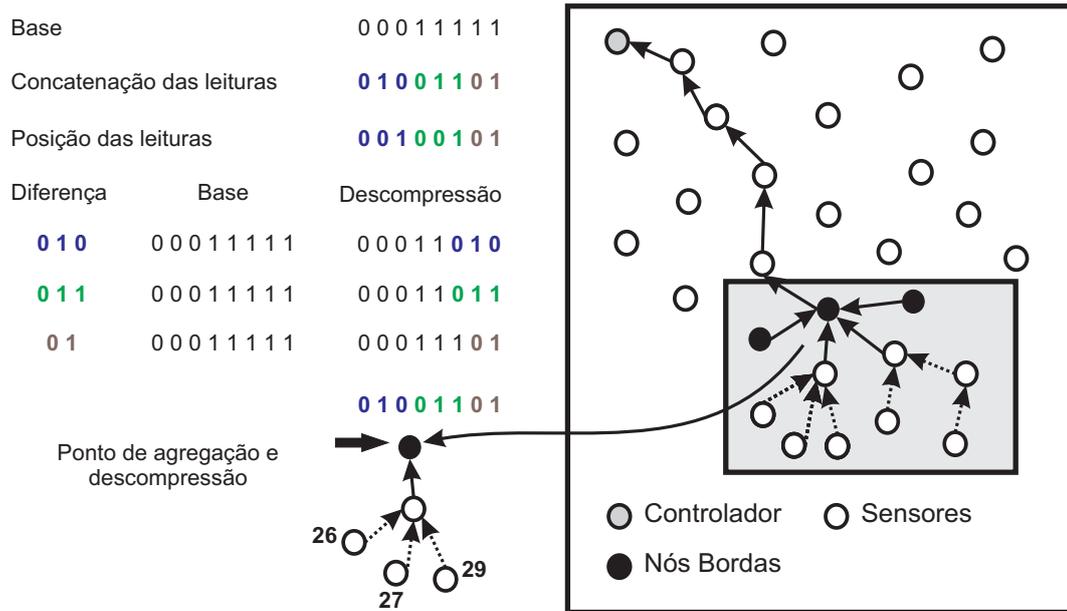


Figura 4.10: Descompressão de leituras.

- **Baixa no nível de energia:** um nó borda que esteja participando ativamente de rodadas pode iniciar o processo de atualização local das tabelas de rota dentro da região alvo quando a sua quantidade de energia disponível cai abaixo de um limite. Esse processo é necessário para atualizar os custos dos nós sensores para os nós bordas de uma região alvo. Assim, novos nós bordas serão escalonados pelos nós sensores durante cada rodada.
- **Falha de nós bordas:** O nó controlador, ao perceber falhas de um nó borda de uma região, pode reiniciar o processo de inundação localizada para realizar a detecção de novos nós bordas.
- **Periodicidade:** Periodicamente, o nó borda pode iniciar o processo de atualização das rotas.

Observa-se que os nós bordas e os nós controladores que iniciam a fase de inundação localizada comportam-se como nós atuadores, pois eles executam ações para controlar a rede.

Para manter a base comum da região alvo sempre atualizada de acordo com as variações dos dados da região, um mecanismo de inundação localizada é usado para propagar uma nova base comum de leituras para a região alvo. Uma nova base comum só é propagada quando a taxa de compressão média, calculada após uma quantidade pré-determinada de rodadas nos nós bordas, cai abaixo de um limiar definido pelo usuário. Se a taxa de compressão média não atingir os limites determinados, uma nova base para atender as variações de leitura é enviada para a região de monitoramento.

## 4.5 Síntese do protocolo

O protocolo de roteamento e o de algoritmo de codificação diferenciada podem ser sumarizados como segue:

### **Fase de configuração:**

1. O nó controlador divide a área que ele deseja monitorar em regiões quadradas.
2. Ele inicia a fase de inundação direcional em direção a cada região alvo.
3. Cada nó intermediário grava o custo para se alcançar o nó controlador como na equação 4.3.
4. Quando o pacote alcança a região, o primeiro nó da região que recebe o pacote se auto-designa como um nó borda.
5. O nó borda adiciona dois novos campos ao pacote, o custo e o número de saltos para se alcançar o nó borda. Ele então inunda a região com o pacote de interesse modificado.
6. Todos os nós dentro da região recebem pacotes de todos os nós bordas. Baseados na energia requerida individualmente pelos nós bordas, eles computam um escalonamento de nós bordas onde os dados serão agregados e comprimidos.
7. Antes de iniciar a fase de coleção de dados, o nó controlador inicia a fase de formação de base comum. Nesta fase, todos os nós sensores da região enviam suas leituras para os nós bordas. Os nós bordas eleitos da região calculam uma base comum de bits e enviam essa base de bits comuns para as suas regiões.

### **Fase de formação de base comum:**

1. O nó controlador envia uma mensagem em direção às regiões alvos para que os nós bordas iniciem a fase de formação de base comum.
2. O nó borda de cada região alvo encaminha esse pedido para os nós sensores de sua região.
3. Os nós sensores começam a enviar suas leituras para o nó borda eleito de sua região.
4. Após o intervalo de tempo indicado para o término dessa fase, o nó borda de cada região calcula a bases comum de dados, enviando-as em seguida para os nós sensores de sua região alvo.

### **Fase de comunicação de dados:**

1. Quando um nó sensor tem uma leitura que precisa ser enviada para o controlador, ele usa o escalonamento para escolher o nó borda para qual o dado será enviado naquela rodada.

2. O nó sensor espera por um tempo inversamente proporcional ao número de saltos para o nó borda antes de enviar o pacote. Na leitura a ser enviada é aplicado o algoritmo de codificação diferenciada.
3. Ao longo de todo o caminho até o nó borda, os pacotes de dados são agregados e codificação diferenciada é aplicada às leituras.
4. O nó borda coleta todos os pacotes e os envia em um único pacote de volta para o nó controlador, usando roteamento probabilístico.

**Fase de manutenção:**

1. Um elemento da rede, quer seja o nó controlador ou o nó borda e quer seja periodicamente ou através da detecção de um evento importante, inunda a região alvo para reconstruir as tabelas de roteamento a fim de manter as rotas consistentes.

## 4.6 Conclusão

O algoritmo de codificação diferenciada reduz a redundância dos dados coletados pelos nós sensores, diminuindo assim a quantidade de mensagens circulando na rede. Em aplicações de monitoramento ambiental, onde observa-se a coleção de dados completa, é desejável fazer a minimização *off-line* dos dados coletados. Como cada nó sensor da região alvo possui uma base comum de bits, onde apenas os bits diferentes para essa base são enviados. A técnica de agregação de pacotes permite unir várias leituras diferenciadas em um único pacote a ser enviado para o nó controlador, diminuindo a probabilidade de colisões no meio sem fio. Espera-se boa economia de energia aplicando-se concomitantemente as técnicas de agregação e compressão de dados lógica baseado em codificação diferenciada. O algoritmo de codificação diferenciada deve ser aplicado juntamente com técnicas que favoreçam o agrupamento de nós com leituras similares, para que a base comum de dados possua maior tamanho possível, diminuindo as diferenças das leituras dos nós sensores para a base comum.

## Capítulo 5

# Resultados de Simulação

O presente capítulo tem como objetivo apresentar os resultados de simulação para o algoritmo de codificação diferenciada apresentado no capítulo 4. A seção 5.1 apresenta os algoritmos usados nas simulações. A seção 5.2 apresenta o algoritmo de codificação por ordenação, usado como comparação nos experimentos. A seção 5.3 apresenta a configuração dos experimentos. A seção 5.4 apresenta as simulações realizadas. A seção 5.5 apresenta as considerações finais sobre os resultados dos testes.

### 5.1 Algoritmos para testes

Para avaliar o algoritmo de codificação diferenciada, usou-se duas versões de protocolos, que une um algoritmo de roteamento e um algoritmo de compressão de dados - algoritmos da camada de rede e da camada de aplicação, respectivamente - cada uma considerando as seguintes peculiaridades:

1. o uso do algoritmo de roteamento original do *Data Funneling* (ver seção 4.3) com o algoritmo *coding by ordering* (ver seção 5.2), também proposto por Petrovic et al. (2003);
2. o uso do algoritmo de roteamento *Data Funneling* com o algoritmo de codificação diferenciada proposto neste trabalho.

### 5.2 Codificação por ordenação

Quando muitas leituras são transmitidas e a ordem em que essas leituras chegam ao destino não é importante para a aplicação (isto é, o transmissor deve escolher a ordem em que ele deve enviar essas leituras), a escolha da ordem em que essas leituras deverão ser enviadas pode ser usada para conduzir informações adicionais ao receptor. Através dessa ordem, pode-se suprimir algumas leituras, onde a ordem das leituras enviadas indica o valor da leitura suprimida. A quantidade de leituras suprimidas depende do intervalo de valores em que os dados podem ser gerados e da quantidade de leituras de diferentes sensores presentes no

codificador (nó sensor). Esse é o método de codificação por ordenação, ou *coding by ordering* (Petrovic et al., 2003).

Por exemplo, considere o caso quando há 4 nós sensores com identificadores 1, 2, 3 e 4, na região a ser monitorada, e que esses nós sejam capazes de gerar leituras inteiras entre 0 e 5. O nó borda pode escolher suprimir o valor gerado pelo nó 4, com leitura de valor 3, e escolher uma ordenação apropriada das leituras dos outros 3 nós para indicar o valor gerado pelo nó 4. Pela Tabela 5.1, pode-se observar que a ordem em que as leituras serão concatenadas é: 2, 3 e 1.

Permutação	Valor inteiro
123	0
132	1
213	2
231	3
312	4
321	5

Tabela 5.1: Mapeamento de permutação de três identificadores para inteiros.

Se um nó sensor receber um pacote de dados, cujas leituras estão na ordem 231, significa que a leitura que foi suprimida tem valor 3 e, assim, sucessivamente.

Neste exemplo, como as leituras possuem 6 variações de valores, então precisa-se de pelo menos 3 leituras, devidamente ordenadas, para que elas possam inferir no valor de uma leitura suprimida. A cada 4 leituras, uma pode ser suprimida.

### 5.3 Configuração dos experimentos

Segundo a classificação de configuração, de sensoriamento e de comunicação dos nós sensores proposta por Ruiz (2003), nos experimentos realizados, identifica-se as seguintes características:

- **Composição homogênea:** a rede é composta por nós sensores que possuem a mesma capacidade de *hardware* (com exceção do nó sorvedouro). Os nós também executam o mesmo *software*.
- **Organização hierárquica:** Nos experimentos, os nós da RSSF estão organizados em grupos, segundo o algoritmo de roteamento do *Data Funneling*.
- **Mobilidade estacionária:** Todos os nós permanecem no local onde foram depositados durante toda a operação da rede. Os nós estão dispostos na área de monitoramento seguindo uma distribuição uniforme de localização.
- **Coleta periódica:** Os nós sensores coletam dados em intervalos regulares. Todos os nós sensores da rede são nós fontes.

- **Disseminação programada:** Os nós fontes disseminam as leituras coletadas em direção ao nó sorvedouro em intervalos regulares.
- **Conexão simétrica:** Todos os nós sensores possuem o mesmo alcance.

Os testes foram realizados no simulador de eventos discretos para redes, o *Network Simulator*, versão 2.30 (NS-2, 2004). Os dados gerados pelos nós sensores foram simulados usando uma distribuição normal de média  $\mu$  e desvio-padrão de  $\sigma$ , informados apropriadamente em cada experimento. Cada nó sensor envia dados de tamanho variado, dependente da codificação diferenciada realizada sobre os mesmos, para o nó sorvedouro. A topologia simulada consiste em uma área quadrada de tamanho variado. O ponto de acesso em todas as simulações sempre se encontra na posição (0,0) da área de monitoramento.

Como em áreas de monitoramento ambiental apresenta uma grande quantidade de nós sensores, realizou-se simulações para uma quantidade de nós que variou de 360 nós a 728 nós.

A largura de banda foi ajustada para 250Kbps. O raio de transmissão dos nós sensores foi definido em 50m, que é o raio de alcance aceitável para que nenhum nó sensor fique desconectado da rede no início da simulação, levando-se em consideração a simulação da rede com a menor quantidade de nós sensores (360 nós em uma área de  $490.000m^2$ ).

O modelo de energia usado no simulador foi o *Energy Model*, com energia inicial dos nós sensores de 100J e com os valores de 0.3W para o consumo na recepção e 0.4W para o consumo na transmissão.

O protocolo de roteamento do *Data Funneling* foi escolhido por ser um protocolo que se baseia nos mecanismos de escolhas de rotas de acordo com a energia dos nós, apresentando um consumo homogêneo da rede, devido à recriação de rotas e ao roteamento probabilístico, características mais comuns em protocolos de roteamento em redes de sensores sem fio.

O intervalo de tempo para recriação de rotas foi de 10min para todos os experimentos realizados. A necessidade de reconstrução de rotas é variável para cada cenário experimentado, no entanto, não se fez um estudo aprofundado sobre qual seria o momento exato de reconstrução das rotas em virtude apenas da energia remanescente dos nós sensores, pois foge ao escopo deste trabalho, cujo objetivo é avaliar mais precisamente o algoritmo de codificação diferenciada.

Os parâmetros comuns usados por todos os experimentos são apresentados na Tabela 5.2. Parâmetros variáveis são informados em cada experimento.

Entre as métricas usadas nos experimentos, usou-se: (i) energia média consumida pela rede e (ii) taxa de compressão média dos nós bordas.

A taxa de compressão calculada leva em consideração apenas o ganho com relação à compressão aplicada pelos algoritmo de codificação (diferenciada e por ordenação). O ganho com a agregação de pacotes não foi contabilizado, pois os dois algoritmos usam agregação de pacotes.

Parâmetros	Valor
Localização do nó sorvedouro	(0,0)
Alcance do rádio	50m
Banda	250 kbps
Tempo de simulação	3600s
Início do tráfego de dados	350s
Fim do tráfego de dados	3600s
Média das leituras	29
Desvio-padrão das leituras	1.0
Intervalo de coleta de leituras	60s
Intervalo de disseminação de leituras	60s
Energia inicial dos nós sensores	100J
Energia consumida na transmissão	0,3W
Energia consumida na recepção	0,4W
Posições dos nós sensores	aleatórios

Tabela 5.2: Parâmetros comuns.

## 5.4 Testes

Para calcular o número de execuções necessárias em cada teste para se ter um nível de confiança de 95%, foram executadas, inicialmente, 15 execuções (amostras) de cada experimento abaixo. Calculou-se a média de cada amostra para se determinar o número ideal de execuções  $n$ , de acordo com a equação 5.1:

$$n = \left( \frac{100zs}{rm} \right)^2, \quad (5.1)$$

onde  $z$  é obtido da tabela de distribuição normal,  $s$  é o tamanho da amostra (15),  $r$  é a acurácia desejada (5%) e  $m$  é a média da amostra.

### 5.4.1 Densidade

Neste experimento, avalia-se a eficiência da utilização dos algoritmos de compressão de dados usados, baseado na concentração dos nós por unidade de área.

Neste teste, manteve-se fixa a área da rede ao se aumentar a quantidade de nós sensores dispostos na área de monitoramento. A Tabela 5.3 exhibe os parâmetros variáveis desse experimento.

Ao proceder o aumento da quantidade de nós da rede, verificou-se um aumento do desempenho do algoritmo de codificação diferenciada.

A Figura 5.1 mostra a energia média consumida pela rede. Embora a energia média consumida pela rede aumente à medida que se aumenta a quantidade de nós fontes, a proporção de aumento cai gradativamente. Isso ocorre em virtude do aumento da taxa de compressão média dos dados, pois a altura da árvore de roteamento tende a aumentar. À medida que mais nós são inseridos na rede, aumenta-se a quantidade de fluxos de nós nos pontos de agregação.

Parâmetros	Valor
Média das leituras	29
Desvio-padrão das leituras	1.0
Quantidade de nós	360, 440, 528, 624 e 728
Quantidade de nós fontes	todos
Tamanho da área monitorada	490.000m <sup>2</sup>

Tabela 5.3: Parâmetros para o experimento de densidade.

O roteamento probabilístico também favorece os nós com nível de energia mais alto. Sendo assim, uma maior quantidade de nós escolherá como o próximo nó intermediário o mesmo nó para encaminhar suas leituras. Nesses nós intermediários, há um aumento da taxa de compressão média.

A energia média consumida aumenta devido às fases de configuração e de manutenção, esta última fase possui maior custo por realizar inundação em toda a rede para manter atualizadas as rotas.

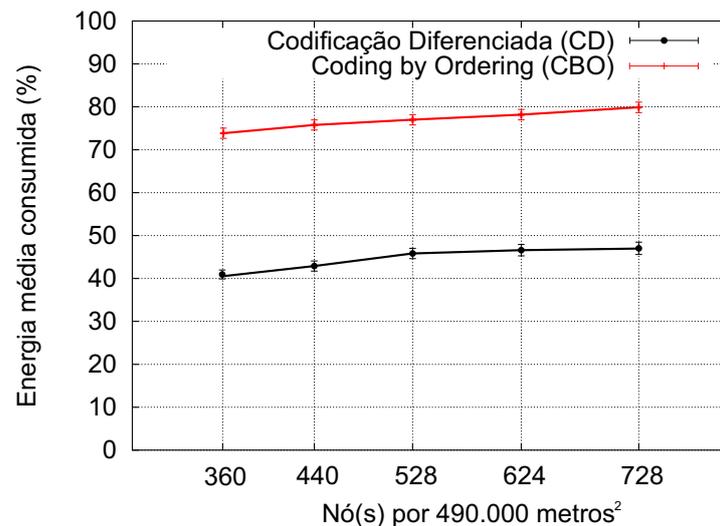


Figura 5.1: Densidade - Energia média consumida pela rede.

O algoritmo de codificação diferenciada apresentou uma taxa de compressão média dos nós bordas entre 50% e 60% em relação a um algoritmo com simples agregação de leituras, enquanto que o algoritmo *coding by ordering* somente conseguiu atingir uma melhora de 20% em relação à agregação de pacotes comum, como pode ser visto na Figura 5.2. A taxa de compressão do algoritmo *coding by ordering* neste experimento atinge um baixo rendimento porque somente a cada 5 leituras (considerando-se 120 leituras possíveis) que chegam em um ponto de agregação durante um mesmo *round*, pode-se suprimir uma leitura. Sendo assim, pacotes de dados que chegam até os nós bordas sem no mínimo 5 leituras neste pacote, não sofrem qualquer compressão de dados, somente a agregação. Outro fator que explica a diferença das taxas de compressão entre esses dois algoritmos é a de que a codificação

diferenciada faz melhor uso dos valores das leituras coletadas pelos nós sensores, enquanto que a codificação por ordenação não utiliza as informações coletadas pelos próprios nós sensores para melhorar a taxa de compressão média de dados.

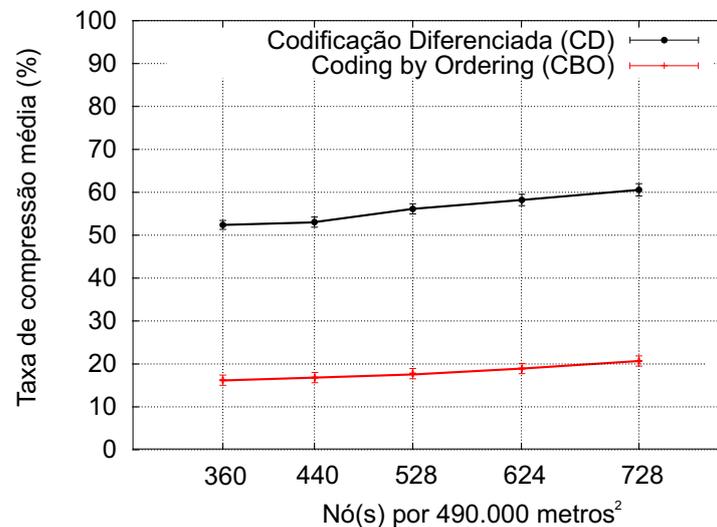


Figura 5.2: Densidade - Taxa de compressão média dos nós bordas.

Em toda a rede, a taxa de compressão média é muito variada, pois essa taxa de compressão depende da quantidade de fluxos de dados que chegam a um nó e se o algoritmo de codificação (diferenciada ou por ordenação) pode ser aplicado ou não, por isso, apresenta-se a taxa de compressão média dos nós bordas nos gráficos de taxa de compressão.

#### 5.4.2 Escalabilidade

Escalabilidade é uma característica desejável em redes de sensores sem fio, pois esse teste indica a habilidade do uso de algoritmos de compressão quando se aumenta a quantidade de nós sensores e a área de monitoramento de maneira uniforme. Um algoritmo/sistema cujo desempenho aumenta com o acréscimo de *hardware*, proporcionalmente à capacidade acrescida, é chamado sistema escalável.

Pretende-se avaliar quão escalável horizontalmente é o algoritmo, ou seja, à medida que se adiciona mais nós à rede de sensores sem fio. O teste de escalabilidade vertical, ou seja, aumento dos recursos dos nós sensores não será realizado.

Os parâmetros utilizados para escalabilidade horizontal podem ser observados na Tabela 5.4.

O gráfico da Figura 5.3 mostra a energia média consumida pelos dois algoritmos de codificação: diferenciada e por ordenação. Em relação ao gráfico correspondente de densidade (Figura 5.1), para redes com 360 e 440 nós, a energia consumida aumenta. Isso acontece porque os nós estão mais concentrados na região, o que não favorece a compressão dos dados, pois a árvore de roteamento formada tende a possuir uma menor altura. Quanto mais balanceada for a árvore de roteamento, maior será o balanceamento da energia consumida por cada nó.

Parâmetros	Valor
Média das leituras	29
Desvio-padrão das leituras	1.0
Quantidade de nós fontes	360, 440, 528, 624 e 728
Tamanho da área monitorada $m^2$	$578^2$ , $639^2$ , $700^2$ , $760^2$ e $821^2$

Tabela 5.4: Parâmetros para a escalabilidade horizontal.

Nas redes com mais de 528 nós, uma melhora é apresentada em relação à energia consumida nas correspondentes redes do gráfico de densidade.

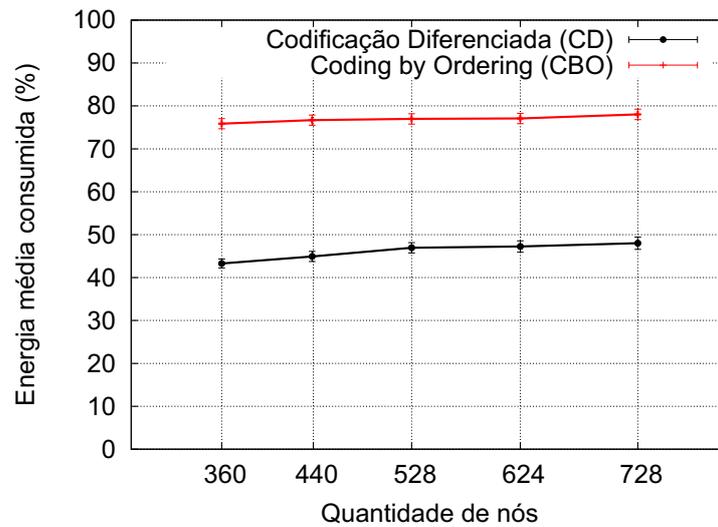


Figura 5.3: Escalabilidade - Energia média consumida pela rede.

Similarmente, a taxa de compressão média dos nós bordas também varia da mesma forma que a energia consumida pelas redes no gráfico da Figura 5.4. Para as redes com 360 e 440 nós, a taxa de compressão apresenta-se pior, porém, sua melhora é observada à medida que os nós ficam mais dispersos na área de monitoramento.

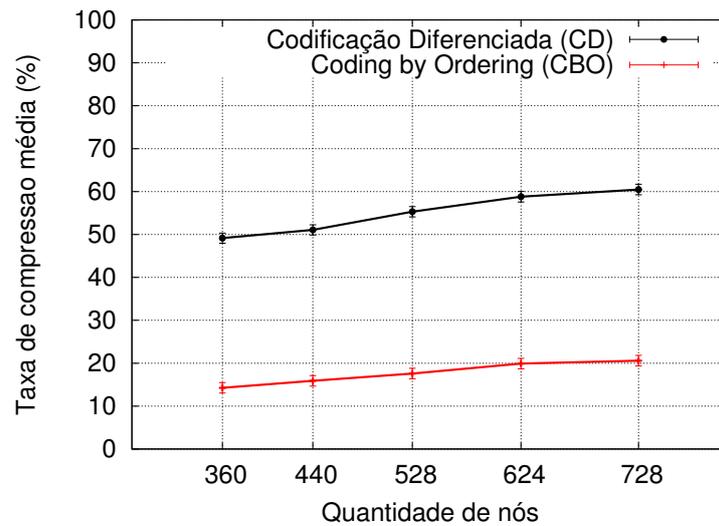


Figura 5.4: Escalabilidade - Taxa de compressão média dos nós bordas.

### 5.4.3 Intervalo de disseminação dos dados

Neste experimento, avaliou-se o comportamento do algoritmo de codificação diferenciada à medida que se aumenta o intervalo de disseminação das leituras dos nós sensores em direção ao nó sorvedouro. Os parâmetros utilizados neste experimento podem ser visto na Tabela 5.5.

Parâmetros	Valor
Intervalo de disseminação dos dados	60s, 120s, 180s e 240s
Média das leituras	29
Desvio-padrão das leituras	1,0
Quantidade de nós fontes	360
Tamanho da área monitorada	490.000m <sup>2</sup>

Tabela 5.5: Parâmetros para disseminação das leituras.

No gráfico da Figura 5.5, pode-se observar que a energia média consumida em relação à energia inicial da rede diminui à medida que se aumenta o intervalo de disseminação dos dados. Isso ocorre devido a um aumento na taxa de compressão dos dados da rede. Na Figura 5.6, pode ser observada a taxa de compressão média dos nós bordas. A taxa de compressão aumenta principalmente pela aplicação da codificação diferenciada nos nós folhas da árvore de roteamento (a partir do intervalo 180) e pela melhora da compressão em nós com baixo grau de vizinhos (em todos os intervalos). O algoritmo de codificação por ordenação também apresenta uma melhora em virtude da redundância temporal, mas a sua taxa de compressão média aumenta em proporções bem menores, pois o aumento da quantidade de fluxos de dados em 2, 3 ou 4 vezes, não aumenta, necessariamente, a taxa de compressão. Por exemplo, se a cada 6 leituras, pode-se suprimir 1 leitura, a cada 12 leituras, pode-se suprimir 2 leituras, ou seja, a taxa de compressão é constante. O aumento da taxa de compressão do *coding*

*by ordering* observado ocorre em decorrência de que a redundância temporal pode favorecer pontos de agregação que estão próximos de suprimir mais uma leitura. Considere um nó sensor que recebe 11 leituras em uma mesma rodada, dobrando-se o intervalo de disseminação dos dados, esse nó irá ter um total de 22 leituras, podendo suprimir 3 leituras, ao invés de apenas 1, caso o intervalo de disseminação não fosse aumentado.

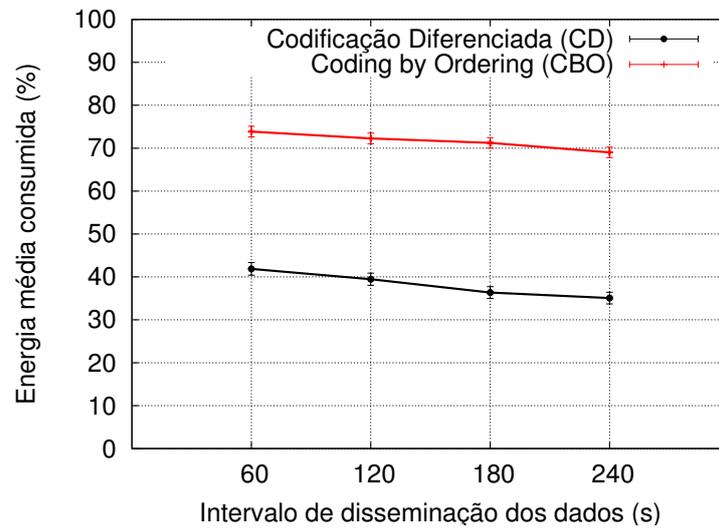


Figura 5.5: Energia média consumida de acordo com o intervalo de disseminação dos dados.

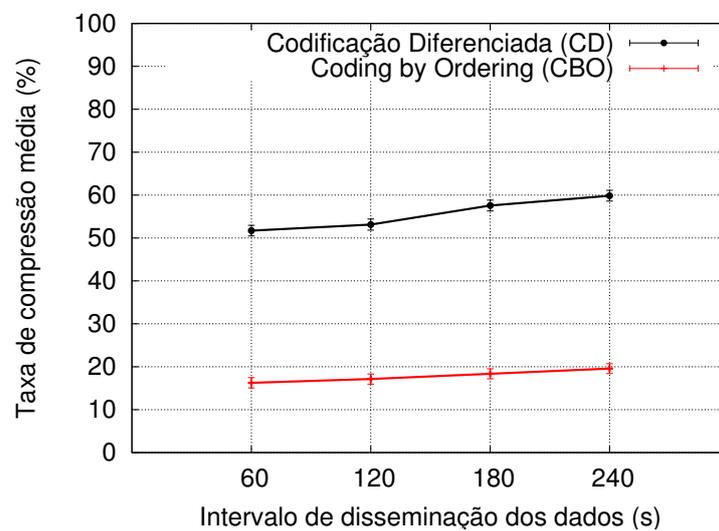


Figura 5.6: Taxa de compressão média dos nós bordas de acordo com o intervalo de disseminação dos dados.

#### 5.4.4 Variação média das leituras

Neste teste avalia-se o desempenho do algoritmo à medida que se aumenta o desvio-padrão das leituras dos nós sensores de uma dada região.

Esse teste é importante para se avaliar o comportamento do algoritmo em regiões com uma maior variedade das leituras. As leituras possuem média 29 e o desvio-padrão dessas leituras varia de 1,0 a 3,0.

Parâmetros	Valor
Média das leituras	29
Desvio-padrão das leituras	variando de 1,0 a 3,0
Quantidade de nós fontes	360
Tamanho da área monitorada	490.000m <sup>2</sup>

Tabela 5.6: Parâmetros para o tamanho do pacote de dados.

O gráfico da Figura 5.7 mostra a energia média consumida à medida que se aumenta o desvio-padrão das leituras. Observa-se que o consumo médio de energia aumenta, pois algumas leituras passam a ser mais diferentes da base, ocorrendo o envio de mais *bits* em direção ao nó borda.

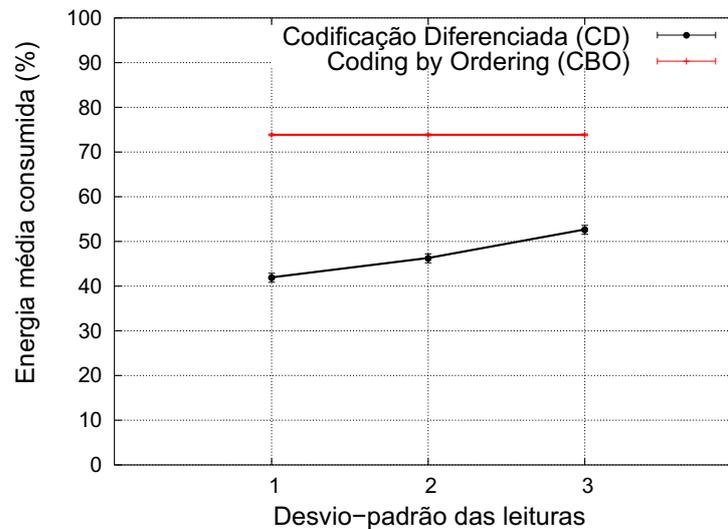


Figura 5.7: Energia média consumida variando-se o desvio-padrão das leituras.

A taxa de compressão média diminui à medida que se aumenta a variação das leituras dos nós sensores de uma mesma região, como observado no gráfico da Figura 5.8. A taxa de compressão média diminui, pois as leituras coletadas são mais diferentes da base, possuindo maior tamanho.

Através deste teste, pode-se concluir que o desempenho do algoritmo de codificação diferenciada depende também de como os nós sensores estão organizados em suas regiões. A divisão em regiões baseadas nas leituras dos nós sensores e do agrupamento de sensores com

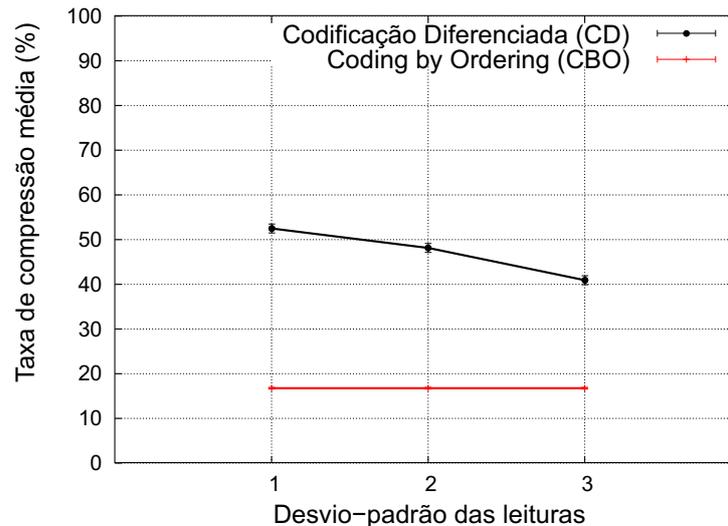


Figura 5.8: Taxa de compressão média dos nós bordas variando-se o desvio-padrão das leituras.

leituras próximas em uma mesma região pode contribuir para aumentar o desempenho do algoritmo. É importante salientar que a eficiência de um algoritmo de compressão de dados, presente na camada de aplicação, depende do bom gerenciamento em todas as camadas da pilha de protocolos da RSSF. Neste teste, observa-se que o agrupamento de nós sensores baseados na proximidade dos valores de leituras contribui para aumentar o tempo de vida útil da RSSF.

A energia média consumida e a taxa de compressão média são constantes para o método *coding by ordering*, tendo em vista que esse método de codificação não faz uso das leituras geradas para obter uma melhor compressão. Mesmo que todas as leituras sejam iguais dentro de uma mesma região, o método não faz supressão das leituras iguais.

#### 5.4.5 Imprecisão das leituras

Pretende-se avaliar o desempenho do algoritmo de codificação diferenciada quando a imprecisão de alguns *bits* nas leituras dos nós sensores for permitida.

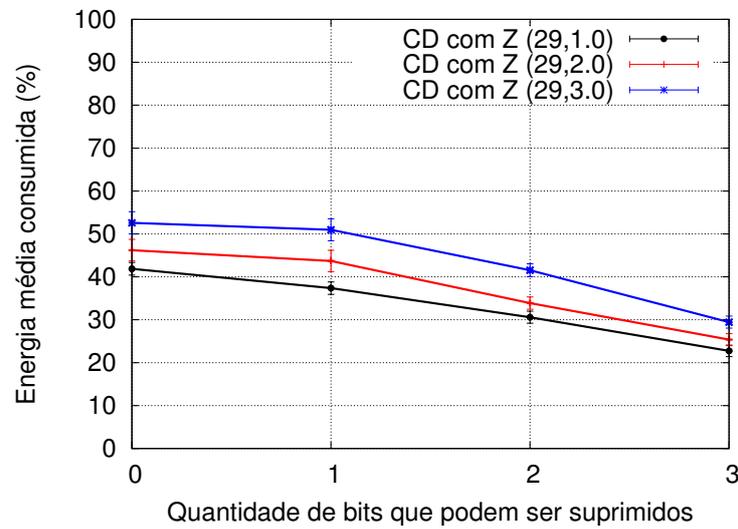
A perda de alguns *bits* na leitura de um nó sensor pode não ser importante para aplicações que não requerem uma precisão exata do dado sensoriado. Para isso, foi simulado um cenário com 360 nós fontes em uma área quadrada de  $490.000m^2$ , onde as aplicações permitiam erros variando de 0 a 3 *bits*, ocasionando um erro na precisão das leituras de 0 a 7 unidades (ver Tabela 5.7).

Observa-se no gráfico da Figura 5.9 que, à medida que se permite mais perdas de *bits*, a energia média consumida pela rede diminui em todos os cenários com diferentes desvios-padrões para as leituras. Isso ocorre devido à transmissão de uma menor quantidade de *bits* dos nós sensores para os nós bordas. À medida que se aumenta o desvio-padrão das leituras dos nós sensores, há uma maior variação na economia do consumo médio de energia. Isso acontece porque, para um cenário onde o desvio-padrão das leituras é maior, o ganho na

Parâmetros	Valor
Variação de <i>bits</i> perdidos	0, 1, 2 e 3
Variação das leituras em unidades	0, 1, 3 e 7
Média das leituras	29
Desvio-padrão das leituras	1, 2 e 3
Quantidade de nós fontes	360
Tamanho da área monitorada	490.00m <sup>2</sup>

Tabela 5.7: Parâmetros para erros.

economia de energia é mais significativo do que para cenários onde as leituras pouco variam, pois esses últimos já teriam menos *bits* para enviar, mesmo se 100% de precisão fosse exigida.

Figura 5.9: Energia média consumida com *bits* suprimidos.

O gráfico da Figura 5.10 mostra a taxa de compressão média dos nós bordas à medida que se aumenta a imprecisão de *bits* nas leituras dos nós sensores. A taxa de compressão também aumenta com uma menor quantidade de *bits* a serem transmitidos ao meio, onde se verifica também que a variação da taxa de compressão é maior em cenários cujas leituras possuem desvio-padrão maior.

O gráfico da Figura 5.11 mostra a latência média da rede. Essa latência diminui à medida que se permite mais imprecisão de *bits*, porque menos *bits* precisam ser enviados para os nós bordas.

Não foram plotados testes com a imprecisão de *bits* usando o método *coding by ordering* porque não há melhora nos resultados, tendo em vista que esse método de codificação não faz uso das leituras geradas para obter uma melhor compressão.

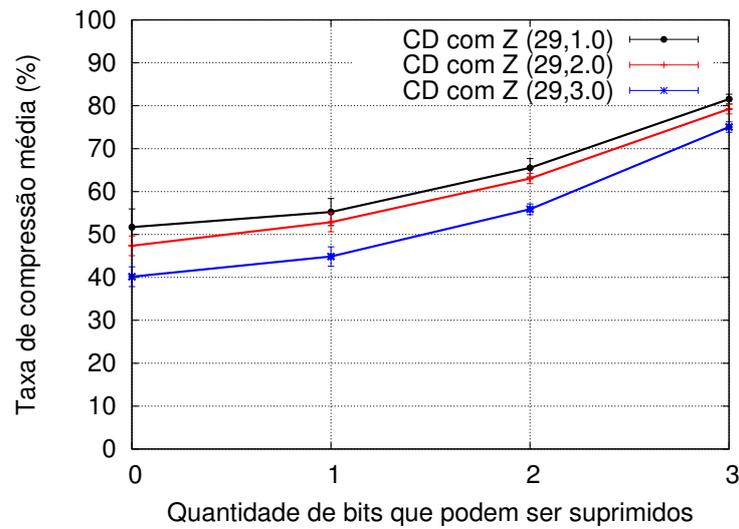


Figura 5.10: Taxa de compressão média dos nós bordas com *bits* suprimidos.

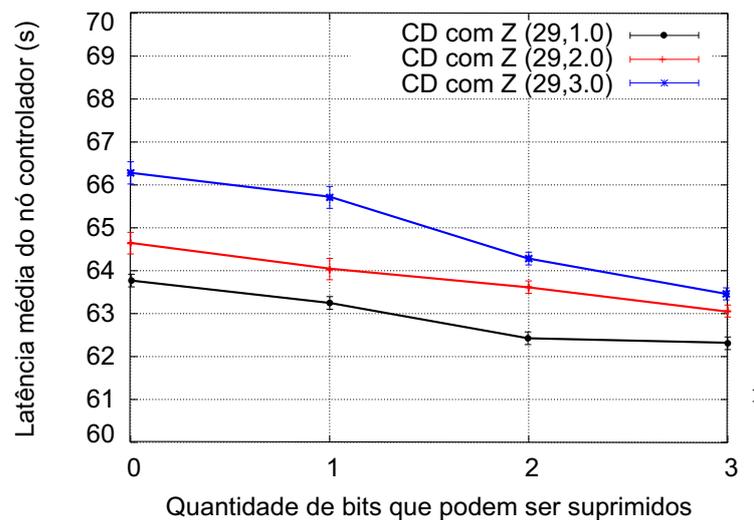


Figura 5.11: Latência média do nó controlador com *bits* suprimidos.

#### 5.4.6 Mudança de base

Neste experimento, realizou-se a mudança de *bits* da base. No tempo de simulação de 1000s, alterou-se a média das leituras dos nós bordas de modo que um dos *bits* menos significativos fosse alterado.

Este teste tem como objetivo verificar a necessidade de mudança de base quando um dos seus *bits* for modificado.

Ao final de cada rodada, o nó borda calcula a taxa de compressão para as leituras recebidas durante a rodada corrente e uma base para essas leituras. A nova base é comparada à antiga. Se houver mudanças, a nova base é enviada propositalmente para se analisar a necessidade do reenvio de uma nova base.

O presente teste avalia o desempenho do reenvio de uma base comum para a região alvo quando há mudanças em um dos quatro *bits* menos significativos (ver Tabela 5.8).

Parâmetros	Valor
Médias das leituras	29
Desvio-padrão das leituras	1
Posições dos <i>bits</i> da base que mudam	1, 2, 3 e 4
Quantidade de nós fontes	360
Tamanho da área monitorada	490.000m <sup>2</sup>

Tabela 5.8: Parâmetros para mudança de base.

Pelo gráfico da Figura 5.12, observa-se que, para o cenário onde a média das leituras é 29 e o desvio-padrão das leituras é 1, reenviar a base quando há mudança do primeiro ou do segundo *bit* menos significativo, aumenta a energia consumida em relação a um cenário quando não há reenvio de base, para o tempo de simulação de 1h. Neste cenário, apenas quando há mudança acima do terceiro *bit*, há uma economia de energia reenviando uma nova base.

Através desse teste, concluiu-se que o envio de uma nova base sempre que houver mudança nem sempre é ideal. A determinação do envio de uma nova base irá depender da média das leituras e de quão diferente é a nova base.

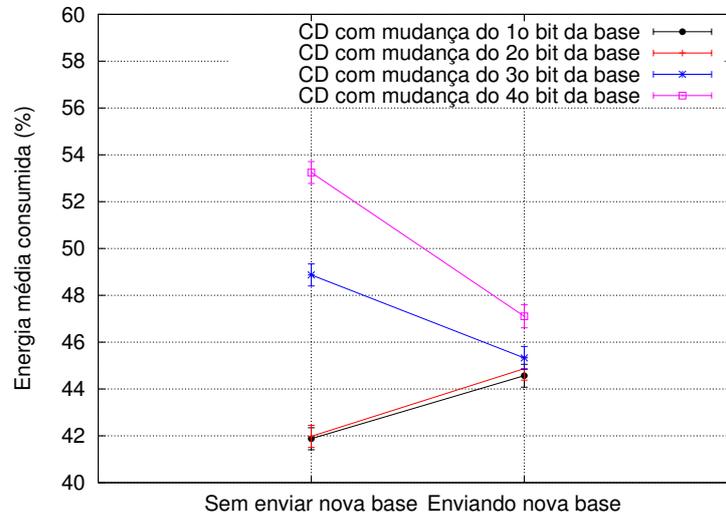


Figura 5.12: Energia média consumida pela rede reenviando uma base.

## 5.5 Conclusão

Neste capítulo, realizaram-se simulações usando o algoritmo de roteamento do *Data Funneling* e dois algoritmos de codificação: diferenciada, proposto neste trabalho, e por ordenação

(Petrovic et al., 2003). Os resultados das simulações apresentaram-se bastantes otimistas quanto à utilização do algoritmo de codificação diferenciada, apresentando bom desempenho em regiões com leituras similares coletadas pelos nós sensores. Bom desempenho também foi apresentado em cenários onde imprecisão das leituras e redundância temporal foram permitidos, pois uma menor quantidade de bits e uma maior taxa de compressão foi alcançada, respectivamente. A atualização da base comum de dados requer um estudo mais aprofundado para se determinar o momento exato de envio de uma nova base, pois nem sempre a atualização de uma base implica uma melhora no desempenho da rede.

## Capítulo 6

# Considerações Finais

A seção 6.1 deste presente capítulo tem como objetivo apresentar as conclusões obtidas com os resultados apresentados neste trabalho. A seção 6.2 apresenta as principais limitações do algoritmo de codificação diferenciada. A seção 6.3 apresenta propostas de trabalhos futuros para ampliar este estudo.

### 6.1 Conclusões

Este trabalho apresenta um algoritmo de codificação diferenciada para redes de sensores sem fio. Em redes com grande quantidade de nós sensoriando o mesmo ambiente, há muitas leituras iguais ou bastante semelhantes, por isso, aproveitando-se do fato da alta redundância de dados, pode-se tentar eliminar essa redundância através do envio de leituras diferenciadas para uma base comum de dados.

Os resultados de simulação mostraram que o algoritmo de codificação diferenciada apresentou boas economias de energia para monitorar regiões onde os nós sensores apresentavam leituras similares. Assim, pode-se concluir que, para um bom desempenho do algoritmo de codificação diferenciada, é importante que o algoritmo de roteamento utilize informações das leituras reportadas pelos nós sensores para dividir a área a ser monitorada em regiões que agrupam nós sensores com leituras similares.

A vantagem da codificação diferenciada é que esse algoritmo permite a compressão lógica dos dados ainda na fonte, permitindo a concatenação de várias leituras diferenciadas ao longo do caminho em direção ao nó controlador nos pontos de agregação. O algoritmo também permite diferentes taxas de compressão para os nós sensores de uma mesma região. Em cenários onde a região possui nós sensores reportando diferentes leituras, o desempenho do algoritmo de codificação diferenciada diminui, em virtude da base comum tentar contemplar as diferentes leituras dos nós sensores. O nó controlador ou os nós bordas podem disparar um evento para tentar a redefinição dos agrupamentos de nós sensores em outras regiões que agrupem nós com leituras similares. Em regiões com leituras similares, onde o desvio-padrão das leituras é pequeno, o algoritmo mostrou-se eficiente. O algoritmo de codificação diferenciada também pode ser usado em combinação com outras técnicas para redução de

dados e a reconstrução da leitura de um nó sensor pode ser realizada em qualquer ponto de agregação, e não somente pelos nós bordas e controladores.

## 6.2 Limitações

O algoritmo de codificação diferenciada, na versão deste trabalho, apresenta algumas limitações de implementação, como: no nível que este algoritmo foi implementado no simulador de redes ns-2 (nível de rede e de aplicação), mesmo quando a leitura diferenciada é menor que um *byte*, é necessário completar todo o *byte* para que a leitura possa ser enviada. À medida que mais de um fluxo de dados ocorre nos pontos de agregação, a codificação diferenciada é aplicada de forma mais eficiente, pois em um mesmo *byte*, pode-se contemplar mais de uma leitura.

Um estudo mais elaborado deve ser realizado para se calcular o melhor momento para se enviar uma nova base quando as leituras dos nós sensores variam bastante. O resultado deste estudo deve ser incorporado à fase de manutenção para garantir melhores economias de energia ao se usar o algoritmo de codificação diferenciada.

Entre as limitações da avaliação, um experimento que poderia ter sido realizado seria a incorporação do algoritmo de codificação diferenciada em um protocolo de roteamento que realizasse o agrupamento inicial dos nós sensores baseados nas suas leituras. A idéia seria identificar regiões com características similares e agrupar os nós por regiões similares. Esse teste poderia avaliar a melhora do tempo de vida útil da rede sabendo-se que, desde o início da atividade da rede, os nós já foram agrupados em regiões com características comuns.

## 6.3 Trabalhos Futuros

Outra versão do algoritmo de codificação diferenciada está em andamento para realizar a codificação dos dados baseados na diferença aritmética para a base comum de dados, ao contrário do algoritmo aqui apresentado, que apresenta uma codificação diferenciada lógica. Resultados preliminares indicam que essa nova versão do algoritmo é ideal para aqueles nós sensores que estão reportando leituras distantes logicamente da base, mas próximos aritmeticamente.

Como trabalho futuro, pretende-se incorporar um protocolo que use tanto o algoritmo de codificação diferenciada lógica, quanto o algoritmo de codificação diferenciada aritmética. O nó sensor irá escolher um dos dois algoritmos para codificar seus dados, avaliando qual deles codifica os dados com um menor número de bits. O protocolo deve ser ainda auto-adaptável para redefinir os agrupamentos de acordo com os valores de energias disponíveis dos nós, bem como com os valores das leituras desses nós.

Pretende-se propor também um modelo *bayesiano* de previsão para construir mapas de características com apenas poucas rodadas do envio de leituras dos nós sensores para a estação-base. Esses modelos apresentam-se adequados para previsão de conteúdo com poucos dados históricos.

# Apêndice A

## Lista de Siglas e Abreviações

**ALVQ** *Adaptive Learning Vector Quantization*

**CDR** *Compression Driven Routing*

**DBA** *Dynamic Bandwidth Assignment*

**DISCUS** *Distributed Source Coding Using Syndromes*

**GD** *Group Data*

**LVQ** *Learning Vector Quantization*

**MSE** *Mean Square Error*

**NS** *Network Simulator*

**OpNet** *Optimized Network Engineering Tools*

**PINCO** *Pipelined In-Network Compression Scheme for Data Collection in Wireless Sensor Networks*

**Q-RC** *Q-Routing with Compression*

**RDC** *Routing Driven Compression*

**RSSF** *Rede de Sensores Sem Fio*

**SPL** *Shared Prefix Length*

**TAM** *Tree Advertisement Message*

# Referências Bibliográficas

- Akyildiz, I. F.; Su, W.; Sankarasubramanian, Y. e Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102 – 114.
- Al-karaki, J. N. e Kamal, A. E. (2004). Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, 11:6 – 28.
- Alqamzi, H. e Li, J. (2005). Exploring distributed and adaptive compression in cluster-based sensor routing. *Second IFIP International Conference on Wireless and Optical Communications Networks*, pp. 530 – 535.
- Arici, T.; Gedik, B.; Altunbasak, Y. e Liu, L. (2003). Pinco: a pipelined in-network compression scheme for data collection in wireless sensor networks. In *Proceedings of 12th International Conference on Computer Communications and Networks*, volume 10, pp. 539 – 544. Citeseer.
- Asada, G.; Dong, M.; Lin, T. S.; Newberg, F.; Pottie, G.; Kaiser, W. J. e Marcy, H. O. (1998). Wireless integrated network sensors: Low power systems on a chip. In *Proceedings of the 24th European Solid-State Circuits Conference*, pp. 551 – 558.
- Beyens, P.; Nowe, A.; Peeters, M. e Steenhaut, K. (2005). Routing with compression in wireless sensor networks: a q-learning approach. *Fifth European Workshop on Adaptive Agents and Multi-Agent Systems*, p. 12.
- Bhatti, S.; Carlson, J.; Dai, H.; Deng, J.; Rose, J.; Sheth, A.; Shucker, B.; Gruenwald, C.; Torgerson, A. e Han, R. (2005). Mantis os: An embedded multithreaded operating system for wireless micro sensor platforms. *Mobile Networks and Applications*, 10(4):579.
- Ciancio, A. e Ortega, A. (2005). A distributed wavelet compression algorithm for wireless multihop sensor networks using lifting. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 4:v/825 – iv/828.
- Dunkels, A.; Gronvall, B. e Voigt, T. (2004). Contiki-a lightweight and flexible operating system for tiny networked sensors. In *29th Annual IEEE International Conference on Local Computer Networks, 2004*, pp. 455–462. IEEE Computer Society.
- Figueiredo, C. M. S. (2007). *Auto-organização em redes de sensores sem fio*. PhD thesis, Universidade Federal de Minas Gerais.

- Gao, J. L. (2002). An adaptive network/routing algorithm for energy efficient cooperative signal processing in sensor networks. In *IEEE Aerospace Conference*, volume 3, pp. 3–1117 – 3–1124. Citeseer.
- Garlan, D.; Siewiorek, D.; Smailagic, A. e Steenkiste, P. (2002). Project aura: Toward distraction-free pervasive computing. *IEEE Pervasive Computing*, pp. 22 – 31.
- Han, C.-C.; Kumar, R.; Shea, R.; Kohler, E. e Srivastava, M. (2005). A dynamic operating system for sensor nodes. In *MobiSys '05: Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pp. 163–176. ACM.
- Heinzelman, W.; Chandrakasan, A. e Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, p. 10.
- Held, G. e Marshall, T. R. (1996). *Data and image compression: Tools and Techniques*. John Wiley & Sons, Inc. New York, NY, USA, 4 edição.
- Hill, J.; Szewczyk, R.; Woo, A.; Hollar, S.; Culler, D. e Pister, K. (2000). System architecture directions for networked sensors. *ACM Sigplan Notices*, 35(11):104.
- Hoang, A. T. e Motani, M. (2005). Collaborative broadcasting and compression in cluster-based wireless sensor networks. In *Proceedings of the Second European Workshop on Wireless Sensor Networks*, pp. 197 – 206. IEEE CNF.
- Hua, G. e Chen, C. W. (2005). Distributed source coding in wireless sensor networks. In *Second International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks*, p. 7.
- Ju, H. e Cui, L. (2005). Easipc: a packet compression mechanism for embedded wsn. *11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pp. 294 – 399.
- Kimura, N. e Latifi, S. (2005). A survey on data compression in wireless sensor networks. In *International Conference on Information Technology: Coding and Computing*, pp. 8 – 13.
- Krishnamachari, L.; Estrin, D. e Wicker, S. (2002). Impact of data aggregation in sensor networks. In *Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops*, pp. 575 – 578.
- Lin, S.; Gunopulos, D.; Kalogeraki, V. e Lonardi, S. (2005). A data compression technique for sensor networks with dynamic bandwidth allocation. In *12th International Symposium on Temporal Representation and Reasoning*, pp. 186 – 188.
- Madden, S.; Franklin, M. J.; Hellerstein, J. M. e Hong, W. (2002). Tag: a tiny aggregation service for ad-hoc sensor networks. *ACM SIGOPS Operating Systems Review*, p. 146.

- Maniezzo, D.; Mazzini, G. e Yao, K. (2002). Energetic trade-off between computing and communication resource in multimedia surveillance sensor network. *Proceedings of the 4th International Workshop on Mobile and Wireless Communications Networks*, pp. 373 – 376.
- Mini, R. A. F.; Val Machado, M.; Loureiro, A. A. F. e Nath, B. (2005). Prediction-based energy map for wireless sensor networks. *Ad Hoc Networks*, 3(2):235–253.
- Nakamura, E. F.; Figueiredo, C. M. S. e Loureiro, A. A. F. (2005a). Information fusion for data dissemination in self-organizing wireless sensor networks. *Networking-ICN*, pp. 585–593.
- Nakamura, F. G.; Quintão, F. P.; Menezes, G. C. e Mateus, G. R. (2005b). An optimal node scheduling for flat wireless sensor networks. *Proceedings of the IEEE International Conference on Networking*, pp. 475–482.
- Noury, N.; Herve, T.; Rialle, V.; Virone, G.; Mercier, E.; Morey, G.; Moro, A. e Porcheron, T. (2000). Monitoring behavior in home using a smart fall sensor and position sensors. In *Proceedings of the 1st Annual International Conference On Microtechnologies in Medicine and Biology*, pp. 607 – 610.
- NS-2 (2004). *The Network Simulator ns-2*. Disponível em <http://www.isi.edu/nsnam/ns/>.
- Pattam, S.; Krishnamachari, B. e Govindan, R. (2004). The impact of spatial correlation on routing with compression in wireless sensor networks. In *Proceedings of the Third International Symposium on Information Processing in Sensor Networks*, pp. 28 – 35. Association for Computing Machinery, Inc, One Astor Plaza, 1515 Broadway, New York, NY, 10036-5701, USA.
- Petrovic, D.; Shah, R. C.; Ramchandran, K. e Rabaey, J. (2003). Data funneling: routing with aggregation and compression for wireless sensor networks. *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 156 – 162.
- Petrovic, J. C. D. e Ramchandran, K. (2003). A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks. In *INFOCOM*, pp. 1054 – 1062.
- Pradhan, S. S.; Kusuma, J. e Ramchandran, K. (2002). Distributed compression in a dense microsensor network. *IEEE Signal Processing Magazine*, 19(2):51 – 60.
- Priyantha, N. B.; Chakraborty, A. e Balakrishnan, H. (2000). The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 32 – 43. ACM.
- Rabaey, L.; Ammer, J. e Silva, J. L. (2000). Pico-radio: ad-hoc wireless networking of ubiquitous low energy sensor/monitor nodes. In *Proceedings of the IEEE Computer Society Annual Workshop on VLSI*, pp. 9 – 12.

- Ruiz, L. (2003). *Maná: uma arquitetura para gerenciamento de redes de sensores sem fio*. PhD thesis, Universidade Federal de Minas Gerais.
- Sankarasubramaniam, Y.; Akan, O. B. e Akyildiz, I. F. (2003). Esrt: Event-to-sink reliable transport in wireless sensor networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking computing*, pp. 177 – 188. ACM.
- Scaglione, A. e Servetto, S. (2005). On the interdependence of routing and data compression in multi-hop sensor networks. *Wireless Networks*, 11(1-2):149 – 160.
- Shah, R. C. e Rabaey, J. (2002). Energy aware routing for low energy ad hoc sensor networks. In *IEEE Wireless Communications and Networking Conference*, volume 1, pp. 350–355. Citeseer.
- Slepian, D. e Wolf, J. (1973). Noiseless coding of correlated information sources. *IEEE Transactions on information Theory*, 19(4):471–480.
- Vasudevan, L. e Ortega, A. (2002). Processing-aware compression for sensor networks. In *Proceedings of the 2nd IEEE Sensor Array and Multichannel Signal Processing*, pp. 239 – 243. Citeseer.
- Wan, C.-Y.; Campbell, A. T. e Krishnamurthy, L. (2002). Psfq: A reliable transport protocol for wireless sensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, p. 11. ACM.
- Watkins, C. e Dayan, P. (1992). Q-learning. In *Machine Learning*, volume 8, pp. 279–292. Springer.
- Woo, A. e Culler, D. (2001). A transmission control scheme for media access in sensor networks. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pp. 221 – 235. ACM.
- Wyner, A. D. e Ziv, J. (1976). The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 22:1 – 10.
- Zhang, H. e Hou, J. (2005). Maintaining sensing coverage and connectivity in large sensor networks. *Wireless ad hoc and Sensor Networks*, 14:89–123.