

PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



ROTEAMENTO GEOGRÁFICO E DESVIO DE BURACOS USANDO *SINKS* DE ALTO ALCANCE EM REDES DE SENSORES SEM FIO

MOYSÉS MENDES DE LIMA

Manaus - AM
Fevereiro de 2019

MOYSÉS MENDES DE LIMA

ROTEAMENTO GEOGRÁFICO E DESVIO DE BURACOS USANDO *SINKS* DE ALTO ALCANCE EM REDES DE SENSORES SEM FIO

Tese apresentada ao Programa de Pós-Graduação em Informática do Instituto de Computação da Universidade Federal do Amazonas como requisito parcial para a obtenção do grau de Doutor em Informática.

ORIENTADOR: HORÁCIO A. B. FERNANDES DE OLIVEIRA

Manaus - AM

Fevereiro de 2019

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Lima, Moysés Mendes de
L732r Roteamento Geográfico e Desvio de Buracos usando Sinks de Alto Alcance em Redes de Sensores sem Fio / Moysés Mendes de Lima. 2019
117 f.: il. color; 31 cm.

Orientador: Horácio Antônio Braga Fernandes de Oliveira
Tese (Doutorado em Informática) - Universidade Federal do Amazonas.

1. Buracos de Roteamento. 2. Roteamento Geografico. 3. Redes de Sensores Sem Fio. 4. Mínimos Locais. I. Oliveira, Horácio Antônio Braga Fernandes de II. Universidade Federal do Amazonas III. Título

FOLHA DE APROVAÇÃO

"Evitando Regiões de Buraco de Roteamento em Redes de Sensores sem Fio"

MOYSÉS MENDES DE LIMA

Tese de Doutorado defendida e aprovada pela banca examinadora constituída pelos Professores:

Prof. Horácio Antonio B. Fernandes de Oliveira - PRESIDENTE

Prof. Edjair Souza Mota - MEMBRO INTERNO

Prof. Eduardo James Pereira Souto - MEMBRO INTERNO

Prof. Carlos Maurício Seródio Figueiredo - MEMBRO INTERNO

Prof. Leandro Nelinho Balico - MEMBRO EXTERNO

Prof. Éfren Lopes de Souza - MEMBRO EXTERNO

Manaus, 07 de Fevereiro de 2019

Dedico este trabalho ao meu Senhor e Salvador Jesus Cristo.

Agradecimentos

Agradeço em primeiro lugar ao meu Senhor Jesus Cristo, que esteve ao meu lado durante a caminhada, renovando minhas forças quando eu pensava em desistir e me surpreendia a cada manhã com sua bondade e o seu amor infinito. A Ele a honra, a glória, a força e o poder para todo o sempre.

Ao meu estimado orientador, Professor Horácio Fernandes, por acreditar neste projeto desde o mestrado e aceitar me orientar pela segunda vez, agora no doutorado. Por me proporcionar conforto emocional e oferecer seus préstimos com paciência e honestidade. Por seus esclarecimentos e constante ajuda nas horas de crise, tornando possível a realização deste trabalho. Agradeço a Deus por sua vida, professor.

Aos Professores Dr. Eduardo James Pereira Souto, Dr. Edjair de Souza Mota, Dr. Maurício Figueiredo, Dr. Leandro Nelinho Balico e Dr. Éfren Lopos de Souza, por aceitarem avaliar e contribuir com ricas orientações esta proposta de doutorado.

Aos Professores Dr. Raimundo Barreto e Dr. Eduardo Nakamura, pelo apoio e incentivo em momentos decisivos desta jornada.

Aos companheiros de sala, do Laboratório de Computação Móvel e Ubíqua (LCMU) pela companhia diária, pelos momentos descontraídos e pelas contribuições prestadas.

Aos meus pais, Moysés Xavier e Maria das Graças, pelo apoio, carinho e pelo incentivo aos estudos.

Ao meu único irmão Rildo Lima, pelo companheirismo e palavras de força.

À Primeira Igreja Batista de Manaus, representada pelos pastores Norton Lages e Rodrigo Soares, pelas orações, palavras de conforto e auxilio espiritual em momentos decisivos.

Ao meu querido filho primogênito Pedro Lima, que por muitas noites teve que jogar sozinho e faltar aos treinos de *Jiu-Jitsu* mas ainda assim, com um sorriso no rosto dizia:"*Está tudo bem, pai. Quando você concluir o doutorado nós voltamos!*". Pelos abraços que recebia dele enquanto me prendia a telas de computador e esquemas gráficos. Por sua gentileza sem fim. Amo você meu filho.

Ao meu filho caçúla Benjamin, enviado a nós como um presente de Deus durante o

processo de doutoramento. Pelos sorrisos que recebia dele quando chegava em casa tarde da noite, após um perĂodo de estudos na UFAM. Pelo toque especial que ele deu a este doutorado, me fazendo desligar do ambiente de infindáveis códigos e textos. Amo você meu filho.

Finalmente, à minha amada esposa Etyanne, fiel companheira, auxiliadora idônea e parceira. Por seu apoio incondicional e seu correto proceder diante das crises. Por não se deixar entristecer nas dificuldades e da exigência de tempo de trabalho. Pela paciência e incontáveis demonstrações de amor e cuidado. Pelo amparo gentil e sem reservas dispensado a mim ao longo de toda esta jornada. Você é incrível meu amor. Amo você.

"The craving of a sluggard will be the death of him, because his hands refuse to wor."
(Proverbs 21:25 - From NVI Bible)

Resumo

A ocorrência de regiões de buraco em Redes de Sensores Sem Fio é um importante desafio a ser superado pois afeta a utilização de protocolos de roteamento no tocante à transmissão de mensagens através da rede. Em algoritmos de roteamento geográfico, o surgimento de buracos de roteamento é comumente creditado ao fenômeno dos mínimos locais (*local minimum*), caracterizado por uma região na grade de sensoreamento onde os nós estão impossibilitados de encaminhar mensagens a um destino pretendido. Sua ocorrência pode ser causada por obstáculos físicos, esgotamento energético, falhas na comunicação entre vizinhos eleitos, implantação incorreta. A maioria das soluções utilizadas para lidar com o problema das regiões de buraco prefere abandonar o mecanismo tradicional de encaminhamento guloso para adotar temporariamente o esquema de roteamento através de perímetro, aplicado aos nós próximos ou ao longo da borda de um buraco. Entretanto, essa estratégia exige que todos os nós conheçam sua localização na grade, o que nem sempre é possível. Além disso, esse tipo de abordagem impõe tráfego excessivo aos nós da borda do buraco e seus vizinhos de um salto de distância, tornando-os cada vez mais utilizados quando se pretende encaminhar pacotes para fora da região de buracos.

Nesta tese de doutorado, propomos quatro (04) soluções de roteamento geográfico para lidar com o problema dos buracos de roteamento em redes de sensores sem fio. Os algoritmos ARESTA, REACT, LRS e BYPATH, foram concebidos para serem capazes de criar caminhos válidos entre a origem e o destino dos pacotes sem fazer uso de coordenadas virtuais, sistemas de posicionamento global (GPS) ou qualquer outro sistema de coordenadas que normalmente gera custos energéticos adicionais e nem sempre estão disponíveis. Nossos algoritmos tiram vantagem do alcance de comunicação do *sink*, equipado com um dispositivo de comunicação de maior potência, de tal forma que em um único salto todos os nós da rede sejam alcançados. O retorno dos nós sensores para o nó *sink* é realizado através de múltiplos saltos, utilizando os valores do RSSI (*Received Signal Strength Indicator*) para calcular o próximo salto em direção ao nó *sink*, calcular o tempo de espera para o envio de de pacotes múltiplos saltos e em alguns casos possibilitar a agregação de dados enquanto desvia buracos de roteamento durante o processo de encaminhamento. Nossas soluções de

desvio de buracos estão aptas a gerar fluxos qualificados para o escoamento de pacotes entre a origem e o destino, mantendo vantagens como altas taxas de entrega de pacotes e economia de energia, observadas em algoritmos clássicos de roteamento geográfico.

Palavras-chave: Buracos de Roteamento; Roteamento Geográfico; Redes de Sensores Sem Fio

Abstract

The occurrence of hole regions in Wireless Sensor Networks is a significant challenge to overcome because it affects the use of geographic routing protocols regarding the transmission of messages through the network. Holes are usually associated with regions where nodes are unable to forward its information received. Its occurrence can be caused by physical obstacles, energy depletion, communication failures between neighbors elected, incorrect implantation and due to the local minimum phenomenon. Most of the solutions used to deal with the hole region problem prefer to abandon the traditional greedy forwarding mechanism to temporarily adopt the perimeter routing scheme applied to nearby nodes or along the edge of a hole. However, this strategy requires that all nodes know their location in the sensing grid, which is not always possible in forest scenarios. Also, this type of approach imposes excessive traffic to the edge nodes of the hole and its neighbors a leap away, making them increasingly used when it is intended to route packets out of the region of holes.

In this thesis proposal, we propose four (04) geographic routing solutions to deal with the problem of routing holes in Wireless Sensor Networks. The ARESTA, REACT, LRS, and BYPATH algorithms are designed to be able to create valid paths between the source and destination of the packets without using virtual coordinates, global positioning systems (GPS), or any other coordinate system that usually generates additional energy costs and are not always available. Our algorithms take advantage out of the communication range of the sink, equipped with a communication device of a higher power, in such a way that in a single jump all the nodes of the network are reached. The return of the sensor nodes to the node is done through multiple jumps using our proposed approach that uses the RSSI (Received Signal Strength Indicator) values to calculate the next hop toward the node sink, figure the wait time for sending multiple hop packets, and still enable data aggregation while diverting routing holes during the routing process. Our hole diversion solutions are capable of generating qualified flows for the flow of packages between source and destination, maintaining advantages such as high packet delivery rates and energy savings, observed in classical geographic routing algorithms.

Keywords: Routing Holes; Geographic Routing; Wireless Sensor Networks

Lista de Figuras

1.1 Valores de RSSI decrescendo à medida que a consulta do *sink* de alto alcance se propaga na rede. Nós mais distantes do *sink* receberão a mensagem com potência menor. 3

1.2 Torre do projeto ATTO, implementado na Amazônia. 4

1.3 Primeiro estágio de funcionamento dos algoritmos propostos 10

2.1 Mecanismo de encaminhamento guloso. A cada passo, o pacote é repassado para o vizinho mais geograficamente próximo do nó *sink*. Baseado em [Park et al., 2010]. 19

2.2 Exemplo de agregação de dados. Nós *A*, *B*, *C* e *D*, repassando informações agregadas. 24

3.1 Exemplo da ocorrência de um buraco entre o nó *x* e o destino *D*. Baseado em [Karp & Kung, 2000]. 30

3.2 Algoritmo GEAR aprendendo caminhos ao longo da região de buraco. Os nós na cor preta (*G, H, I*) tiveram esgotamento energético e por isso não pode receber pacotes. [Yu et al., 2001]. 31

3.3 Regra local de detecção de uma região de buracos. O círculo sólido representa o alcance da comunicação do nó *x*. O círculo pontilhado representa a circunferência $\Delta\, uxv$. Baseado em [Fang et al., 2004]. 32

3.4 Desvio de buraco usando posicionamento virtual. (a) Nó 12 simulando um buraco tanto para Greedy quanto para MFR. (b) Caminho que pode ser encontrado através do algoritmo MFR-ViP. (c) Caminhos em volta do buraco. Baseado em [You et al., 2009b]. 34

3.5 Etapa de aproximação executada a partir das informações dos nós da borda do buraco. A linha vermelha representa a borda do polígono aproximado. Baseado em [Le et al., 2012]. 36

4.1 (a) Nó da região de buracos não possui vizinho elegível a próximo salto e é marcado como nó preso. (b) Após temporizador expirar nós encaminham suas pacotes de dados a vizinhos elegíveis. (c) Regra de desvio aplicada: Conforme a configuração da grade, o nó irá optar por um vizinho que supre as exigências para o encaminhamento 40
4.2 Cenário base de implementação de todos os algoritmos descritos nesta tese. . . 44
4.3 Resultados obtidos ao se avaliar a taxa de entrega. 45
4.4 Resultados obtidos ao se avaliar as escolhas de próximo salto. 47

5.1 (a) Cada nó envia uma mensagem de *broadcast* após temporizador expirar. O nó preso (em vermelho) detecta que está em uma região de buracos, pois não possui vizinhos mais próximos ao *sink*. (b) Início do processo de transmissão às cegas após detecção de buraco. (c) Fluxo gerado a partir de nossa abordagem oportunista. 50
5.2 Resultados obtidos ao se avaliar a taxa de entrega de pacotes e quantidade de pacotes transmitidos em relação ao alcance da comunicação 55
5.3 Resultados obtidos ao se avaliar as decisões geográficas de próximo salto, a taxa de entrega em relação ao alcance da comunicação e à imprecisão do RSSI, além dos pacotes transmitidos em relação ao alcance da comunicação 56

6.1 (a) Após a consulta única do *sink*, nós próximos ao *sink* (vizinhos de um salto de distância) elegem a si como salto geográfico (*nós em cor preta*). Durante a etapa de estimativa de distância, essa informação de vizinho real do *sink* é divulgada. (b) Pacotes que tendem ao buraco serão encaminhados ao nó preso. (c) Rota alternativa gera a a partir da divulgação de informações dos nós vizinhos ao *sink*. 61
6.2 Resultados obtidos ao se avaliar a taxa de entrega. 65
6.3 Resultados obtidos ao se avaliar a quantidade de pacotes transmitidos. 66
6.4 Resultados obtidos ao se avaliar a taxa de agregação de dados, os erros na escolha de próximo salto e a quantidade de transmissões na rede. 66

7.1 (a) Nós próximos ao *sink* se auto-elegem como salto geográfico (*nós em cor preta*) no momento em que recebem a consulta do *sink*. Durante a etapa de estimativa de distância, essa informação de vizinho real do *sink* é divulgada. (b) Nó da região de buracos também divulga sua distância estimada aos vizinhos, mas sem a informação de rota até o *sink*. (c) Nós irão escolher o vizinho mais próximo ao destino, optando pela rota gerada a partir dos vizinhos do *sink*. . . . 70
7.2 Fluxo de Pacotes: Grade com 512 nós. 74
7.3 Fluxo de Pacotes: Grade com 1152 nós. 74

7.4 Resultados obtidos ao se avaliar a taxa de entrega e a os erros na escolha do próximo salto. 75
7.5 Resultados obtidos ao se avaliar a quantidade de pacotes transmitidos. 76
7.6 Resultados obtidos ao se avaliar a quantidade de saltos e o tamanho médio dos fluxos gerados até o *sink*. 78
7.7 Qualidade dos caminhos de fluxo gerados pelas três abordgens, em uma grade com 512 nós. 79
7.8 Qualidade dos caminhos de fluxo gerados pelas três abordgens, em uma grade com 1152 nós. 79

Lista de Tabelas

1.1 Resumo dos cenários envolvendo torres altas, aplicáveis à esta tese. 6
1.2 Resumo das principais características de cada abordagem 13

2.1 Classificação dos Protocolos de Roteamento 18

3.1 Classificação das abordagens apresentas 38

4.1 Configuração do cenário de simulação do algoritmo ARESTA. 44

5.1 Configuração do cenário de simulação do algoritmo REACT. 53

6.1 Configuração do cenário de simulação do algoritmo LRS. 63

7.1 Configuração do cenário de simulação do algoritmo BYPATH. 74

Acrônimos

AoA Ângulo de Chegada (*Angle of Arrival*)

ATTO Torre Alta de Observação Amazônica (*Amazonian Tall Tower Observatory*)

DoS Negação de Serviços (*Denial of Service*)

EEDRS An energy efficient and load balanced distributed routing scheme for wireless sensor networks with holes

GEAR Geographical and Energy Aware Routing

GF Mecanismo de encaminhamento guloso (*Greedy Forwarding*)

GF–ViP Greedy Forwarding with Virtual Position

GPSR Greedy Perimeter Stateless Routing for Wireless Networks

HDAR Hole Detection and Adaptive Geographical routing

HORA A Distributed Coverage Hole Repair Algorithm for Wireless Sensor Networks

HVP Greedy forwarding with Virtual Hierarchical Position

IoT Internet das Coisas (*Internet of Things*)

LQI Indicador de Qualidade do Enlace (*Link Quality Indication*)

MAC Camada de Acesso ao Meio (*Media Access Control Layer*)

MICAz Módulo de 2.4GHz que permite comunicação sem fio entre sensores com baixo consumo de energia

QoS Qualidade de Serviços (*Quality of Service*)

QUDG Quasi Unit Disk Graph

RSSF Redes de Sensores Sem Fio

RSSI Indicator de Potência do Sinal Recebido (*Received Signal Strength Indicator*)

SINR Interferência do Sinal de Rádio com Ruído (*Signal to Interference Noise Ratio*)

TDoA Diferença do Tempo de Chegada (*Time Difference of Arrival*)

ToA Tempo de Chegada (*Time of Arrival*)

UDG Unit Disk Graph

VCap Virtual Coordinate assignment protocol

Sumário

Agradecimentos **ix**

Resumo **xiii**

Abstract **xv**

Lista de Figuras **xvii**

Lista de Tabelas **xix**

Acrônimos e Nomenclaturas **xx**

1 Introdução **1**
1.1 Motivação 3
1.2 Declaração do Problema 6
1.3 Hipóteses da pesquisa e abordagem Proposta 7
1.4 Objetivos 11
1.4.1 Objetivo Geral 11
1.4.2 Objetivos Específicos 12
1.5 Principais contribuições 12
1.5.1 Comentário sobre publicações e submissão das contribuições 13
1.5.2 Principal contribuição de pesquisa 14
1.6 Organização desta Tese de Doutorado 14

2 Fundamentos Teóricos **17**
2.1 Visão Geral sobre Protocolos de Roteamento 17
2.2 Roteamento Geográfico 19
2.3 Regiões de Buraco 20
2.3.1 Buracos de Cobertura 21

2.3.2 Buracos de Roteamento . . . 22
2.3.3 Buracos Causados por Falhas ou Bloqueios na Comunicação . . . 22
2.3.4 Buracos Causados por Agentes Maliciosos . . . 23
2.4 A Fusão de Dados em RSSF . . . 23
2.4.1 Agregação de Dados . . . 24
2.5 Estimativa de Distância Baseada no RSSI . . . 25
2.6 Definição do Problema . . . 27

3 Trabalhos Relacionados **29**
3.1 Introdução . . . 29
3.1.1 Greedy Perimeter Stateless Routing for Wireless Networks (GPSR) . . . 30
3.1.2 Geographical and Energy Aware Routing (GEAR) . . . 30
3.1.3 Locating and Bypassing Routing Holes in Sensor Networks (BOUNDHOLE) 31
3.1.4 Virtual Coordinate Assignment Protocol (VCap) . . . 32
3.1.5 Greedy Forwarding with Virtual Position (GF–ViP) . . . 33
3.1.6 Hole Detection and Adaptive Geographical Routing (HDAR) . . . 33
3.1.7 A Distributed Coverage Hole Repair Algorithm for Wireless Sensor Networks (HORA) . . . 35
3.1.8 Void Aware in Geographic Routing for Wireless Sensor Networks (VAGR) . 35
3.1.9 An Energy Efficient and Load Balanced Distributed Routing Scheme for Wireless Sensor Networks with Holes . . . 36
3.1.10 An Enhanced Location–Free Greedy Forward Algorithm with Hole Bypass Capability in Wireless Sensor Networks (RSSR) . . . 37
3.2 Classificação dos Trabalhos Apresentados . . . 38
3.3 Comentários Finais . . . 38

4 Agregação de Dados com Desvio de Buracos para Redes de Sensores Sem Fio com *Sinks* de Alto Alcance **39**
4.1 Algoritmo ARESTA . . . 40
4.2 Avaliação de Desempenho . . . 43
4.2.1 Metodologia . . . 43
4.2.2 Impacto da Escalabilidade da Rede . . . 46
4.2.3 Impacto do Raio de Comunicação . . . 46
4.2.4 Impacto da Densidade da Rede . . . 46
4.2.5 Impacto da Imprecisão do RSSI a Curta Distância . . . 47
4.2.6 Impacto da Regra de Desvio de Buracos na Escolha do Próximo Salto. . . . 47
4.3 Considerações finais . . . 48

5 Roteamento Geográfico Oportunista e Agregação de Dados com Contorno de Buracos em Redes de Sensores Sem Fio **49**
5.1 Algoritmo REACT 49
5.2 Avaliação de Desempenho 53
5.2.1 Metodologia 53
5.2.2 Impacto da Escalabilidade da Rede 54
5.2.3 Impacto do Raio de Comunicação 55
5.2.4 Impacto da Densidade da Rede 56
5.2.5 Impacto da Imprecisão do RSSI a Curta Distância 57
5.3 Considerações finais 57

6 Usando *Sinks* de Longo Alcance para contornar Buracos de Roteamento em Redes de Sensores Sem Fio **59**
6.1 Algoritmo LRS 60
6.2 Avaliação de Desempenho 62
6.2.1 Metodologia 63
6.2.2 Impacto da Densidade da Rede 64
6.2.3 Impacto da Imprecisão do RSSI a Curta Distância 64
6.2.4 Impacto da Escalabilidade da Rede 65
6.2.5 Impacto do Raio de Comunicação dos Nós 67
6.3 Considerações finais 67

7 Evitando Buracos de Roteamento em Redes de Sensores Sem Fio com *Sinks* de Alto Alcance **69**
7.1 Algoritmo BYPATH 70
7.2 Avaliação de Desempenho 72
7.2.1 Metodologia 73
7.2.2 Impacto da Densidade da Rede 75
7.2.3 Impacto da Imprecisão do RSSI a Curta Distância 76
7.2.4 Impacto da Escalabilidade da Rede 76
7.2.5 Impacto do Raio de Comunicação dos Nós 77
7.2.6 Qualidade dos Fluxos de Escoamento de Pacotes 77
7.3 Considerações finais 80

8 Conclusões **81**
8.1 Considerações Finais 81
8.2 Trabalhos Futuros 82

Referências Bibliográficas **83**

Capítulo 1

Introdução

Uma Rede de Sensores sem Fio (RSSF) consiste em um conjunto de nós sensores depositados em uma área de interesse que de forma colaborativa monitoram condições ambientais físicas e químicas em diferentes locais [Akyildiz et al., 2002b]. Por esta razão, as RSSFs são essenciais no monitoramento de condições ambientais tais como temperatura, luminosidade, movimento e presença de certos tipos de objetos. Na maior parte das aplicações, os nós sensores possuem limitações energéticas e de largura de banda, fazendo com que técnicas de otimização do consumo de energia e comunicação sejam necessárias.

Quando consideramos o número de saltos entre nós regulares e o nó *sink* em uma RSSF, podemos classificar as abordagens como redes de único salto e redes de múltiplos saltos [Akyildiz et al., 2002a]. Na primeira classificação, os nós regulares enviam suas informações diretamente ao *sink*, simplificando a implementação de soluções de envio de pacotes. Entretanto, é necessária a utilização de um dispositivo de comunicação sem fio de longo alcance e consequentemente de maior consumo energético. Já na abordagem em múltiplos saltos, os nós regulares enviam suas informações ao *sink* utilizando comunicação de curto alcance estabelecida com um ou mais vizinhos em localização intermediária.

Uma das limitações encontradas em RSSF é o alcance da comunicação dos nós no campo de sensoreamento, uma vez que maioria dos dispositivos possui alcance máximo de 100 metros [Piotrowski et al., 2012]. Neste sentido, diversas aplicações em redes de sensores já demandam a comunicação de longo alcance já que os nós estão separados por grandes distâncias, característica que possibilita o monitoramento de uma grande área geográfica. Atualmente, a maioria das soluções de roteamento empregadas utiliza a comunicação em múltiplos saltos. Contudo, em aplicações que requerem monitoramento regular e fazem uso de um grande número sensores densamente implantados e distantes do *sink*, a utilização da comunicação em múltiplos saltos pode ser considerada como um fator limitante, pois exige grande cobertura a partir dos nós que normalmente cobrem apenas algumas centenas de metros. Neste sentido, soluções que exploram a comunicação de único salto em RSSF tem

se mostrado atraentes quando associadas ao grande alcance de comunicação dos nós [Willis & Kikkert, 2005, Zennaro et al., 2010].

Segundo Piotrowski et al. [2012], a comunicação de longo alcance entre nós em uma RSSF é raramente considerada, pois neste tipo de cenário são consideradas características energéticas específicas e inerentes à cada abordagem. Além disso, a transmissão de dados ainda depende de uma variedade de parâmetros que interferem diretamente na eficiência da solução. De acordo com Singh et al. [2010], em uma RSSF densamente implantada as limitações energéticas se tornam ainda mais evidentes, o que requer a implementação de um conjunto de controles de rede e funcionalidades adicionais como sincronização, localização dos nós, agrupamento (*clustering*) e segurança. Neste sentido, os protocolos de roteamento tradicionais possuem algumas deficiências que devem ser tratadas na fase de implementação, levando em conta as características singulares das RSSFs e os desafios presentes no desenvolvimento de aplicações.

Os protocolos de roteamento geográficos têm sido vistos como uma das principais soluções de roteamento em Redes de Sensores Sem Fio (RSSFs) por serem escaláveis, dinâmicos e possuírem uma alta taxa de entrega de dados [Kim et al., 2005, You et al., 2009a]. Contudo, em algumas situações os métodos envolvendo informações geográficas não podem ser utilizados, especialmente em cenários onde a informação de localização não está disponível ou não existe [Éfren Souza et al., 2012]. Neste sentido, o encaminhamento guloso é a forma mais simples de se implementar o roteamento geográfico e é particularmente atraente nas redes de sensores. A técnica de roteamento geográfico utiliza uma estratégia simples de geração de rotas em que um nó sensor sempre encaminha um pacote ao vizinho geograficamente mais próximo ao nó que representa o destino final de um pacote, neste caso um *sink*.

Algoritmos de roteamento geográfico se destacam por agregar à implementação vantagens como escalabilidade, eficiência energética e baixo custo em termos de confecção de rotas [Boukerche et al., 2008, Karp & Kung, 2000, Leong et al., 2007]. Cada nó conhece a posição dos seus vizinhos, a sua posição e o destino da mensagem a ser encaminhada, normalmente em múltiplos saltos. Entretanto, o correto funcionamento de tais algoritmos requer a execução prévia de um sistema de atribuição de coordenadas para a obtenção das posições dos nós, o que nem sempre está disponível em cenários ubíquos. Além disso, a ocorrência de buracos de roteamento durante o encaminhamento das mensagens ao *sink* pode ser considerada como um dos maiores desafios para o funcionamento da abordagem, já que o mecanismo clássico de encaminhamento guloso não é capaz de superar regiões de buracos sem o auxílio de mecanismos de roteamento por perímetro. Os buracos estão normalmente associados a regiões onde os nós estão impossibilitados de encaminhar adiante informações. Nestes casos, a incapacidade de transmissão das mensagens pode ser creditada à ausência de

comunicação com o próximo salto ou ainda ao esgotamento energético de um ou mais nós envolvidos na comunicação [Fang et al., 2004].

Avanços na área de comunicação sem fio demonstram que a propagação do sinal de longo alcance se mostra confiável e pode ser utilizada para estimar distâncias em uma RSSF. Tal estimativa é baseada nos valores de RSSI (Received Signal Strength Indicator) dos nós envolvidos na comunicação, considerando a relação entre a força do sinal recebido e a distância como uma função de distância entre o emissor e o receptor [Al Alawi, 2011]. Conforme discutido por Lima et al. [2013b], um cenário aplicável à implementação destas tecnologias é o projeto ATTO (Amazonian Tall Tower Observatory), que consiste em uma torre de 320 m de altura e alta capacidade de comunicação, cujo objetivo é mapear eventos e fornecer medidas confiáveis de gases do efeito estufa [Tollefson, 2010]. O sinal emitido pelo dispositivo de comunicação instalado na torre poderia alcançar um grande número de nós estrategicamente implantados nas proximidades deste dispositivo emissor. Neste sentido, a estratégia utilizada pelos algoritmos de roteamento geográfico pode ser utilizada para proporcionar aos nós regulares o encaminhamento de pacotes de dados a este *sink* de grande capacidade em múltiplos saltos, utilizando para tanto os valores de RSSI percebidos pelos nós dispostos na grade. Por fim, aplicações que tirem proveito dessa arquitetura de funcionamento podem ser facilmente implementadas como alternativa ao modelo convencional de comunicação, em que o nó *sink* e os nós regulares possuem o mesmo alcance de comunicação.

1.1 Motivação

A utilização de abordagens baseadas em roteamento geográfico é especialmente atrativa em redes sem fio alocadas em grandes áreas e que utilizem o esquema de comunicação em múltiplos saltos. A abordagem permite que cada nó encaminhe pacotes ao próximo salto.

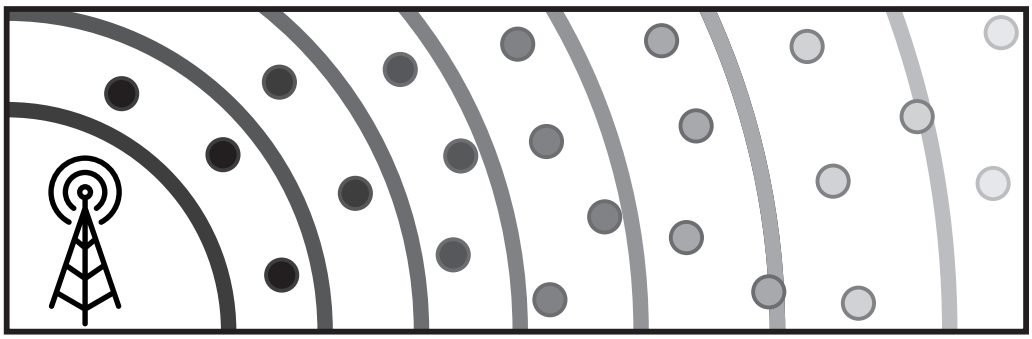


Figura 1.1. Valores de RSSI decrescendo à medida que a consulta do *sink* de alto alcance se propaga na rede. Nós mais distantes do *sink* receberão a mensagem com potência menor.

O encaminhamento guloso pode garantir a entrega dos pacotes e alcançar o mesmo desempenho dos algoritmos de caminho mínimo. Segundo [Seada et al., 2004], o encaminhamento guloso pode ser um método eficaz para reduzir o custo de entrega de pacotes quando assumimos que a densidade da rede é suficiente, a localização dos nós é precisa e a comunicação entre os nós é confiável, independente do alcance do sinal. Um dos grandes desafios na utilização de abordagens geográficas de encaminhamento de mensagens é a ocorrência de regiões de buraco durante o envio dos pacotes ao *sink*.

Visando sobrepor as limitações identificadas, propomos quatro algoritmos de roteamento geográfico, livres de sistemas de coordenadas e capazes de lidar com buracos de roteamento em Redes de Sensores Sem Fio (RSSF). Os algoritmos foram assim nomeados: ARESTA, REACT, LRS e BYPATH. As soluções foram concebidas para tirar proveito do grande alcance do nó *sink*, equipado com um dispositivo de comunicação de maior potência, e de tal forma que em um único salto todos os nós da rede sejam alcançados. Em cada uma das abordagens, o retorno dos nós sensores para o nó *sink* é realizado através de múltiplos saltos, utilizando nossa nova técnica de encaminhamento guloso que utiliza os valores do RSSI (*Received Signal Strength Indicator*) para calcular o próximo salto em direção ao nó *sink*, calcular o tempo de espera que possibilita a agregação de dados e ainda detectar e contornar regiões de buraco durante o processo de encaminhamento.



Figura 1.2. Torre do projeto ATTO, implementado na Amazônia.

A motivação para o desenvolvimento desta tese, surge a partir da implementação do projeto ATTO (*Amazonian Tall Tower Observatory*), ilustrado na Figura 1.2, que consiste

em uma torre no meio da floresta amazônica com uma estrutura física de 325 m de altura e alta capacidade de comunicação, utilizada para mapear eventos climáticas e fornecer medidas confiáveis de fontes e sumidouros de gases de efeito estufa como CO_2, CH_4 e N_2O, disponível em [Tollefson, 2010]. A torre está localizada a cerca de 150 km ao norte da cidade de Manaus e é mantida por pesquisadores da Alemanha e do Brasil.

Além do projeto ATTO, outros cenários envolvendo torres altas podem ser beneficiados com nossa arquitetura de comunicação. Nos Estados Unidos da América do Norte, a Divisão de Monitoramento Global (GMD) da Administração Nacional Oceânica e Atmosférica realiza pesquisas direcionadas a três (03) grandes desafios, que incluem: (*i*) reações de gases de efeito estufa e do ciclo de carbono; (*ii*) mudanças na composição das nuvens, aerossóis e radiação de superfície e; (*iii*) recuperação do ozônio na camada estratosférica. A seguir, destacamos alguns projetos desenvildos pela GMD que utilizam estruturas de torres equipadas com sensores e outros dispositivos de monitoramento.

Localizado no condado do Colorado, o projeto BAO (*Boulder Atmospheric Observatory*) é utilizado para estudar a camada atmosférica limite da terra e prover ensaios para calibração de sensores atmosféricos. A iniciativa é mantida pela Divisão de Ciências Físicas do Laboratório de Pesquisas de Sistemas terrestres (ERSL) desde a década de 1970 e consiste em uma torre alta de 300 m de altura, equipada com sensores para o monitoramento de radiação solar, CO_2 e CO. As medições da torre incluem uma variedade de sistemas de sensoriamento remoto e entraram em pleno funcionamento em abril de 2007 [Kaimal & Gaynor, 1983, Xing-Sheng et al., 1983]

Em 1994 , a *WLEF tower* foi implementada nas proximidades da cidade de Park Falls, Winsconsin. Com uma estrutura física de 396 m de altura, a 472 metros acima do nível do mar, a torre foi concebida para avaliar o nível de carbono das florestas na região e examinar a influência das trocas envolvendo CO_2 na atmosfera. De acordo com Yi et al. [2000], o programa se propõe a analisar os efeitos das misturas envolvendo CO_2 dentro e acima da camada atmosféricam, sobre o continente.

Por fim, a SCT (*South Carolina Tower*), ou Torre da Carolina do Sul, provê medições contínuas de CO_2, CO e H_2O, com o objetivo de expandir a pesquisa do ciclo de carbono no *Savannah River Site*[1]. Com uma estrutura física de 305 m de altura, a torre é equipada com sensores e dispositivos de comunicação que tiram proveito a uma rede meteorológica já implementada na região que é operada pelo Laboratório Nacional do Rio Savannah [Werth et al., 2015].

A Tabela 1.1 sumariza os cenários discutidos. Em todos os casos, um *sink* de alta capacidade de comunicação poderia ser utilizado para alcançar nós sensores estrategicamente

[1]O SRS (Savannah River Site) é uma reserva nuclear localizada no estado da Carolina do Sul, Estados Unidos, construída durante a década de 1950 para refinar materiais nucleares de uso militar.

posicionados nas proximidades de cada torre.

Projeto/Programa	Altura da torre	Principais medições
BAO [Xing-Sheng et al., 1983]	300 m	CO_2, CH_4, H_2, SF_6 e N_2O
WLEF [Yi et al., 2000]	472 m	CO_2, CH_4 e CO
ATTO [Tollefson, 2010]	325 m	CO_2, CH_4 e N_2O
SCT [Werth et al., 2015]	305 m	CO_2, CH_4, H_2 e N_2O

Tabela 1.1. Resumo dos cenários envolvendo torres altas, aplicáveis à esta tese.

Ao longo desta tese, discutiremos o problema das regiões de buraco em redes de sensores sem fio, bem como soluções para lidar com sua ocorrência. A maior parte desta tese está focada em algoritmos de roteamento geográfico que utilizam o mecanismo de encaminhamento guloso (*greedy forwarding*) e ainda implementam soluções para detecção e desvio de regiões de buraco.

1.2 Declaração do Problema

Algoritmos de roteamento geográfico possuem vantagens particularmente importantes para redes de sensores sem fio no que tange à escalabilidade da rede, eficiência energética, carga de trabalho reduzida a partir dos nós regulares e baixo consumo de recursos de processamento e memória [Boukerche et al., 2008]. Neste sentido, os nós que participam da etapa de roteamento não necessitam gravar as informações sobre todos os vizinhos envolvidos na criação de rotas, pois a escolha de um vizinho apto a encaminhar pacotes é feita a partir das informações trocadas em um salto de distância. Essa característica singular, tem sido explorada por aplicações de monitoramento emergentes que optam por abordagens geográficas de comunicação em múltiplos saltos, a partir do conhecimento dos vizinhos de um salto de distância.

A técnica de encaminhamento guloso, que consiste na escolha de um vizinho, geograficamente mais próximo ao destino final da etapa de roteamento tem sido amplamente explorada por algoritmos de roteamento geográfico. Esse processo de escolha de um vizinho guloso, ou seja, o mais próximo possível ao destino final dos pacotes, permite a criação de caminhos de escoamento eficazes e facilitam a adoção de outras estratégias de repasse de informações como a agregação de dados e o agrupamento proposital de um conjunto de nós, técnica conhecida como *clustering*. Entretanto, a ocorrência de buracos de roteamento formados a partir do fenômeno dos mínimos locais (*local minimum*) é encarada com uma das principais limitações à correta implementação da abordagem geográfica por impedir o repasse de pacotes ao destino pretendido. A ocorrência dos mínimos locais se dá quando

não há vizinho de um salto de distância que esteja geografica mente mais próximo ao destino final dos pacotes. Nesta condição, o nó que deveria dar continuidade ao escoamento de pacotes fica "preso" em uma região de ausência de vizinhos, inviabilizando a utilização do mecanismo de encaminhamento guloso.

Grande parte dos algoritmos de roteamento geográfico se utiliza do mecanismo de encaminhamento guloso no processo de repasse de dados até o *sink*. Apesar de apresentarem resultados superiores aos encontrados em estratégias já conhecidas de roteamento como a difusão [Zhao et al., 2009] e o roteamento oportunista clássico [Biswas & Morris, 2004], abordagens geográficas não operam na presença de buracos de roteamento. Como já discutido por Karp & Kung [2000], as abordagens aptas a encaminhar pacotes na presença de buracos se utilizarão de mecanismos que explorem os limites da região de buracos, a fim de descobrir rotas válidas em torno do buraco que permitam o repasse de pacotes presos (ou impedidos). Entretanto, uma vez que os limites do buraco (bordas) tenham sido estabelecidos, o custo aplicado no processo de contorno pode levar ao esgotamento das reservas de energia entre os nós, ou perto, ou ao longo da borda.

Visando suprir as limitações já discutidas, propomos um conjunto de soluções capaz de lidar com a ocorrência dos mínimos locais a partir das informações coletadas de vizinhos de um salto de distância e da distância dos nós regulares em relação ao nó *sink*. Nossos algoritmos não exploram o perímetro da região de buracos e também não necessitam de informações sobre o posicionamento dos nós regulares na grade. Ainda, nossas soluções são baseadas nas abordagens de roteamento geográfico e roteamento oportunista. Esta última, é uma técnica de repasse de pacotes a vizinhos de um salto que aproveita mensagens de *broadcast*) para implementar parâmetros de auto-eleição do próximo salto de roteamento em direção a um destino final.

1.3 Hipóteses da pesquisa e abordagem Proposta

Nossos algoritmos de roteamento tiram proveito da grande capacidade de comunicação do nó *sink* através de um dispositivo de comunicação de maior potência. Neste sentido, as abordagens terão como métrica os valores de RSSI divulgados pelo *sink*, de forma que os nós regulares estejam aptos a estimarem suas distâncias entre si e entre o nó *sink*.

A abordagem clássica de detecção e contorno de buracos, discutida em [Karp & Kung, 2000], faz uso massivo do mecanismo de roteamento por perímetro para o reconhecimento de buracos de roteamento, provido pelo prévio conhecimento das informações de localização sobre os vizinhos de um salto de distância. Durante a etapa de reconhecimento da borda do buraco, os nós localizados na região de perímetro precisam trocar informações em múltiplos

saltos gerando sobrecarga de trabalho a estes dispositivos. Em nossos algoritmos, um nó regular não faz uso de nenhum sistema de atribuição de coordenadas virtuais ou ainda um GPS. Desta forma, precisávamos habilitar nossas soluções de forma que um nó preso na região do buraco, ou seja, vizinhos elegíveis a próximo salto, fosse capaz de aproveitar os valores de RSSI divulgados pelo *sink* e pelos vizinhos de um salto de distância para construir caminhos de escoamento válidos que circundassem a região de lacuna. Desta forma, nossa primeira hipótese busca responder ao seguinte questionamento: É possível criar rotas para o escoamento de pacotes até o *sink*, que contornem um buraco de roteamento detectado durante a operação da rede, sem que haja a necessidade de explorar os limites da borda desta região de buracos a Afim de proporcionar melhor gestão de recursos energéticos de nós sensores localizados em volta do buraco? Para desenvolver esse primeiro mecanismo de detecção e desvio, consideramos os valores de RSSI divulgados pelo sink e percebidos pelos nós regulares dispostos na grade poderiam ser utilizados como métrica. A partir desta hipótese, desenvolvemos o algoritmo ARESTA [Lima et al., 2016] e compramos seu desempenho ao do algoritmo RADR [Lima et al., 2013b], que é capaz de utilizar os valores do RSSI como parâmetro de configuração para o encaminhamento de pacotes em múltiplos saltos até o destino final, neste caso, o nó *sink*. A escolha pelo RADR é justificada pela similaridade dos cenários explorados, já que a estratégia de se utilizar um *sink* com grande capacidade de comunicação é inovadora e ainda pouco explorada. Além disso, os valores de RSSI utilizados como métrica para o processo de estimativa de distâncias entre os nós regulares e *sink* também pode ser utilizada para melhorar a qualidade dos caminhos de escoamento de pacotes gerados durante a operação dos algoritmos. O funcionamento do algoritmo bem como a análise dos mecanismos de detecção e desvio de buracos proposto são apresentados no Capítulo 4.

A segunda melhoria ocorre na etapa de encaminhamento de pacotes até o no *sink*. Neste sentido, realizamos um estudo comparativo entre as abordagens de roteamento geográfico empregadas para lidar com buracos de roteamento. A partir deste estudo, observamos que uma das estratégias utilizadas para a escolha eficiente do próximo salto de geográfico em direção ao *sink* é a adoção de algoritmos de roteamento oportunista, que visa o repasse de pacotes a vizinhos de um salto de distância com base em prioridades de eleição. Tal mecanismo é bastante utilizado para melhorar as taxas de entrega de pacotes e em alguns casos a diminuir carga de trabalho da rede [Eu & Tan, 2012, Huo & Wang, 2008]. Considerando a eficiência alcançada pelos algoritmos oportunistas, nossa segunda hipótese pretende responder à seguinte pergunta: É possível aprimorar a etapa de encaminhamento de pacotes na presença de buracos, de forma que os nós aptos e repassar pacotes em rotas geográficas, ou seja, nós a um salto de distância e geograficamente mais próximos ao *sink*, sejam capazes de eleger a si como saltos geográficos em direção ao *sink*, em múltiplos saltos? Com base

nesta segunda hipótese, implementamos o algoritmo de roteamento geográfico oportunista chamado REACT, avaliado e discutido em [Lima et al., 2017]. A nova técnica de encaminhamento de pacotes implementada proporcionada pelo REACT aproveita mensagens de *broadcast* oportunistas para o repasse de pacotes em direção ao *sink*. O mecanismo é capaz de proporcionar altas taxas de entrega, já que durante a etapa de encaminhamento mais de um vizinho apto (elegível) é escolhido para o repasse de informações em múltiplos saltos. A técnica de auto-eleição do próximo salto e repasse de informações em múltiplos saltos já foi discutida em trabalhos anteriores [Lima et al., 2013a]. A técnica de roteamento oportunista e os mecanismos para lidar com a ocorrência de buracos são apresentados no Capítulo 5.

Com a finalidade de estender a funcionalidade dos métodos de detecção e desvio de buracos a partir da grade de sensoreamento proposta nesta tese, nossa terceira hipótese responde à seguinte pergunta: Caminhos de escoamento de pacotes a partir dos nós regulares podem ser gerados com base em informações trocadas entre vizinhos de um salto de distância ainda na etapa inicial, de forma que a construção dessas rotas aproveite o conhecimento dos nós vizinhos de um salto de distância do nó *sink*? Motivados por esta hipótese, desenvolvemos nosso terceiro algoritmo que chamamos de LRS [Lima et al., 2018]. A idéia chave desta nova contribuição é a melhoria do processo de encaminhamento de pacotes a partir do nó preso, ou seja, um nó impedido de encaminhar pacotes ao *sink* por ausência de vizinhos aptos. Neste sentido, durante a etapa de estimativa de distâncias entre vizinhos de um salto, os nós próximos ao nó *sink* repassam aos vizinhos a informação de que possuem uma rota até o *sink*. Desta forma, os nós da grade de sensoreamento conhecem um caminho seguro até o *sink* e podem optar por este caminho sempre que for necessário. Essa informação adicional é utilizada pelo nó preso quando este detecta que está em uma região de buracos. O novo algoritmo é apresentado no Capítulo 6.

A quarta melhoria implementada a partir desta tese, se refere à forma de lidar com a ocorrência de buracos de roteamento. Neste sentido, a partir da terceira contribuição discutida anteriormente, nossa quarta hipótese procura responder a seguinte questão: É possível evitar o envio de pacotes à uma região de buracos durante a etapa de encaminhamento guloso? A resposta para este questionamento foi obtida a partir do desenvolvimento do algoritmo BYPATH. Nesta abordagem, a região de buracos é identificada previamente viabilizando a utilização do mecanismo de encaminhamento guloso. O algoritmo é capaz de superar buracos e gerar rotas alternativas que não só desviam o buraco, mas também evitam o envio de pacotes a vizinhos próximos ao nó preso, ou nó da região de buracos. O mecanismo de criação de rotas utilizado no algoritmo BYPATH também é capaz de proporcionar balanceamento de carga aos nós da borda ou proximidades do buraco, pois cria mais de um caminho de escoamento de pacotes. A abordagem é apresentada e discutida no Capítulo 7.

Nossos algoritmos executam três etapas iniciais, comuns às quatro soluções. Na etapa

inicial, o nó *sink* envia uma consulta à toda rede em um único salto de comunicação, conforme descrito na Figura 1.3(a). Ao perceberem a consulta, os nós regulares estimam suas distâncias em relação nó *sink* usando a técnica de RSSI, conforme já ilustrado anteriormente. Em seguida, cada nó irá enviar a seus vizinhos de um salto de distância um pacote de anúncio, que contém a sua distância estimada para o *sink*, conforme ilustrado na Figura1.3(b). Desta forma, os nós recebem diversos pacotes deste tipo, um de cada vizinho, e armazenam esses dados em suas tabelas de roteamento.

Para minimizar a carga dos nós intermediários durante do processo de repasse de mensagens, um temporizador é iniciado com o tempo de espera calculado a partir da potência de sinal percebida por cada nó. Assim, quanto mais baixo for o valor do RSSI percebido, menos tempo o nó terá que esperar para enviar seus pacotes em direção ao *sink*. Em nosso trabalho, o parâmetro utilizado para o cálculo estimado da distância foi dado em metros e o tempo utilizado na agregação de dados foi dado em segundos. O temporizador considera que cada metro corresponde a um segundo. Desta forma, após ter seu temporizador configurado e ativo, conforme ilustrado na Figura1.3(c), cada nó irá esperar um tempo em segundos antes de encaminhar suas informações. Atualmente, não existem soluções que tirem proveito desse conjunto de funcionalidades para a economia de energia, detecção e desvio de regiões de buraco e ainda prover agregação de dados durante o processo de transmissão das mensagens em múltiplos saltos.

Em nosso cenário de simulações, a camada física do modelo topológico de redes não é considerada e por consequência não fornece uma camada de acesso ao meio (*MAC Layer*) em termos reais. Neste sentido, para simular a perda de mensagens, o simulador utilizado fornece *plug-ins* para o descarte mensagens aleatoriamente ou com base em um modelo de interferência, como o modelo SINR (*Signal to Interference Noise Ratio*) ou interferência do ruído de sinal. Para definir o alcance da comunicação entre os nós, utilizamos os modelos de conectividade UDG (*Unit Disk Graph*) [Clark et al., 1991] e QUDG (*Quasi Unit Disk Graph*) [Kuhn et al., 2004], já disponíveis no ambiente de simulações.

Por fim, visando otimizar o processo de escolha de vizinhos elegíveis a próximo salto de encaminhamento, aprimoramos o processo de divulgação de distâncias pelos nós regulares nos algoritmos LRS e BYPATH, de tal forma que os fluxos de dados em direção ao *sink* fossem previamente gerados a partir de seus vizinhos imediatos, ou seja, nós que possuem o *sink* como vizinho irão repassar tal informação em múltiplos saltos. Assim, os nós responsáveis pelo encaminhamento de mensagens irão encaminhar pacotes em direção ao *sink* escolhendo vizinhos que possuem a informação de rota segura, gerada a partir dos vizinhos do *sink*.

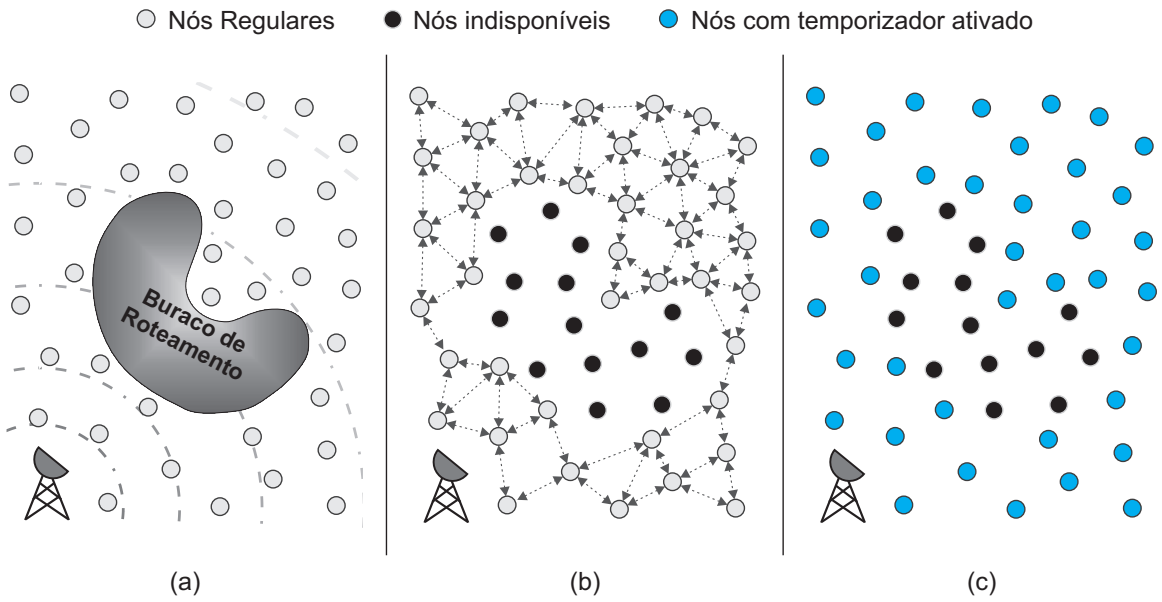


Figura 1.3. Primeiro estágio de funcionamento dos algoritmos propostos

1.4 Objetivos

Em Redes de Sensores Sem Fio (RSSFs), a ocorrência de regiões de buraco pode influenciar de forma significativa no desempenho das soluções de roteamento empregadas. Como já explanado, tais regiões podem ocorrer dinamicamente formando um tipo de área não balanceada, onde a densidade da rede é precária e pode levar a exaustão energética de parte da rede entre outras falhas de funcionamento. Interferências causadas por falhas nos nós, barreiras físicas como árvores, montanhas e construções também contribuem para o surgimento de regiões de buraco.

Nesta tese de doutorado, propomos quatro (04) soluções de roteamento geográfico, capazes de tirar vantagem desta nova arquitetura de comunicação e, ao mesmo tempo, capaz de lidar com regiões de buraco em RSSF e ainda realizar agregação de dados enquanto encaminha pacotes de dados ao *sink*. A ideia é prover um modelo alternativo ao mecanismo encaminhamento guloso, que seja capaz de detectar e contornar regiões de buraco, considerando que o nó *sink* utiliza um dispositivo de comunicação de longo alcance de tal forma que seja possível alcançar a toda a rede em uma única consulta. Os nós regulares irão responder as consultas do *sink* em múltiplos saltos, aplicando nossa técnica de encaminhamento guloso. Além disso, nosso algoritmo utiliza um temporizador configurado em cada nó que é utilizado para a agregação de dados durante o processo de encaminhamento. Os algoritmos tiram proveito dos valores de RSSI (*Received Signal Strength Indicator*) para calcular o próximo salto em direção ao nó *sink*, configurar uma tabela de vizinhos elegíveis a próximo salto e calcular um tempo de espera que possibilita a agregação de dados. Além disso,

nossos algoritmos consideram as informações recebidas a partir dos vizinhos de um salto de distância antes do início do encaminhamento para detectar regiões de buraco e criar fluxos para o escoamento dos pacotes em direção ao *sink*.

1.4.1 Objetivo Geral

Desenvolver uma solução eficiente de roteamento geográfico, composta de quatro algoritmos de encaminhamento guloso, livre de qualquer atribuição de coordenadas virtuais ou qualquer outro sistema de coordenadas custoso, que seja capaz de lidar com a ocorrência de buracos de roteamento, provocados pelo fenômeno dos mínimos locais em Redes de Sensores Sem Fio, tirando proveito do grande alcance de comunicação do nó *sink* em alcançar todos os nós da rede em uma única consulta.

1.4.2 Objetivos Específicos

Para atingir nosso objetivo geral, os seguintes objetivos específicos devem ser cumpridos:

1. Aprimorar umas das abordagens de roteamento geográfico da literatura para que seja possível lidar com buracos de roteamento sem a dependência de sistemas de coordenadas virtuais ou qualquer outro mecanismo custoso de obtenção de coordenadas;

2. Otimizar o encaminhamento de pacotes em direção ao *sink* através da estimativa de distância baseada nos valores de RSSI percebidos pelos nós regulares;

3. Demonstrar a eficácia de nossos algoritmos de roteamento geográfico em cenários com ocorrência de buracos, em termos de quantidade de pacotes trafegados na rede e, consequentemente, a taxa de pacotes entregues com sucesso ao destino final;

4. Melhorar a qualidade dos caminhos de roteamento gerados a partir de nossos algoritmos de roteamento durante a etapa de repasse de pacotes ao *sink*, realizada em múltiplos saltos;

5. Validar a eficácia das soluções de roteamento propostas, comparando-as ao mecanismo tradicional de detecção e desvio de buracos baseado em técnicas de roteamento por perímetro e mapeamento da borda do buraco, disponível na literatura;

6. Estender o mecanismo de reconhecimento e desvio de buracos que faz uso de um *sink* de alto alcance para que seja possível evitar o envio de pacotes a regiões de buraco.

Através deste conjunto de objetivos pretendemos implementar um conjunto de soluções para a detecção e criação de rotas em torno de regiões de buraco em Redes de Sensores Sem Fio, de forma a aumentar a taxa de pacotes entregues ao destino mantendo todos os benefícios do mecanismo clássico de encaminhamento guloso.

1.5 Principais contribuições

Os algoritmos desenvolvidos a partir da execução desta tese são descritos e discutidos nos Capítulos 4, 5, 6 e 7, onde são apresentados os algoritmos de roteamento geográfico ARESTA, REACT, LRS e BYPATH. Os algoritmos são capazes de lidar com uma região de buracos, utilizando a informação dos vizinhos de um salto de distância e os valores do RSSI propagados pelo *sink*. Nossas soluções criam caminhos alternativos até o *sink*, enquanto contornam a região de buracos durante a etapa de roteamento. Diversos experimentos foram realizados para avaliar o desempenho de nossos algoritmos. Os resultados evidenciam os benefícios introduzidos através da utilização de *sink* de grande capacidade de comunicação, bem como de nossos mecanismos de detecção e contorno de buracos de roteamento.

Destacamos em ordem de concepção as contribuições alcançadas nesta tese:

1. **ARESTA**: Um algoritmo de roteamento geográfico com agregação de dados capaz de detectar e desviar regiões de buraco em Redes de Sensores sem Fio (RSSF) utilizando *sinks* de alto alcance, publicado em [Lima et al., 2016];

2. **REACT**: Um algoritmo de roteamento geográfico oportunista, capaz de detectar e desviar regiões de buraco através da auto-eleição do próximo salto em RSSF utilizando *sinks* de alto alcance, publicado em [Lima et al., 2017];.

3. **LRS**: Um mecanismo de detecção e contorno de buracos de roteamento, utilizando um sink de longo alcance para a confecção de caminhos válidos para o escoamento de pacotes ao sink, publicado em [Lima et al., 2018]

4. **BYPATH**: Um algoritmo geográfico capaz de evitar buracos de roteamento em RSSF, submetido ao periódico Ad Hoc Networks em Dezembro de 2018 e em fase de revisão.

A Tabela 1.5 sumariza as principais características dos algoritmos propostos.

1.5.1 Comentário sobre publicações e submissão das contribuições

Nesta tese apresentamos quatro soluções de roteamento geográfico que lidam com buracos de roteamento em Redes de Sensores Sem Fio. Afim de validar nossas soluções ante

	Ténica de detecção de buraco	**Envio a partir do nó preso (buraco)**	**Implementa agregação de dados**
ARESTA	Baseada no conhecimento dos vizinhos imediatos	Encaminha pacotes contornando o buraco	SIM
REACT	Baseada no conhecimento dos vizinhos imediatos	Utiliza mensagens de *broadcast* oportunistas	SIM
LRS	Baseada em informações sobre vizinhos do *sink*	Escolhe uma rota criada a partir dos vizinhos do *sink*	SIM
BYPATH	Baseada em informações sobre vizinhos do *sink*	Nó preso não será acionado	NÃO

Tabela 1.2. Resumo das principais características de cada abordagem

a comunidade científica nacional e internacional, submetemos três de nossas abordagens para eventos científicos e um periódico de alto impacto, como veremos a seguir.

Os resultados produzidos pelo algoritmo ARESTA, foram submetidos e aceitos no XXXIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2016), ocorrido entre os dias 30 de Maio e 03 de Junho de 2016 na cidade de Salvador, BA. Após o aprimoramento da abordagem utilizada no algoritmo ARESTA, propusemos e submetemos os resultados obtidos pelo algoritmo REACT ao periódico de alto impacto *Ad Hoc Networks*. Como resultado, em 06 de Setembro de 2017 o REACT ficou disponível online, sendo publicado em definitivo no **Volume 67** do periódico em Dezembro de 2017 .

Finalmente, propusemos e desenvolvemos o algoritmo LRS. Os resultados obtidos através do algoritmo foram submetidos aceitos no Simpósio Internacional *ISCC (Symposium on Computers and Communications)* promovido anualmente pelo *IEEE (Institute of Electrical and Electronics Engineers)*. Artigo está disponível na plataforma digital *IEEExplore*, deste Novembro de 2018 .

1.5.2 Principal contribuição de pesquisa

Considerando os resultados obtidos com a implementação do algoritmo BYPATH, observamos que a combinação dos mecanismos de detecção de buracos aliada ao conhecimento sobre os vizinhos válidos do nó *sink* e o aperfeiçoamento da etapa de estimativa de distância entre os nós regulares e o *sink*, representam um importante avanço referente a abordagens geográficas operando na presença de buracos de roteamento. Nossos resultados evidenciam que o algoritmo BYPATH se mostrou uma solução eficaz em todas as etapas do roteamento, sendo capaz superar buracos de roteamento e ainda manter níveis elevados de taxa de sucesso na entrega de pacotes ao *sink*, enquanto diminui a carga dos nós localizados no perímetro do buraco. Desta forma, entendemos que a quarta contribuição tese, o algoritmo BYPATH, foi capaz de agregar um conjunto de funcionalidades que a coloca em evidência entre as soluções discutidas neste trabalho.

1.6 Organização desta Tese de Doutorado

O Capítulo 2 apresenta uma visão geral sobre Redes de Sensores Fio, além de desafios de projeto e características inerentes ao cenário proposto. Também identificamos e apresentamos considerações relevantes sobre a metodologia envolvida no roteamento geográfico e o mecanismo de encaminhamento guloso, mostrando tanto os benefícios da abordagem geográfica quanto suas limitações em relação ao processo de encaminhamento de mensagens. Um breve comentário sobre fusão de dados e sua importância para as RSSFs também é apresentado. Neste contexto, damos ênfase à técnica de agregação de dados, utilizada para reduzir o número de informações redundantes, além de reduzir o consumo de energia da rede. Por fim, apresentamos uma visão geral sobre o problema das regiões de buraco em RSSF, os impactos de sua ocorrência e soluções de detecção e desvio de tais regiões.

Um resumo dos trabalhos relacionados é apresentado no Capítulo 3, onde são mostrados e categorizados algoritmos que utilizam abordagens geográficas e são capazes de lidar com regiões de buraco em RSSF.

Nos Capítulos 4, 5, 6 e 7, apresentamos e avaliamos as quatro (04) soluções de roteamento geográfico propostas, que não necessitam de sistemas de coordenadas e ainda são capazes de lidar com regiões de buraco em RSSF. Os algoritmos (i) ARESTA, (ii) REACT, (iii) LRS e (iv) BYPATH, levam em consideração que o nó *sink* está equipado com um dispositivo de comunicação de longo alcance de tal forma que seja possível alcançar todos os nós da grade em uma única consulta. O retorno dos nós sensores para o nó *sink* é realizado através de múltiplos saltos, utilizando nossa técnica de roteamento geográfico livre de sistemas de coordenadas que utiliza os valores do RSSI (*Received Signal Strength Indicator*) para calcular o próximo salto em direção ao nó *sink*, calcular o tempo de espera que possibilita a agregação de dados e ainda contornar regiões de buraco durante o processo de encaminhamento. Finalmente, no Capítulo 8 apresentamos as considerações finais, bem como as sugestões para a continuidade do presente trabalho.

Capítulo 2

Fundamentos Teóricos

Neste capítulo, apresentamos conceitos necessários para o entendimento sobre os protocolos de roteamento (seção 2.1) e o mecanismo de encaminhamento guloso (*greedy forwarding*), organizados na seção 2.2. Os conceitos sobre a ocorrência das regiões de buraco, seus impactos e soluções existentes são apresentados na seção 2.3. Também classificamos soluções de detecção e desvio de regiões de buraco em quatro categorias, para melhor entendimento.

A seção 2.4 apresenta uma visão geral sobre fusão de dados em Rede de Sensores sem Fio (RSSF) e sua importância para o tratamento de informações redundantes, além de um tópico sobre o mecanismo de agregação de dados. Já na seção 2.6, apresentamos algumas definições importantes para um melhor entendimento desta tese, ao passo que os conceitos relacionados à estimativa de distância baseada no RSSI, bem como o modelo utilizado neste trabalho são apresentados na seção 2.5.

2.1 Visão Geral sobre Protocolos de Roteamento

Roteamento é o processo de escolha de um caminho válido, entre a origem e o destino, conforme a solicitação de transmissão recebida. Em RSSFs, o roteamento difere das redes convencionais (que utilizam endereços IP) em vários aspectos. Não há infraestrutura definida, os nós estão sujeitos a falhas e restrições de energia. Por esta razão, protocolos de roteamento precisam atender um conjunto de requisitos como: autonomia, eficiência energética, escalabilidade, resiliência, heterogeneidade (em relação aos sensores) e mobilidade [Akkaya & Younis, 2005, Villalba et al., 2009].

Muitos protocolos de roteamento têm sido desenvolvidos para RSSFs. Os autores Goyal & Tripathy [2012] classificaram os protocolos de roteamento em cinco categorias: Protocolos baseados em Localização, Centrados em Dados, Hierárquicos, baseados em Caminhos Múltiplos e baseados em QoS. Um resumo com as informações sobre cada protocolo

é apresentado na Tabela 2.1.

Tabela 2.1. Classificação dos Protocolos de Roteamento

Protocolo	Descrição
Baseados em Localização	Nós sensores são endereçados através de suas informações de localização, necessárias para o cálculo da distância entre dois nós.
Centrados em Dados	Difere dos protocolos tradicionais na maneira em que os nós enviam suas informações ao *sink*. Quando um nó envia seu dado ao *sink*, sensores intermediários podem realizar agregação, nos dados originados de múltiplos sensores, para então encaminhar os dados agregados até o *sink*.
Hierárquicos	Protocolos organizados em uma hierarquia, em que a rede é composta por agrupamentos (*clusters*) de sensores. Cada grupo é gerenciado por um nó específico, chamado *cluster head*, responsável pelas atividades de transmissão de dados dos membros do grupo.
Baseados em Múltiplos Caminhos	Considera que durante uma transmissão de dados até o *sink*, dois paradigmas são seguidos: roteamento *single-path* e *multipath*. No primeiro, cada nó envia seu dado ao *sink* através do menor caminho. Já no segundo, cada sensor (origem) encontra do primeiro k menor caminho até o *sink*, dividindo sua carga entra os caminhos encontrados.
Baseados em QoS	Protocolos baseados em métricas de QoS consideram requisitos de comunicação fim-a-fim enquanto tais como atraso, confiabilidade e tolerância a falhas enquanto configuram melhores opções de rota através da rede.

Quando o nó encontra-se distante ou fora do alcance de comunicação de um segundo nó, a estratégia de encaminhamento de pacotes se dá através de múltiplos saltos. Desta forma, nós intermediários irão repassar pacotes aos vizinhos a um salto de distância, que estejam dentro do alcance da comunicação deste último nó. Em muitas aplicações de RSSFs não é possível atribuir identificadores globais para cada nó devido ao grande número de nós implantados. Essa ausência de identificação global, juntamente com a implantação aleatória de nós sensores, traz desafios no tocante a escolha de um próximo salto.

2.2 Roteamento Geográfico

Soluções baseadas em roteamento geográfico, têm sido propostas para facilitar o transporte de dados nas RSSFs. As abordagens utilizam a posição dos nós da rede ao invés de seus endereços para criar caminhos em direção ao destino final das mensagens [Boukerche et al., 2008]. Cada nó só precisa conhecer a posição dos seus vizinhos, normalmente a um salto de distância, para a criação das rotas. Uma forma simplificada de utilização dos algoritmos de roteamento geográfico é utilizar o mecanismo de encaminhamento guloso (*greedy forwarding*). A técnica realiza o roteamento dos pacotes através do encaminhamento de mensagens a um vizinho geograficamente mais próximo ao destino final dos pacotes [Lima et al., 2013a]. A Figura 2.1, mostra o funcionamento da estratégia de encaminhamento guloso. O nó fonte *S* seleciona o nó *A* que está geograficamente mais próximo do destino *D*. O mesmo ocorre em relação ao nó *A*, que seleciona o nó *B*, geograficamente mais próximo do destino *D*. Finalmente, o nó *B* encaminha o pacote replicado ao destino *D*.

O roteamento geográfico possui diversas características que vão ao encontro das singularidades encontradas nas RSSFs como por exemplo, não possuir monitoramento de estado, uma vez que os nós só precisam armazenar informações sobre seu posicionamento atual de um número limitado de destinos ou *sinks*. Esta é uma característica importante, pois nós sensores possuem limitação de memória. Além disso, sendo baseado em informações locais, o roteamento geográfico é escalável e portanto, aplicável a RSSFs densas [Nati, 2008].

Um dos trabalhos mais relevantes da literatura sobre protocolos de roteamento geográfico é o *Greedy Perimeter Stateless Routing for Wireless Networks (GPSR)*, discutido por Karp & Kung [2000]. A abordagem utiliza a posição dos nós e o destino do pacote para tomar decisões de encaminhamento guloso, através das informações sobre os vizinhos a um salto de distância. O roteamento geográfico se baseia no conhecimento da localização física

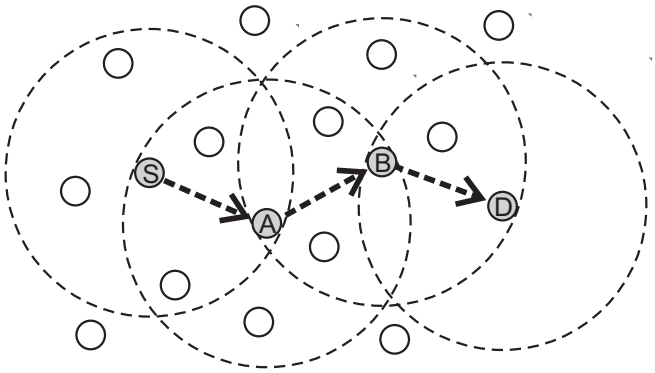


Figura 2.1. Mecanismo de encaminhamento guloso. A cada passo, o pacote é repassado para o vizinho mais geograficamente próximo do nó *sink*. Baseado em [Park et al., 2010].

dos nós sensores. Os nós podem ser equipados com dispositivos GPS mas deixam de ser economicamente viáveis. A alternativa é a utilização de algoritmos de localização para o cálculo das coordenadas físicas. Na abordagem clássica, a informação de posição do destino do pacote é anexada ao cabeçalho do pacote de forma que os saltos intermediários sejam capazes de reunir informações sobre a origem e o destino final dos pacotes de dados.

Conforme discutido por Chen & Varshney [2007], a maioria das abordagens de roteamento geográfico opera de duas formas: (a) utilizando o mecanismo de encaminhamento guloso para o repasse ao próximo salto e; (b) utilizando um mecanismo de prevenção de lacunas de roteamento ou regiões de buraco. No primeiro modo, a escolha do próximo salto é feita de acordo com as posições do nó atual, seus vizinhos e o nó que representa o destino final. Já na segunda forma de operação, se um nó deseja encaminhar pacotes a um destino mas está impossibilitado de escolher o próximo salto em direção a este destino, a abordagem utilizada alterna para o modo de recuperação de rotas em direção ao destino. Neste modo, o nó irá tentar encaminhar suas informações considerando a borda da região de buracos encontrada. Desta forma, o pacote será encaminhado a vizinhos localizados na borda do buraco até que uma rota válida será definida em direção ao destino final.

Na seção seguinte apresentamos os conceitos sobre as regiões de buracos, sua ocorrência e impactos nas RSSF.

2.3 Regiões de Buraco

A ocorrência de regiões de buraco em RSSF é um grande desafio para a utilização de protocolos de roteamento geográfico no tocante à transmissão de mensagens através da rede. Neste sentido, as regiões de buraco surgem como uma categoria específica de problema associada tanto ao fenômeno dos mínimos locais (*local minimum*), amplamente discutido em [Fang et al., 2004, Karp & Kung, 2000, Yu et al., 2007], quanto a obstáculos físicos, esgotamento energético, falhas na comunicação entre vizinhos eleitos e implantação incorreta no campo ou área de interesse do sensoreamento [Chen & Varshney, 2007].

Uma das alternativas mais utilizadas para lidar com o problema das regiões de buraco é abandonar o mecanismo de encaminhamento guloso tradicional e adotar o conhecido esquema de roteamento por perímetro. Este esquema de roteamento é aplicado sobre os nós próximos ou ao longo da borda de uma região de buraco e envolve o conhecimento que nós possuem sobre os limites do buraco e manutenção das rotas ao redor deste buraco. Contudo, de acordo com Le et al. [2012], este esquema de roteamento não satisfaz o requisito de balanceamento de carga, pois impõe tráfego demasiado aos nós da borda do buraco, tornando-os mais frequentemente usados do que outros nós.

Como forma de facilitar o entendimento sobre a ocorrência das regiões de buraco, neste trabalho utilizamos a categorização de buracos discutida por Ahmed et al. [2005] e Khan & Javed [2008]. Segundo os autores, há quatro (04) tipos principais de buracos em redes de sensores: *(i)* buracos de cobertura, *(ii)* buracos de roteamento, *(iii)* buracos causados por falhas ou bloqueios na comunicação e, *(iv)* buracos causados por agentes maliciosos. A seguir, as características e impactos de cada problema identificado.

2.3.1 Buracos de Cobertura

Os nós em uma Rede de Sensores sem Fio (RSSF) são normalmente implantados de forma aleatória em uma área de sensoreamento. A forma como essa área é monitorada é chamada de cobertura, e é uma das métricas mais importantes em redes de sensores [Wang et al., 2009]. Problemas relacionados à cobertura em RSSF podem ser formulados de várias formas e englobando diferentes cenários. De acordo com Wang [2011], o modelo de cobertura adotado por uma RSSF é utilizado para medir, entre outros aspectos, a capacidade de sensoreamento e a qualidade do dispositivo de sensoreamento utilizado.

Particularmente na utilização de protocolos de roteamento, os problemas relacionados à cobertura apontam para limitações clássicas deste tipo de abordagem que são o esgotamento de energia e topologia de rede. Neste sentido, em relação à topologia da rede, uma simples realocação de nós ou a ainda uma re-implantação na área sensoreada poderia facilmente suprir tal limitação. De acordo com Di Francesco et al. [2011] essa estratégia se mostra ineficiente para lidar com os buracos de cobertura – regiões onde a densidade da rede não é adequada ou não está ajustada para permitir a correta comunicação entre os nós dispostos na grade de sensoreamento.

Segundo Jabeur et al. [2013], buracos de cobertura (*coverage holes*) ocorrem com frequência em cenários onde a implantação dos nós é feita de forma aleatória, ou na presença de obstáculos naturais e demais obstruções. Ainda de acordo com os autores, a existência de tais buracos está atrelada ao grau de cobertura desejado para cada aplicação. Neste sentido, um buraco de cobertura não irá existir se todo ponto em uma dada área é coberto (alcançado) por pelo menos k sensores, onde k é o grau requerido de cobertura para aquela rede. De forma geral, a ocorrência de buracos de cobertura tende a afetar o desempenho das RSSFs, diminuindo a confiabilidade das soluções de roteamento, alterando a topologia das redes, interrompendo a comunicação entre os nós e ainda aumentando a carga de trabalho dos nós localizados ao redor do buraco.

2.3.2 Buracos de Roteamento

Os buracos de roteamento (*routing holes*) podem surgir em uma área onde os nós sensores não estão disponíveis ou não podem participar do processo de roteamento de pacotes na rede devido a mau funcionamento, obstáculos geográficos de bateria ou destruição do nó a partir de um evento externo (incêndio, terremoto, etc). Nestes casos, um protocolo de roteamento tem duas opções a serem aplicadas ao cenário: (a) rejeitar pacotes encaminhados ao nó preso ou; (b) resolver o problema de rotas até o destino através de algum mecanismo de escolha de caminhos alternativos.

De acordo com Ahmed et al. [2005], lacunas de comunicação originadas ainda na implantação dos sensores podem causar buracos de roteamento. Esse tipo de lacuna entre os nós leva a grande maioria dos algoritmos de roteamento geográfico a falhar durante a transmissão as mensagens na rede [You et al., 2009b]. Além disso, buracos de roteamento também podem ocorrer devido ao fenômeno dos mínimos locais (*local minimum*), problema frequentemente enfrentado pelos mecanismos de encaminhamento guloso, que ocorre quando não há vizinho elegível que esteja geograficamente mais próximo ao destino final.

2.3.3 Buracos Causados por Falhas ou Bloqueios na Comunicação

Os buracos causados por falhas ou bloqueios na comunicação (*jamming holes*), podem ocorrer quando se torna impossível existir comunicação eficaz entre os nós da rede [Antil & Malik, 2014, Ghosh et al., 2014]. Entre os motivos de sua ocorrência destacamos a intrusão na frequência de rádio alocada e a utilização ininterrupta do canal sem fio. Sua ocorrência também se dá devido a implantação fora de ordem dos nós e que por este motivo podem enviar pacotes com informações irrelevantes (lixo) e assim comprometer a utilização do canal sem fio. É possível ainda que nós comprometidos por algum atacante externo (portanto maliciosos) inundem a rede com pacotes inúteis com o objetivo de exaustão da rede. O alto nível de ruído em uma determinada área de sensoreamento pode ser detectado por vizinhos de forma que uma rota alternativa seja configurada nos nós ao redor do buraco.

A ocorrência de um buraco de bloqueio pode ser dar de forma intencional e não intencional, conforme discutido por Ahmed et al. [2005]. Quando ocorre de forma intencional, significa que um atacante externo está atentando contra o correto funcionamento da rede interferindo deliberadamente na comunicação entre os nós. Em interferências não intencionais, um ou mais nós da rede estão operando incorretamente e continuamente ocupam largura de banda do canal sem fio impedindo a comunicação de nós vizinhos. Buracos de bloqueio também podem ocorrer em regiões onde eventos ou objetos detectados possuem comportamentos diferentes do esperado, por exemplo, um nó implantado e configurado para detectar

diferentes escalas de luminosidade é incapaz de detectar os valores esperados nesta região bloqueada de buraco [Khan & Javed, 2008].

2.3.4 Buracos Causados por Agentes Maliciosos

Buracos causados por agentes maliciosos (*worm holes*) são formados quando nós implantados em diferentes partes da grade de sensoreamento iniciam trocas de mensagens entre si a partir de diferentes partes (regiões) da rede. Conforme discutido por Jabeur et al. [2013], nós maliciosos podem criar túneis de comunicação entre si e iniciar o encaminhamento de mensagens entre as extremidades do túnel, esgotando os recursos do canal sem fio nestas conexões que ocorrem em paralelo. De acordo com Ahmed et al. [2005], tais buracos podem ser classificado com um tipo de ataque por negação de serviço (*DoS*), pois o nó receptor malicioso receptor replica a mensagem incorreta a uma outra parte rede. Isso faz com que nós localizados em diferentes partes de redes acreditem que são vizinhos, resultando em convergência incorreta de roteamento.

2.4 A Fusão de Dados em RSSF

A Fusão de Dados é uma importante abordagem tecnológica em RSSFs e tem sido apontada como alternativa para o pré-processamento dos dados de forma distribuída, aproveitando a capacidade de processamento dos nós. Algoritmos de fusão de dados são projetados para lidar com um grande número de dados, de forma a remover informações redundantes e minimizar a quantidade de dados transmitidos na rede, conforme discutido por [Loureiro et al., 2003] e [Li et al., 2011].

As RSSFs são consideradas redes centradas em dados, ou seja, o interesse nos dados sensoreados não se restringe à aplicação sendo comum a todas as atividades que possam tirar proveito da correlação existente entre estes dados. Assim, as atividades como roteamento devem permitir que os dados sejam analisados no nível da aplicação para decidir de estes serão retransmitidos, fundidos ou suprimidos. De acordo com Nakamura et al. [2007], os métodos mais comuns de fusão de dados são: (1) agregação, (2) inferência, (3) estimação. Os métodos de agregação são os mais simples e produzem como resultado um dado de menor representatividade do que o conjunto dos dados utilizados na fusão. A vantagem destes métodos reside na redução do volume de dados que trafegam pela rede e inclui operações de agregação como média, máximo, e mínimo. Os métodos de inferência têm como objetivo processar dados e tirar conclusões a respeito dos mesmos [Abdelgawad & Bayoumi, 2012]. A decisão é tomada com base no conhecimento da situação percebida. Exemplos destes métodos são: Fusão Bayesiana e Filtros Dempter-Shafer. Já os métodos de estimação têm

como objetivo estimar o vetor de estado de um processo a partir de um vetor ou sequência de vetores de medições de sensores. Exemplos destes métodos são: *quadrados mínimos, quadrados mínimos ponderados* e *Filtros de Kalman*.

Um dos pontos fundamentais no desenvolvimento de algoritmos de fusão de dados, está no tratamento adequado de informações redundantes. Este requisito precisa ser atendido, afim de se obter dados relevantes e ainda: reduzir a quantidade de transmissão de dados, reduzir a colisão de dados durante a transmissão, reduzir o congestionamento da rede, economizar energia de um nó e além disso, prolongar o tempo de vida da rede. Lin et al. [2010] considera que a acurácia alcançada na entrega dos dados ao *sink* é apontada como um benefício direto da fusão de dados. Neste sentido, a acurácia é medida pelo total de dados enviados por todos os nós durante o ciclo de vida da rede.

2.4.1 Agregação de Dados

A agregação de dados é uma forma especial de fusão de dados. Na agregação, os dados originados de diferentes fontes são combinados através da eliminação de duplicidades. Algumas estratégias de agregação podem ser originadas a partir de cada nó ou ainda serem conduzidas por nós específicos na rede. Neste sentido, a agregação de dados consiste em um mecanismo utilizado para suprimir a redundância nas mensagens oriundas de diferentes fontes, além de reduzir o consumo de energia da rede. De acordo com Al-karaki & Kamal [2004], a agregação de dados é apresentada como técnica de roteamento essencial para a estratégias envolvendo formação de clusters e diminuição da dados redundantes, pois determinadas aplicações geram grande quantidade de tráfego de dados. Já em [Yoneki & Bacon, 2005], agregação em RSSFs é apontada como estratégia de combinação ou sumarização de informações de alguns ou todos os sensores. Em cenários onde vários nós sensores possuem informações semelhantes, agregação de dados é utilizada para diferenciar dados simplesmente coletados de dados ou informações úteis [Abbasi & Younis, 2007]. Esta característica é fundamental para o processamento das informações e o repasse dos dados agregados.

Cohen et al. [2001] defende que a agregação de dados é um processo que inclui uma coleção de dados, uma composição flexível e programada destes dados em um novo dado refinado de menor volume somados à entrega destes dados agregados a um destino pretendido. A técnica é utilizada para resolver problemas de implosão e sobreposição no roteamento centrado em dados.

Uma RSSF é geralmente concebida como uma árvore *multicast* inversa. Neste sentido, o nó *sink* inicia o roteamento através de uma consulta divulgada em múltiplos saltos, solicitando informações a partir dos nós da grade. Após isso, os dados provenientes dos múltiplos nós sensores são agregados e enviados ao destino final, conforme ilustrado na 2.2.

Em cenários com a ocorrência de buracos de roteamento, o nó preso na região do buraco é normalmente responsável pelo repasse de uma grande quantidade de pacotes oriundos de nós mais distantes do destino final dos pacotes, normalmente um nó *sink*. Isso ocorre na maioria dos algoritmos geográficos, pois a rota usual dos pacotes é implementada a partir da técnica de encaminhamento guloso, que obedece ao critério de encaminhar pacotes a vizinhos geograficamente mais próximos ao destino final. Nestes casos, a agregação de dados pode aprimorar a etapa de repasse de pacotes pois pode armazenar as informações recebidas e agrupá-las em pacotes de dados agregados. Neste trabalho, utilizamos a agregação de dados com forma de reduzir o número de informações redundantes, diminuir a carga de trabalho imposta naturalmente imposta aos nós da borda do buraco e aumentar o tempo de vida da rede, além da economia de energia que é um benefício direto da agregação de dados.

2.5 Estimativa de Distância Baseada no RSSI

Em RSSFs, diferentes métodos podem ser utilizados para estimar distâncias e ângulos entre dois nós. Neste contexto, os sistemas de localização utilizam métodos que incluem o tempo de chegada e diferença do tempo de chegada (ToA/TDoA), ângulo de chegada do sinal (AoA) e o RSSI (*Received Signal Strength Indicator*), ou Indicador de Potência do Sinal Recebido [Boukerche et al., 2007], [Zhu et al., 2009].

O RSSI é um dos parâmetros mais utilizadas em estudos referentes às RSSFs. O RSSI pode ser usado para estimar a distância entre dois nós baseado na força do sinal recebido pelo outro nó. Um nó que está transmitindo informação envia um sinal com um a certa força que irá decrescer conforme a propagação. Quanto maior a distância do nó que recebe a informação, menor será a força do sinal. Esta estratégia tem sido amplamente utilizada

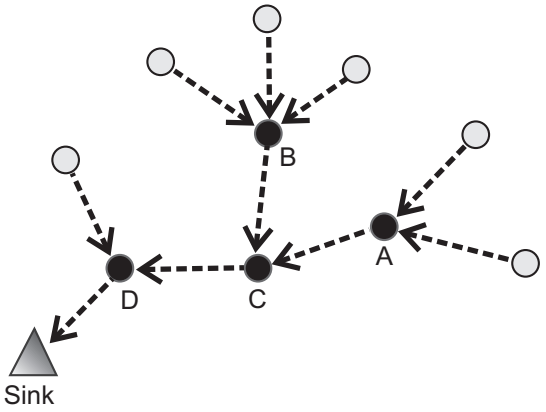


Figura 2.2. Exemplo de agregação de dados. Nós *A*, *B*, *C* e *D*, repassando informações agregadas.

como parâmetro para aferição da potência de sinal, recebida pelos nós.

Na literatura é possível encontrar diversos modelos de propagação de ondas de rádio para redes sem fio. Tais modelos podem prever a intensidade média do sinal recebido a determinada distância do transmissor, além da variabilidade da intensidade do sinal em áreas muito próximas a um local específico. De acordo com Benkic et al. [2008], três são os modelos mais utilizados em RSSFs:

- **Modelo de propagação no espaço livre:** a potência recebida por uma antena, colocada em uma distância de separação d de uma antena transmissora, é dada pela conhecida equação do espaço livre (equação de *Friis*) [Rappaport, 2002],

$$P_r(d) = C_f \frac{P_t}{d^2}$$

  onde P_t é a potência transmitida, $P_r(d)$ a potência recebida, e C_f uma constante de transmissão. Atualmente, muitos dispositivos de sensoreamento são equipados com circuitos capazes de perceber e aferir medições de valores do RSSI propagado, como *Smartphones*, *Tablets* e dispositivos projetados para Internet das Coisas (IoT).

- **Modelo de reflexão no solo (modelo de dois raios):** considera o caminho direto e um caminho de propagação refletido no solo entre o transmissor e o receptor,

$$P_r(d) = C_t \frac{P_t}{d^4}$$

  onde C_t representa as características da transmissão para o modelo.

- **Modelo *Log*-Distância:** derivado de métodos empíricos e analíticos, o modelo considera que a potência média recebida irá decrescer, de forma logarítmica, em relação à distância do transmissor. O modelo é bastante utilizado em simulações envolvendo RSSFs. A simplificação do modelo é dada pela fórmula,

$$P_r(d) \propto \frac{P_t}{d^{\alpha}}$$

  onde α é chamado de gradiente distância-potência. O modelo é amplamente discutido por Santi [2005] e Jacinto [2012].

Em condições ideais, o valor do RSSI pode ser usado para o cálculo da distância estimada entre um transmissor e um receptor. Conforme defendido por Jacinto [2012], o RSSI é apresentado como parâmetro confiável do que se refere à qualidade do canal sem fio em zonas com baixa ou alta taxa de recepção e acima do limiar de sensibilidade da antena.

A confiabilidade do RSSI, utilizado como métrica também foi discutida por Holland et al. [2006]. Nesta abordagem, o RSSI é comparado à técnica LQI (*Link Quality Indicator*) e apresenta uma margem de erro muito inferior a este último. Por fim, a estimativa de distância baseada no RSSI também foi amplamente discuta por Al Alawi [2011] e Ferreira et al. [2014]. Ambos afirmam que os valores do RSSI podem ser utilizados como parâmetro para a aferição de distâncias entre os nós em uma RSSF.

Nesta tese, o RSSI foi utilizado como parâmetro de configuração para estimar as distâncias entre os nós regulares e o *sink*, de acordo com o modelo *Log*-Distância. A aferição é baseada na potência do sinal recebido, de forma que nós mais distantes do *sink* recebem valores de RSSI menores (devido à propagação do sinal). Neste sentido, nosso cenário de simulação abstrai a camada de hardware, não atribuindo tarefas à camada de acesso ao meio (MAC), em termos reais.

2.6 Definição do Problema

Nesta seção, serão apresentadas algumas definições para um melhor entendimento deste trabalho.

Definição 1 - Escopo do problema: Considere uma Rede de Sensores sem Fio (RSSF) como sendo um grafo euclidiano $G = (V, E, r)$, com as seguintes propriedades:

- $n = |V|$ é o número de nós e r, o raio de comunicação;
- $V = v_0, v_1, ..., v_{n-1}$, onde v_0 é o único nó sorvedouro, e v_i é um nó fonte para todo $0 < i < n$;
- $\langle i, j \rangle \in E$, se e somente se v_j é o pai de v_i, ou seja, v_i envia seus dados para o sorvedouro através de v_j;

Definição 2 - Algoritmo de Consulta de Dados: Diferentemente de tradicionais esquemas de roteamento fim-a-fim que tentam manter rotas entre nós de origem e destino, este trabalho está focado em algoritmos de roteamento para consulta de dados, a fim de dar suporte à agregação de dados *in-network* [Boukerche et al., 2005]. Neste algoritmo, um *sink* equipado com um dispositivo de comunicação mais potente, envia uma consulta a todos os nós da rede em um único salto. Em seguida, os nós sensores com dados relacionados montam um pacote de resposta que é enviado para o nó *sink*, em múltiplos saltos. Em cada salto, os nós sensores podem olhar o conteúdo dos pacotes e efetuar agregação de dados em múltiplos pacotes de entrada para encaminhar um único pacote agregado, de acordo com a função de agregação especificada na consulta. Neste trabalho, consideramos funções simples de agregação de dados [Krishnamachari et al., 2002] como: *máximo, mínimo e média*.

Definição 3 - Mensagem de Broadcast: Uma mensagem de *broadcast* é uma forma de comunicação em que um nó (origem) $s \in V(G)$ dissemina pacotes a todos os vizinhos.

Definição 4 - Região de Buracos: Uma região ou área onde os nós sensores não estão disponíveis ou os nós disponíveis não podem participar na etapa de roteamento dos dados coletados devido a mau funcionamento, esgotamento energético ou destruição causada por evento externo [Fang et al., 2004].

Definição 5 - Nó Preso (stuck node): Um nó preso (*stuck node*) é assim chamado quando um ou mais pacotes encaminhados a partir vizinhos de salto de distância, via abordagens geográficas, não podem ser repassados adiante por ausência de um ou mais vizinhos geográficos a partir deste nó, que dêem continuidade ao processo de roteamento até o destino [Kanno et al., 2009].

Capítulo 3

Trabalhos Relacionados

3.1 Introdução

Neste capítulo, apresentamos soluções para o problema da ocorrência das regiões de buracos em Redes de Sensores Sem Fio. Os trabalhos aqui apresentados abordam descrições e informações gerais sobre o funcionamento de cada abordagem com suas características inerentes.

Um grande número de protocolos de roteamento geográfico à disposição na literatura utilizam o mecanismo de encaminhamento guloso (*greedy forwarding*) para suprir a etapa de encaminhamento de pacotes a um destino desejado. Os nós que participam do processo de encaminhamento são chamados de nós de encaminhamento (*forwarding nodes*) e sua escolha é feita a partir de um conjunto de informações sobre nós e suas posições, normalmente dispostos em uma tabela de vizinhos presente em cada nó da rede. Desta forma, o nó que irá repassar a mensagem a um vizinho é responsável por decidir qual a melhor opção de escolha entre os vizinhos disponíveis, tomando como base a distância entre este vizinho e o destino final dos pacotes. Neste caso, o vizinho que está mais próximo a esse destino final deverá ser escolhido como próximo salto e assim o processo de escolha se repete até o nó de destino ser alcançado. Entretanto, na ocorrência de regiões de buraco alguns nós não serão capazes de eleger um nó intermediário como próximo salto pois não haverá vizinhos mais próximos ao destino final do que o próprio nó que precisa encaminhar a mensagem adiante. Por isso, na intenção de suprir tal limitação algumas abordagens de roteamento geográfico que tiram proveito do mecanismo de encaminhamento guloso são capazes de reconhecer regiões de buracos e ainda gerar caminhos válidos em direção ao destino.

Nas seções a seguir, apresentamos soluções disponíveis na literatura para lidar com o problema da ocorrência de regiões de buraco em RSSFs, utilizadas por protocolos de roteamento geográfico durante a etapa de encaminhamento de pacotes a um destino.

3.1.1 Greedy Perimeter Stateless Routing for Wireless Networks (GPSR)

O *Greedy Perimeter Stateless Routing for Wireless Networks (GPSR)* [Karp & Kung, 2000], utiliza a posição dos nós e o destino do pacote para tomar decisões de encaminhamento guloso, baseado nas informações dos os vizinhos a um salto de distância. O algoritmo consiste de dois métodos de encaminhamento de pacotes: encaminhamento guloso, usado quando aplicável, e encaminhamento por perímetro, utilizado em regiões em que o primeiro falha. Nesta abordagem, quando um pacote atinge uma região onde o encaminhamento guloso não é possível, o algoritmo passa a operar em modo de roteamento por perímetro. Uma vez que o pacote encontra uma borda, ou um caminho ótimo, o pacote percorre a superfície do perímetro, alternando entre as bordas no sentido anti-horário, estratégia conhecida como regra da mão direita. Na Figura 3.1, um exemplo do funcionamento do algoritmo é mostrado. Neste exemplo, x está mais próximo a D do que seus vizinhos w e y. O arco pontilhado, que é o alcance de comunicação de D, possui raio igual a distância entre x e D. Embora as opções de caminhos $(x \rightarrow y \rightarrow z \rightarrow D)$ e $(x \rightarrow w \rightarrow v \rightarrow D)$, existam para D, x não irá optar pelo encaminhamento a w ou y utilizando encaminhamento guloso, pois ambos estão fora da regra de encaminhamento. Neste caso, o algoritmo opta pelo mecanismo de encaminhamento por perímetro com a aplicação da regra da mão direita.

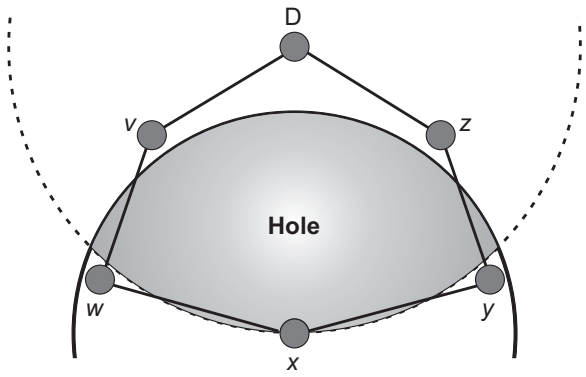


Figura 3.1. Exemplo da ocorrência de um buraco entre o nó x e o destino D. Baseado em [Karp & Kung, 2000].

3.1.2 Geographical and Energy Aware Routing (GEAR)

Um protocolo que utiliza a informação geográfica dos nós, para propagar dados a uma região retangular específica na grade de sensoreamento foi proposto por Yu et al. [2001]. O *Geographical and Energy Aware Routing (GEAR)*, foi projetado para prover eficiência energética e utiliza uma seleção de vizinhos cientes de energia para encaminhar pacotes em

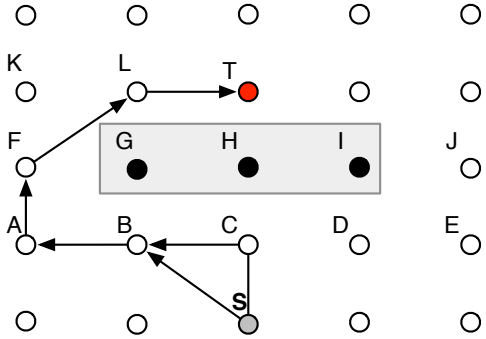


Figura 3.2. Algoritmo GEAR aprendendo caminhos ao longo da região de buraco. Os nós na cor preta (*G, H, I*) tiveram esgotamento energético e por isso não pode receber pacotes. [Yu et al., 2001].

direção ao destino. A abordagem também utiliza o encaminhamento geográfico recursivo e um mecanismo de difusão controlado para disseminar informações em uma região específica da área de sensoreamento. O processo de encaminhamento se dá em duas etapas. Na primeira etapa, o algoritmo encaminha pacotes a uma determinada região quando existe pelo menos um nó próximo ao destino desejado. Caso não haja vizinhos com distância menor que a do próprio nó, significa que há uma região de buracos e ser vencida, conforme ilustrado na Figura 3.2. Neste caso, um mecanismo de aprendizagem de custo em relação ao próximo salto é utilizado. A segunda etapa é a disseminação de pacote em uma determinada região. Nesta etapa, quando um o pacote alcança uma região desejada um algoritmo de encaminhamento geográfico recursivo é utilizado para verificar se a densidade de nós na dada região é alta ou não. Caso a densidade seja alta, o algoritmo recursivo cria cópias do pacote e divide a região em quatro sub-regiões, dando uma cópia a cada região criada. As sub-regiões criadas são novamente dividas até que uma sub-região contenha apenas um nó. Desta forma, os nós em uma região específica obterão de fato uma cópia do pacote. De outra forma, se a densidade for baixa, um mecanismo de difusão controlada será utilizado ao invés do algoritmo recursivo.

Por fim, há dois tipos de custo que o GEAR considera: (1) custo estimado utilizado no encaminhamento simples, referente ao nível de energia residual e na distância e; (2) custo do aprendizado de rotas utilizado para lidar com regiões de buraco.

3.1.3 Locating and Bypassing Routing Holes in Sensor Networks (BOUNDHOLE)

O algoritmo BOUNDHOLE [Fang et al., 2004], considera como foco principal de sua abordagem a ocorrência do fenômeno dos mínimos locais, em relação às soluções de encaminhamento guloso. Na abordagem, os autores consideram a existência de um beco sem

saída de forma que os nós localizados nesta região estão impossibilitados de encaminhar suas informações ao destino. A abordagem implementa uma regra local de percepção da região de lacuna chamada de *TENT rule*, utilizada para testar se o nó está em um beco sem saída (ou região de buraco). Essa regra é aplicada à topologia de rede existente e através dela cada nó mantém informações de todos os vizinhos imediatos (vizinhos de *1*–salto), no sentido anti-horário. Conforme ilustrado na Figura 3.3, para cada par de nós adjacentes *uv* em relação a *x*, a bissetriz perpendicular de *ux* e *vx* será traçada de forma que haja uma interseção no ponto *O*. A abordagem necessita somente das informações sobre ângulo em relação a seus vizinhos de *1*–salto para proceder com a detecção.

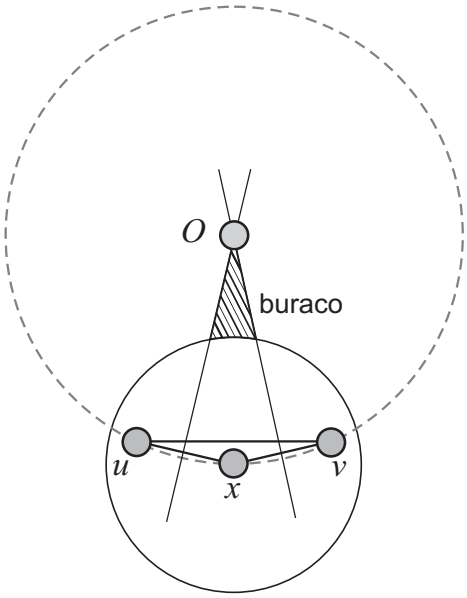


Figura 3.3. Regra local de detecção de uma região de buracos. O círculo sólido representa o alcance da comunicação do nó *x*. O círculo pontilhado representa a circunferência $\Delta\, uxv$. Baseado em [Fang et al., 2004].

3.1.4 Virtual Coordinate Assignment Protocol (VCap)

O *Virtual Coordinate assignment protocol (VCap)* apresentado em [Caruso et al., 2005], define um sistema de coordenadas para distâncias computadas em saltos. O protocolo assume que cada nó é estático e tem o mesmo alcance de transmissão. Além disso, cada nó tem um identificador único (ID) utilizado para evitar laços, que podem ser facilmente removidos, permitindo a cada nó escolher uma faixa de transmissão A implementação do VCap está dividida em quatro fases baseadas em mensagens broadcast com o objetivo de identificar três nós ancora (X, Y e Z), dinamicamente definidos durante a inicialização. O algoritmo atribui a cada nó um trio de coordenadas dadas pela distância em saltos dos nós

âncoras. Quando a última fase é completada, cada nó irá utilizar o trio de coordenadas atribuídas. No VCap, o sistema de coordenadas utiliza um protocolo de roteamento geográfico para entregar os pacotes às zonas de destino. A sobrecarga de armazenamento para cada nó é limitada pelo armazenamento do trio de coordenadas atribuído. A abordagem ainda é capaz de lidar com regiões de buraco. Neste caso, na ocorrência do fenômeno o pacote que está sendo enviado irá seguir um mecanismo de detecção e desvio chamado *local detour*, utilizado para aprimorar a fase de encaminhamento guloso, de forma que o progresso do pacote que será desviado (ao redor do buraco) seja garantido e, se for o caso, mudar de rota caso um melhor caminho não seja encontrado.

3.1.5 Greedy Forwarding with Virtual Position (GF–ViP)

O *Greedy Forwarding with Virtual Position (GF–ViP)* [You et al., 2009b] foi desenvolvido para melhorar a taxa de Encaminhamento Guloso em RSSFs esparsas, com pequenos buracos encontrados ao longo do Encaminhamento. O algoritmo foi proposto com dois complementos, o *Virtual Multi-Level Position (MVP)* e o *Greedy Forwarding with Virtual Hierarchical Position (HVP)*. Cada algoritmo possui duas variações que utilizam algoritmos Gulosos baseado em progresso *Most Forwarding Within Radius (MFR)* [Takagi & Kleinrock, 1984].

A aplicação do encaminhamento guloso com a detecção de regiões de buraco durante todo o processo de roteamento é uma das principais vantagens desta solução. Neste sentido, o método explora o posicionamento virtual através da criação de um ponto médio em relação a todos os vizinhos diretos. Desta forma, a informação sobre o posicionamento é armazenada nos próprios nós e em seus vizinhos diretos. Cada nó possui informações sobre sua posição virtual e a de seus vizinhos. O mecanismo de desvio de buracos foi ilustrado na Figura 3.4. Ao invés de comparar as localizações geográficas atuais de cada nó, o algoritmo utiliza posicionamento virtual para selecionar o próximo salto ao redor do buraco. Nesta etapa, cada só calcula sua posição e transmite ao seu vizinho virtual direto. Assim, o nó virtual conhece sua posição e a posição dos vizinhos virtuais. Esse posicionamento virtual de um nó fornece uma indicação de como os vizinhos diretos são localizados ao redor do nó. Isto será utilizado para mostrar a tendência de encaminhamento geográfico durante o processo de encaminhamento. Neste caso, os nós sensores precisam interagir com os seus vizinhos diretos para obterem as informações virtuais de posicionamento. Por fim, é importante frisar que o excesso de interações entre vizinhos localizados ao longo do perímetro de um buraco de roteamento pode levar ao aumento do tamanho do buraco.

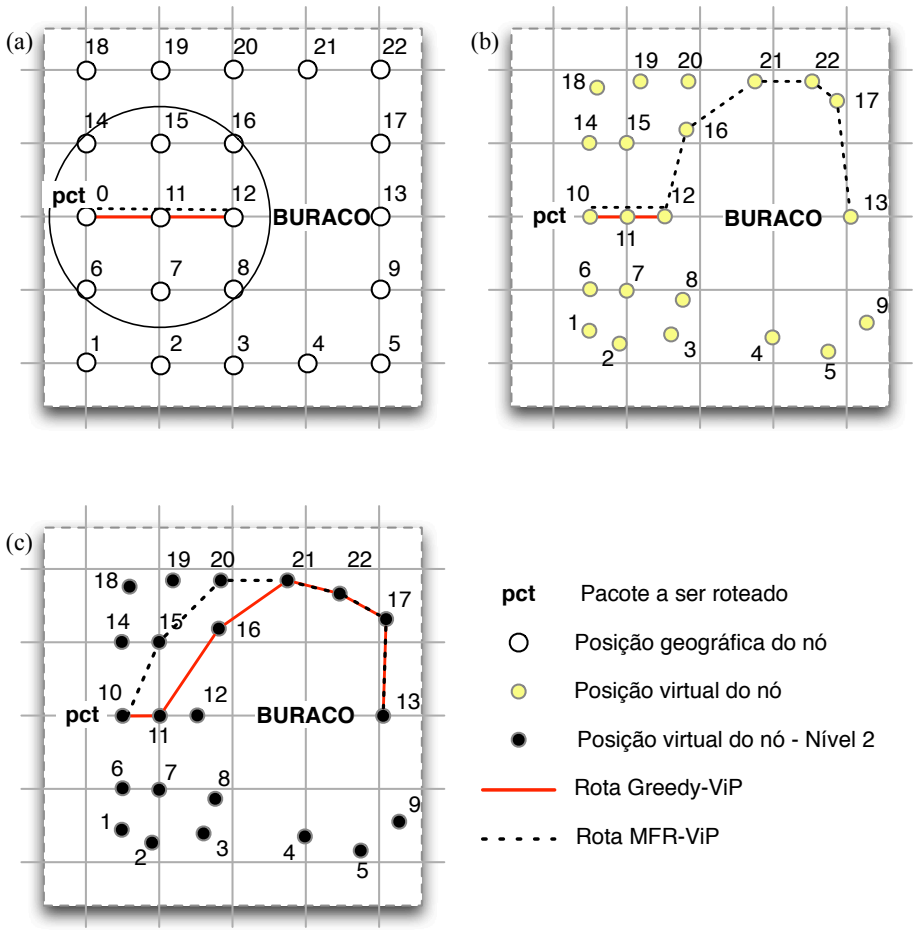


Figura 3.4. Desvio de buraco usando posicionamento virtual. (a) Nó 12 simulando um buraco tanto para Greedy quanto para MFR. (b) Caminho que pode ser encontrado através do algoritmo MFR-ViP. (c) Caminhos em volta do buraco. Baseado em [You et al., 2009b].

3.1.6 Hole Detection and Adaptive Geographical Routing (HDAR)

Os autores Yang & Fei [2010] propuseram um algoritmo de roteamento geográfico oportunista capaz de detectar buracos em RSSF chamado HDAR. O foco do algoritmo é lidar com o fenômeno dos mínimos locais através do conhecimento sobre o alcance de comunicação dos nós na região de buraco. A abordagem considera que se o ângulo entre dois nós localizados na borda do buraco for maior do que 120 graus, então o algoritmo inicia a etapa de detecção de buracos. Ao invés da distância Euclidiana, o algoritmo considera a relação de distância da rede inteira como métrica para a detecção do buraco. O algoritmo ainda implementa um mecanismo de anúncio para garantir que nós localizados nas proximidades do buraco e que serão afetados de forma significativa (mudança de rota) sejam informados sobre a existência de uma região de buracos. Assim, os nós afetados poderão ajustar de

forma adaptativa seus próximos salto em direção ao destino, evitando as regiões de buraco.

3.1.7 A Distributed Coverage Hole Repair Algorithm for Wireless Sensor Networks (HORA)

A abordagem distribuída *HOle Repair Algorithms (HORA)* [Sahoo & Liao, 2015], considera a mobilidade limitada dos nós para selecionar nós móveis baseado nas informações de cobertura da rede. A bordagem considera que os nós foram implantados de forma aleatória a rede está completamente conectada. Devido à implantação aleatória, é possível que um número considerável de nós sobreponha outros nós já implantados em algumas partes da rede. Desta forma, para detectar regiões de buraco na rede, os nós com grau maior de cobertura são movidos para manter a densidade uniforme sem aumentar o grau de cobertura dos vizinhos de um nó móvel. A fim de manter a cobertura da rede equilibrada e uniforma, o algoritmo HORA considera que a conexão entre os vizinhos imediatos é estável e não há distúrbios na grade. Um outro fator considerado para o sucesso da abordagem é assumir que a cobertura de um nó móvel nunca é alterada e não haverá alterações de configuração do nó mesmo este sendo um nó móvel.

3.1.8 Void Aware in Geographic Routing for Wireless Sensor Networks (VAGR)

Uma solução para lidar com lacunas de comunicação, que levam à ocorrência de buracos em RSSF, foi apresentada e discutida por Revathi & Malathy [2016]. O foco principal da solução é formar um círculo virtual através das coordenadas de cada nó para auxiliar o repasse de informações a partir de nós localizados na borda da região de buracos (lacuna). Através da utilização deste círculo virtual é possível evitar falhas na utilização do mecanismo de encaminhamento guloso, que não opera na presença de buracos. O mecanismo utilizado para lidar com lacunas na comunicação do VAGR ocorre em três etapas que abrangem a detecção da lacuna, o mapeamento de coordenadas virtuais e a fase de divisão de regiões de interesse. Na primeira etapa, enquanto o algoritmo está operando com o mecanismo de encaminhamento guloso e detecta um buraco, o nó que se encontra na região do buraco (onde o encaminhamento guloso falhou) é definido como nó de descoberta, pois descobriu (identificou) o buraco. Este nó gera uma cópia dos dados a serem transmitidos e armazena temporariamente este pacote. A partir de então, este nó gere um pacote de detecção de buraco que será utilizado para percorrer a borda do buraco até retornar ao nó de descoberta novamente. Na etapa de mapeamento de coordenadas virtuais, as coordenadas armazenadas pelos nós da borda na primeira etapa são agora utilizadas para criar um sistema de coorde-

nadas virtuais organizado de tal forma que entre a origem do pacote e o destino final não haja na estrutura de roteamento nenhuma lacuna, ou região de buracos. Por fim, na terceira etapa o algoritmo considera a posição da região de buracos e a localização do destino dos pacotes para iniciar um processo de divisão da região em sub-regiões que serão utilizadas para o encaminhamento de pacotes.

3.1.9 An Energy Efficient and Load Balanced Distributed Routing Scheme for Wireless Sensor Networks with Holes

Uma abordagem de roteamento que explora o conhecimento prévio sobre buracos de roteamento para criar rotas de escape a partir da cooperação entre vizinhos próximos ao buraco foi proposta por Nguyen et al. [2017]. Durante o funcionamento da abordagem, os nós regulares cooperam para determinar para determinar um polígono em um dado buraco para a partir daí disseminar informação entre os nós que se encontram dentro e ao redor do buraco. A técnica utilizada na abordagem foi concebida originalmente para aumentar a eficiência energética. As informações disseminadas previamente são controladas por parâmetros de comunicação pré-definidos e são utilizadas para ajustar o processo de encaminhamento de mensagens em torno do buraco. A solução foi capaz prolongar o tempo de vida da rede, enquanto gerava caminhos de roteamento eficientes ao redor do buraco. Neste sentido, durante o processo de encaminhamento de pacotes os pacotes transmitidos podem evitar o buraco de roteamento e a região deste buraco. A abordagem considera que cada nó conhece sua própria posição física e a de seus vizinhos de um salto de distância. Além disso, a localização do destino final dos pacotes também é conhecida pelos nós que irão originar o processo de envio de pacotes. Para facilitar a identificação e citação deste trabalho, criamos o acrónimo EEDRS para referenciá-lo.

3.1.10 An Enhanced Location–Free Greedy Forward Algorithm with Hole Bypass Capability in Wireless Sensor Networks (RSSR)

A ideia chave do algoritmo RSSR [Oliveira et al., 2015], é aproveitar a vantagem da grande capacidade do *sink* e equipar este nó com um dispositivo de comunicação potente, de forma que o pacote seja enviado a todos os nós da rede em um único salto. O nós regulares da rede podem replicar mensagem do *sink* usado comunicação em múltiplos saltos. Os nós que participam do processo de encaminhamento das mensagens são capazes de detectar se estão em uma região de buracos. O funcionamento da abordagem se dá da seguinte forma: A mensagem do sink atinge os nós da rede com diferentes valores de RSSI de modo os nós mais distantes tenham indicação da força do sinal baixo durante a propagação. O objetivo é

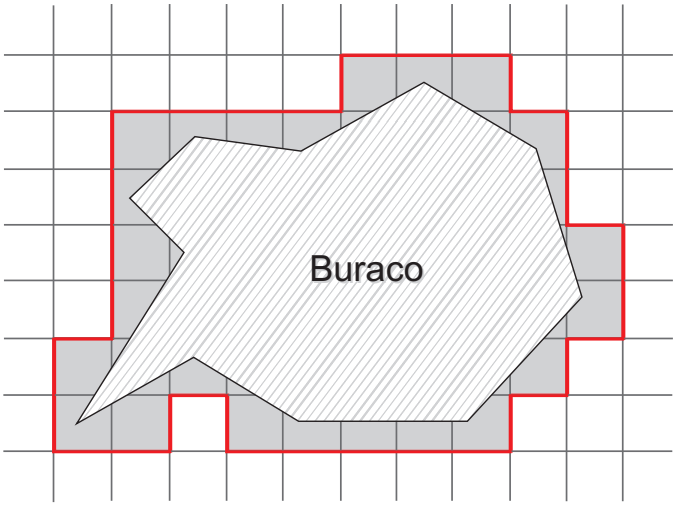


Figura 3.5. Etapa de aproximação executada a partir das informações dos nós da borda do buraco. A linha vermelha representa a borda do polígono aproximado. Baseado em [Le et al., 2012].

encaminhar o pacote ao vizinho que recebe a mensagem com o sinal mais forte, na teoria, o vizinho mais próximo do *sink*. Baseado nesta informação, duas variações do algoritmo são propostas: *RSSR-Election* e *RSSR-Selection*. Na primeira abordagem, um algoritmo de eleição de líder, que não requer troca de pacotes foi proposto para a escolha do próximo salto. O objetivo é que os nós próximos do *sink*, que receberam RSSI alto, irão esperar por um curto período de tempo antes de serem eleitos como líder, e o nós mais distantes do *sink* irão eleger a si como o próximo salto após um período tempo mais longo. Quando um nó elege a si como o próximo salto, ele imediatamente encaminha o pacote replicado ao *sink*. Na segunda abordagem, nós vizinhos trocam informações e o vizinho com valor de RSSI mais elevado é selecionado. Quando o *sink* envia a primeira mensagem, todos os nós trocam informações com seus vizinhos, estimando suas distâncias de transmissão através de anúncios. Um nó recebe um anúncio de cada vizinho e grava essa informação em uma tabela de roteamento, Assim, o nó checa se a mensagem é para ele e, se for o caso, espera um período curso de tempo, envia uma réplica para o vizinho próximo do *sink* e salva o procedimento em uma tabela de Encaminhamento de pacotes.

O algoritmo RSSR, que permite a execução da estratégia de encaminhamento guloso e não requer nenhuma informação de localização ou coordenadas virtuais, também é capaz de lidar com regiões de buracos. A abordagem especifica como um nó pode perceber se está ou não na borda de uma região de buracos e ainda como encaminhar pacotes até o destino na ocorrência de um buraco. Os autores consideram que para um nó pertencer a uma região de buracos, duas checagens são importantes: (i) um nó não possuir pelo menos um vizinho mais próximo ao nó *sink*; e (ii) um nó possuir vizinhos mais próximos, mas todos pertencentes à

região de buracos.

3.2 Classificação dos Trabalhos Apresentados

As soluções apresentadas e discutidas neste trabalho foram classificadas a fim de se destacar características compartilhadas entre as abordagens como ciência energética, detecção e desvio de buracos e informações sobre posicionamento e/ou localização. A classificação é mostrada na Tabela 3.1.

	Coord. Virtual	GPS	Ciente de Energia	Considera a densidade	Nó divulga sua posição
GPSR [Karp & Kung, 2000]		√	√		
GEAR [Yu et al., 2001]		√	√		
BOUNDHOLE [Fang et al., 2004]		√			√
VCap [Caruso et al., 2005]	√			√	√
GF-ViP [You et al., 2009a]	√				√
HDAR [Yang & Fei, 2010]		√			
HORA [Sahoo & Liao, 2015]		√	√	√	
RSSR [Oliveira et al., 2015]			√	√	√
VAGR [Revathi & Malathy, 2016]	√		√		
EEDRS [Nguyen et al., 2017]		√	√		

Tabela 3.1. Classificação das abordagens apresentas

3.3 Comentários Finais

Neste capítulo, apresentamos os trabalhos mais relevantes à nossa pesquisa em curso. Verificamos que, com exceção da abordagem RSSR, disponível em [Oliveira et al., 2015], todas as abordagens de roteamento fazem uso de algum sistema de atribuição de coordenadas virtuais ou algum outro sistema de coordenadas. Nossos algoritmos são capazes de construir rotas até o destino final dos pacotes a partir dos valores de RSSI divulgados através de um *sink* com grande capacidade de comunicação. Desta forma, pretendemos demonstrar a eficácia de nossas abordagem através dos resultados obtidos e apresentados nos próximos 04 capítulos.

Capítulo 4

Agregação de Dados com Desvio de Buracos para Redes de Sensores Sem Fio com *Sinks* de Alto Alcance

Um dos pontos fracos na utilização de algoritmos que exploram informações geográficas em RSSFs é a necessidade de informações sobre a localização dos nós na grade. Neste sentido, se faz necessária a utilização de um sistema de localização que em muitas ocasiões pode ser custoso e susceptível a falhas. Erros de localização são variáveis *gaussianas* independentes que consideram que a variação do erro em cada nó regular pode variar de forma aleatória [Éfren Souza et al., 2012].

Neste capítulo, propomos algoritmo de roteamento geográfico capaz de detectar e contornar regiões de buracos em RSSF chamado ARESTA (Agregação de Dados com Desvio de Buracos para Redes de Sensores Sem Fio com *Sinks* de Alto Alcance). A abordagem utiliza um aprimoramento do mecanismo de encaminhamento guloso clássico para lidar com regiões de buracos e ainda prover agregação de dados durante o processo de encaminhamento dos pacotes ao destino final. A abordagem considera que o nó *sink* está equipado com um dispositivo de comunicação de grande alcance de tal forma que em um único salto todos os nós da rede sejam alcançados. O retorno dos nós sensores para o nó *sink* é realizado através de múltiplos saltos, utilizando uma nova técnica de encaminhamento guloso que utiliza os valores do RSSI (*Received Signal Strength Indicator*) para estimar distâncias entre os nós, escolher o próximo salto em direção ao nó *sink*, calcular o tempo para o envio de pacotes que possibilita a agregação de dados e ainda detectar regiões de buraco. A confiabilidade do RSSI , utilizado como métrica para estimar posições e distâncias entre nós já foi discutida em trabalhos anteriores [Dan et al., 2011, Holland et al., 2006, Huo & Wang, 2008, Jacinto, 2012, Zhu et al., 2009].

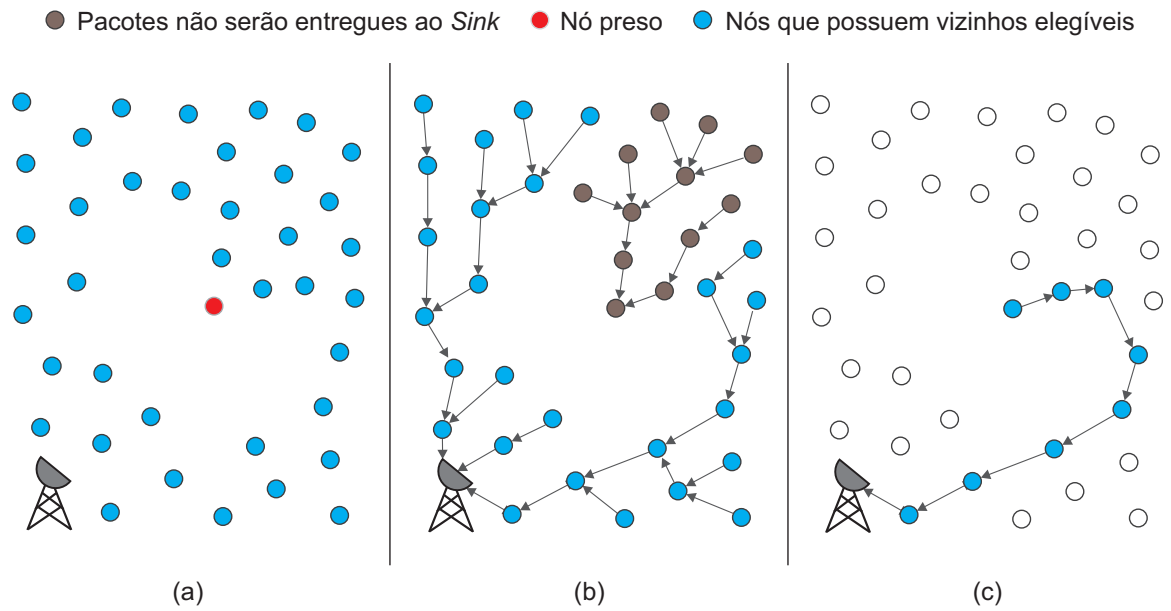


Figura 4.1. (a) Nó da região de buracos não possui vizinho elegível a próximo salto e é marcado como nó preso. (b) Após temporizador expirar nós encaminham suas pacotes de dados a vizinhos elegíveis. (c) Regra de desvio aplicada: Conforme a configuração da grade, o nó irá optar por um vizinho que supre as exigências para o encaminhamento

4.1 Algoritmo ARESTA

Na presente seção apresentamos o algoritmo ARESTA. A abordagem considera a possibilidade de se equipar o nó *sink* com um dispositivo de comunicação de alto alcance de forma que em um único salto o nó *sink* pode enviar uma consulta para toda a rede, aproveitando o grande alcance na comunicação. O algoritmo ARESTA também é capaz de detectar e desviar buracos de roteamento, traçando rotas alternativas até encontrar um vizinho que esteja geograficamente próximo ao destino, em nosso caso o *sink*, conforme ilustrado na Figura 4.1. Este novo algoritmo de roteamento geográfico utiliza o mecanismo de encaminhamento guloso e foi desenvolvido para alcançar padrões como confiabilidade, escalabilidade e altas taxas de entrega, sem a necessidade da informação de posicionamento dos nós sensores, além de executar agregação de dados durante o processo de encaminhamento dos pacotes ao *sink*.

A abordagem, descrita no Algoritmo 1, inicia quando o nó *sink* envia uma consulta para toda a rede (linha 10). Os nós sensores, ao receberem essa consulta, estimam suas distâncias ao nó *sink* usando a técnica de RSSI (linhas 11 e 12), conforme já ilustrado anteriormente. Em seguida, cada nó irá enviar a seus vizinhos um pacote de anúncio, que contém a sua distância estimada para o *sink* (linha 13). Ao mesmo tempo, os nós receberão diversos pacotes deste tipo, um de cada vizinho, e salvarão tais dados de distâncias nas suas tabelas

Algoritmo 1 ARESTA

▷ **Variables:**
1: $tabelaVizinhos_i = \emptyset$; {Distâncias dos vizinhos ao *sink*}
2: $dadosAgregados_i = \emptyset$; {Conjunto de dados agregados}
3: $dadosConsulta_i$; {Dados obtidos a partir da consulta do *sink*}
4: $tempoEnvio_i$; {Tempo de espera para envio da resposta}
5: $temporizador_i$; {Temporizador usado na agregação de dados}
6: $proximoSalto_i$; {Próximo salto para o envio dos dados agregados}
7: $saltoElegivel_i$; {Próximo salto elegível em caso de desvio de buraco}
8: $dist_i$; {Distância para o *sink*}

▷ **Input:**
9: *Sink* recebe da Central de Monitoramento uma $consulta(consultaId_k, consulta_k)$;

Action:
10: Envia $consulta(consultaId_k, consulta_k)$ para todos os nós.

▷ **Input:**
11: $msg_i = consulta(consulta_k, consulta_k)$;
12: $dist_i = estimaDistancia(msg_i)$;
Action:
13: Envia $anuncio(n_i, dist_i)\ \forall\ n_j \in vizinho_i$. {Divulgando a distância}
14: **if** $dadosConsulta_i = recRetornoConsulta(consulta_k)$ **then**
15: $dadosAgregados_i := dadosAgregados_i \cup dadosConsulta_i$; {Agrega informação ao pacote}
16: **end if**
17: $tempoEnvio_i = raioMaxComunicSink - dist_i$; {Tempo de agregação}
18: $temporizador_i.inicia(tempoEnvio_i)$; {Inicia temporizador}

▷ **Input:**
19: $msg_i = anuncio(vizinho_k, dist_k)$;
20: $tabelaVizinhos_i = tabelaVizinhos_i \cup (vizinho_k, dist_k)$;
21: $proximoSalto_i = vizinhoProximoAoSink()$;
Action:
22: **if** $dist_i < proximoSalto_i.dist$ **then** {Estou em um buraco de roteamento?}
23: $saltoElegivel_i := desvioDeBuraco(tabelaVizinhos_i)$; {Qualifica o próximo salto como elegível}
24: Envia $resposta(dadosAgregados_i)$ para $saltoElegivel_i$;
25: **end if**

▷ **Input:**
26: $temporizador_i.expirou()$; {Tempo para agregação expirou}
Action:
27: Envia $resposta(dadosAgregados_i)$ para $proximoSalto_i$; {Encaminha dado agregado}
28: $dadosAgregados_i = \emptyset$;

▷ **Input:**
29: $msg_i = resposta(dadosAgregados_k)$;
Action:
30: **if** $n_i \neq 0$ **then** {Sou um nó regular?}
31: $dadosAgregados_i := dadosAgregados \cup dadosAgregados_k$;
32: **if** $temporizador_i.inativo()$ **then** {O dado foi enviado?}
33: Envia $resposta(dadosAgregados)$ para $proximoSalto_i$;
34: $dadosAgregados_i = \emptyset$;
35: **end if**
36: **else** {Eu sou o nó *Sink*}
37: Envia o dado recebido à Central de Monitoramento.
38: **end if**

de roteamento (linha 19 e 20). Cada nó irá verificar se possui dados a serem retornados à consulta do *sink*. Se tiver, tais dados serão adicionados à lista de dados agregados (linhas 14–16). O próximo passo do algoritmo é calcular um tempo de espera para agregação dos dados antes de enviar a resposta ao *sink*. Neste sentido, um temporizador é iniciado em cada nó com o tempo de espera calculado a partir dos valores obtidos pelo RSSI do *sink* (linhas 17 e 18). Este tempo será utilizado para habilitar o início envio dos pacotes de dados e é definido

de acordo com a potência do sinal recebido por cada nó, ou seja, quanto mais baixo o RSSI, menos tempo o nó terá que esperar para enviar seus pacotes em direção ao *sink*. O parâmetro utilizado para o cálculo estimado da distância foi dado em metros e o tempo utilizado na agregação de dados foi dado em segundos. O temporizador considera que cada metro corresponde a um segundo. Desta forma, cada nó irá esperar um tempo em segundos antes de encaminhar suas informações. Após a etapa de configuração, as informações agregadas a serem encaminhadas ao próximo salto são atualizadas (linha 21).

Antes do temporizador de um nó expirar, este checa se há um vizinho mais próximo ao *sink* com base em sua tabela de vizinhos construída na etapa de estimativa de distâncias entros os nós da grade. Caso contrário, o algoritmo considera que este nó se encontra em uma região onde a utilização da técnica de encaminhamento guloso não é possível, ou seja, em uma região de buraco. O nó corrente então executa nosso mecanismo de detecção e contorno de buracos (linha 23), que seleciona um novo vizinho em direção ao *sink* a partir de uma tabela vizinhos. O mecanismo de detecção de contorno de buracos permite que o próximo salto seja eleito de acordo com a distância do *sink* em relação a cada vizinho, considerando como elegível o vizinho geograficamente mais próximo do *sink* em relação ao nó preso no buraco.

O nó preso nesta região de buraco não possui vizinhos elegíveis para o encaminhamento de pacotes através da técnica de encaminhamento guloso. Desta forma, o encaminhamento de um pacote contornando o buraco (linha 24) é feito de acordo com as distâncias dos vizinhos armazenadas na tabela de roteamento de cada nó, de forma que o vizinho a ser escolhido como próximo salto esteja o mais distante possível em relação ao nó corrente, e o mais próximo possível em relação ao *sink*. Esta mesma mensagem contém o identificador de cada mensagem encaminhada ao nó que na região de buracos. Tais informações serão utilizadas para atualizar a tabela de roteamento dos nós que permite a escolha do próximo salto. Este processo é repetido até que seja escolhido um vizinho apto a executar o encaminhamento guloso.

Quando o temporizador de um nó expira (linha 26), um pacote com todos os dados agregados é enviado ao vizinho mais próximo do nó *sink* (linha 27). Este pacote será recebido pelo vizinho escolhido (linha 29) que, provavelmente, não terá seu temporizador expirado ainda, por estar mais próximo do *sink*. Este último nó irá simplesmente agregar os dados recebidos em sua lista de dados agregados (linha 32) e continuar esperando que seu temporizador expire para enviar esses dados adiante (linhas 26–28). Finalmente, quando o *sink* recebe algum pacote de resposta, ele envia os dados recebidos de volta à central de monitoramento (linhas 37).

Em alguns casos raros, é possível que o temporizador de um determinado nó tenha expirado (e o seu dado agregado já tenha sido encaminhado) antes de receber todos os dados

dos seus vizinhos. Isso pode ocorrer na presença de erros de RSSI elevados, propagados durante o processo estimativa de distância e configuração do temporizador para a agregação de dados. Caso isto ocorra, o pacote recebido é encaminhado imediatamente ao vizinho mais próximo do *sink*, através da técnica de encaminhamento guloso (linhas 32–35). O ARESTA implementa agregação de dados com forma de reduzir o número de informações redundantes e aumentar o tempo de vida da rede, também proporcionado pela economia de energia que é um benefício direto da agregação de dados.

4.2 Avaliação de Desempenho

Nesta seção, avaliamos o algoritmo ARESTA e comparamos sua performance com o algoritmo RADR, discutido por Lima et al. [2013b]. A escolha do algoritmo RADR foi feita a partir da constatação de que abordagens envolvendo torres altas considerando os valores de RSSI propagados é inovador e possui rara base de comparação. Neste sentido, como um de nossos trabalhos prévios, o RADR, trata do mesmo cenário e já foi comparado a outros algoritmos de roteamento geográfico, seus resultados foram utilizados como base de conhecimento para a validação da primeira hipótese de nossa pesquisa.

4.2.1 Metodologia

A avaliação de desempenho foi realizada através de simulações usando o simulador Sinalgo ETH-Zurich [2014], que provê um ambiente completo para simulação de algoritmos distribuídos. Em nossos algoritmos, consideramos o modelo de interferência SINR para simular a perda de pacotes durante o processo a execução dos algoritmos. As simulações foram realizadas em um campo de sensoreamento de $130\,\mathrm{m} \times 130\,\mathrm{m}^2$. O posicionamento dos nós foi feito com base em uma grade perturbada, mantendo a densidade de $0.03\,\mathrm{nós/m}^2$. Em relação aos resultados, as curvas representam valores médios, enquanto as barras de erro representam intervalos de confiança para 95 % a partir de 33 instâncias independentes (sementes aleatórias).

Os parâmetros de simulação foram baseados no sensor *MICAz* e os valores utilizados são mostrados na Tabela 4.1. É importante observar, que esta tabela de parâmetros expressa valores de referência para se analisar as diferentes métricas propostas. Neste sentido, quando pretendíamos avaliar o impacto da escalabilidade, por exemplo, os parâmetros referentes à quantidade de nós eram manipulados (dispostos em um intervalo) enquanto que o restante dos parâmetros permanecia fixo, de acordo com o proposto e descrito na tabela de configuração.

Parâmetro	Valor utilizado
Área monitorada	$130\,\text{m} \times 130\,\text{m}^2$
Quantidade de nós	512 nós (grade perturbada)
Posição do Sink	Canto inferior esquerdo
Densidade	$0.03\,\text{nós/m}^2$
Alcance da comunicação	30 m
Imprecisão do RSSI a curta distância	15 % da dist. real

Tabela 4.1. Configuração do cenário de simulação do algoritmo ARESTA.

Quanto à topologia da rede, nós assumimos que a implantação dos nós obedece a estrutura de uma grade perturbada em que a localização de cada nó sofre um pequeno distúrbio gerado por um erro gaussiano aleatório. O nó *sink* foi posicionado no canto inferior esquerdo enquanto que os nós regulares ocupam a grade uniformemente.

Para a execução dos algoritmos apresentados nesta tese de doutorado, a grade de sensoreamento foi configurada com a inserção de um polígono irregular côncavo em forma de *L*, localizado no centro desta grade, de tal forma que os nós de um lado do buraco não consigam se comunicar com os nós do outro lado, ou seja, a parte oposta do buraco, conforme ilustrado na Figura 4.2. Tal configuração pode ser considerada como o pior cenário, já que um buraco baseado em uma circunferência não cria uma lacuna de comunicação nem mesmo denota o fenômeno dos mínimos locais. Outros tipos de polígonos não podem criar cenários de pior caso para o problema explorado neste trabalho.

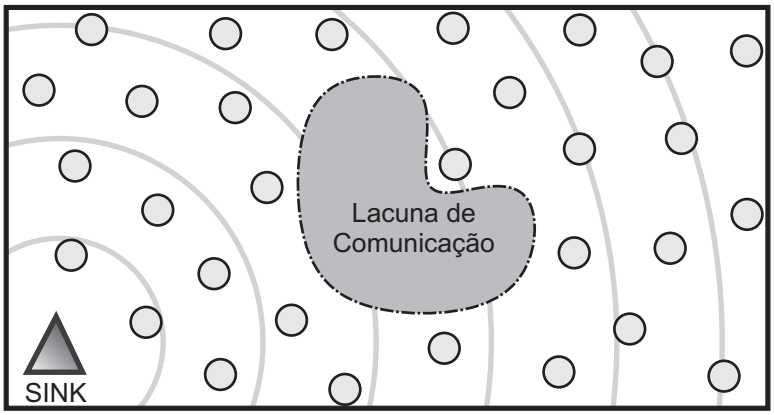


Figura 4.2. Cenário base de implementação de todos os algoritmos descritos nesta tese.

Durante as simulações, cada nó sensor gera um pacote de dados com eventos detectados. Neste aspecto, a economia de transmissões foi alcançada através da função de agregação. Esta função recebe um conjunto pacotes de entrada e retorna apenas um pacote com os dados agregados do conjunto de pacotes. Esta função é efetuada em cada salto de rede. Além disso, o número de pacotes encaminhados sem a utilização da agregação de dados também foi registrado.

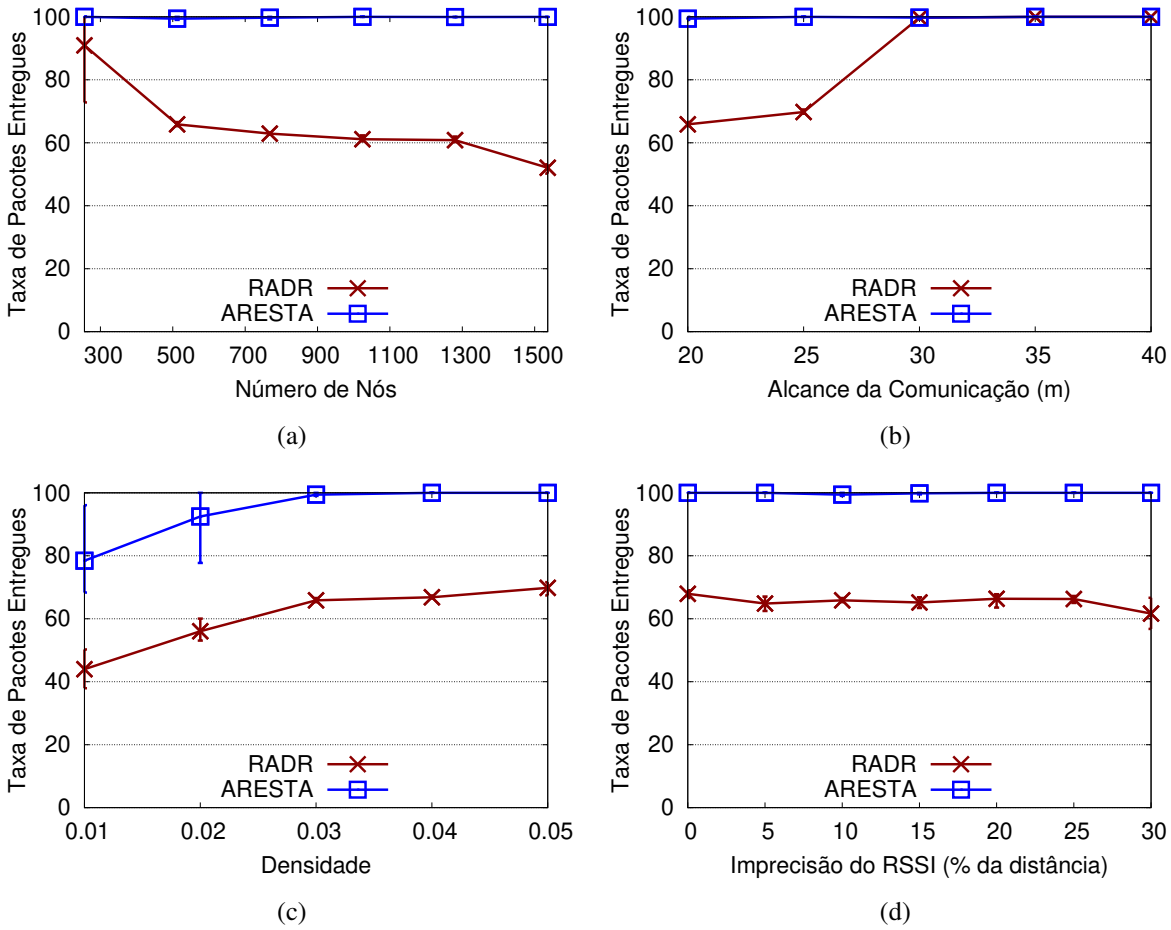


Figura 4.3. Resultados obtidos ao se avaliar a taxa de entrega.

Para avaliar o desempenho do algoritmo ARESTA utilizamos quatro métricas, descritas a seguir:

- **Pacotes entregues:** razão entre o número total de pacotes de dados gerados pelos nós sensores e a quantidade de pacotes de dados recebidos pelo nó *sink*;

- **Economia de transmissões:** razão entre o número de transmissões que não foram necessárias devido à agregação de dados e o número de transmissões que seriam efetuadas sem agregar dados;

- **Acertos na escolha do próximo salto:** um acerto ocorre quando um pacote sai de um nó em direção ao seu vizinho, geograficamente mais próximo do nó *sink*, enquanto que um erro se dá quando este pacote foi enviado para um vizinho que não é o mais próximo do *sink*, ou seja, ele poderia ter sido enviado para um vizinho melhor.

4.2.2 Impacto da Escalabilidade da Rede

A escalabilidade da rede foi avaliada através do aumento do número de nós na rede de 256 para 1536 nós. Este esta alteração na quantidade de nós atinge diretamente o tamanho da área monitorada de modo a manter a densidade constante, ou seja, em 0.03 nós/m^2. Na Figura 4.3(a), é possível observar que o algoritmo RADR não consegue entregar todos os dados, uma vez que não é capaz de lidar com buracos de roteamento. Neste sentido, como já discutido anteriormente, não haviam trabalhos similares ou passíveis de comparação justa em relação ao algoritmo ARESTA. Assim sendo, o algoritmo ARESTA consegue entregar todos os pacotes na maioria das vezes uma vez que foi concebido para detectar e superar buracos de roteamento. O resultado demonstra claramente que nossa solução é capaz de operar na presença de buracos, até mesmo quando aumentamos o número de nós.

4.2.3 Impacto do Raio de Comunicação

Para avaliarmos o impacto do raio de comunicação dos nós em nosso algoritmo, aumentamos este parâmetro de 20 m até 40 m. Raios de comunicação pequenos reduzem o número de nós vizinhos, o que consequentemente reduz o número de possibilidades para o próximo salto [Boukerche et al., 2008]. A Figura 4.3(b), mostra que o número de pacotes entregues com sucesso do algoritmo RADR diminui para cerca de 65 % quando o raio de comunicação é 20 m e fica em torno de 70 % quando o alcance aumenta para 25 m. Neste caso, os nós possuem poucos vizinhos elegíveis, gerando ainda mais buracos de rede. Neste cenário, o algoritmo RADR interrompe o processo de encaminhamento guloso. Contudo, em todos os casos nosso algoritmo ARESTA é capaz de entregar todos os dados agregados.

4.2.4 Impacto da Densidade da Rede

O impacto da densidade da rede em nosso algoritmo foi avaliado começando com uma densidade relativamente baixa de 0.01 nós/m^2 e aumentando tal densidade até 0.05 nós/m^2. Em densidades baixas, um nó terá poucos vizinhos, dificultando a agregação e consequentemente o encaminhamento do pacote adiante, enquanto que ao se aumentar a densidade, o número de vizinhos de cada nó aumentará. Podemos observar pela Figura 4.3(c) que o algoritmo ARESTA continua funcionando corretamente e é capaz de operar de forma eficiente mesmo em baixas densidades ao contrário do algoritmo RADR, que em baixas densidades consegue entregar cerca de 40 % dos pacotes. Por fim, observamos que número de pacotes entregues com sucesso pelo algoritmo ARESTA é de cerca de 80 % quando a densidade está em 0.01 nós/m^2. Neste caso, os nós possuem poucos vizinhos e isso aumenta as chances de escolha de um salto elegível para o encaminhamento dos pacotes.

4.2.5 Impacto da Imprecisão do RSSI a Curta Distância

Neste trabalho avaliamos o impacto da imprecisão do RSSI a curta distância (estimativa de distância entre os nós regulares) aumentando essa imprecisão de 0 % para 30 % da distância real entre os nós. Como é possível observar na Figura 4.3(d), o algoritmo ARESTA não é afetado de forma significativa pela imprecisão do RSSI, além de se mostrar confiável no encaminhamento de pacotes ao *sink*. Por outro lado, o algoritmo RADR é afetado durante o processo de estimativa de distância entre os vizinhos prejudicando as decisões de próximo salto a partir de uma área com a presença de um buraco.

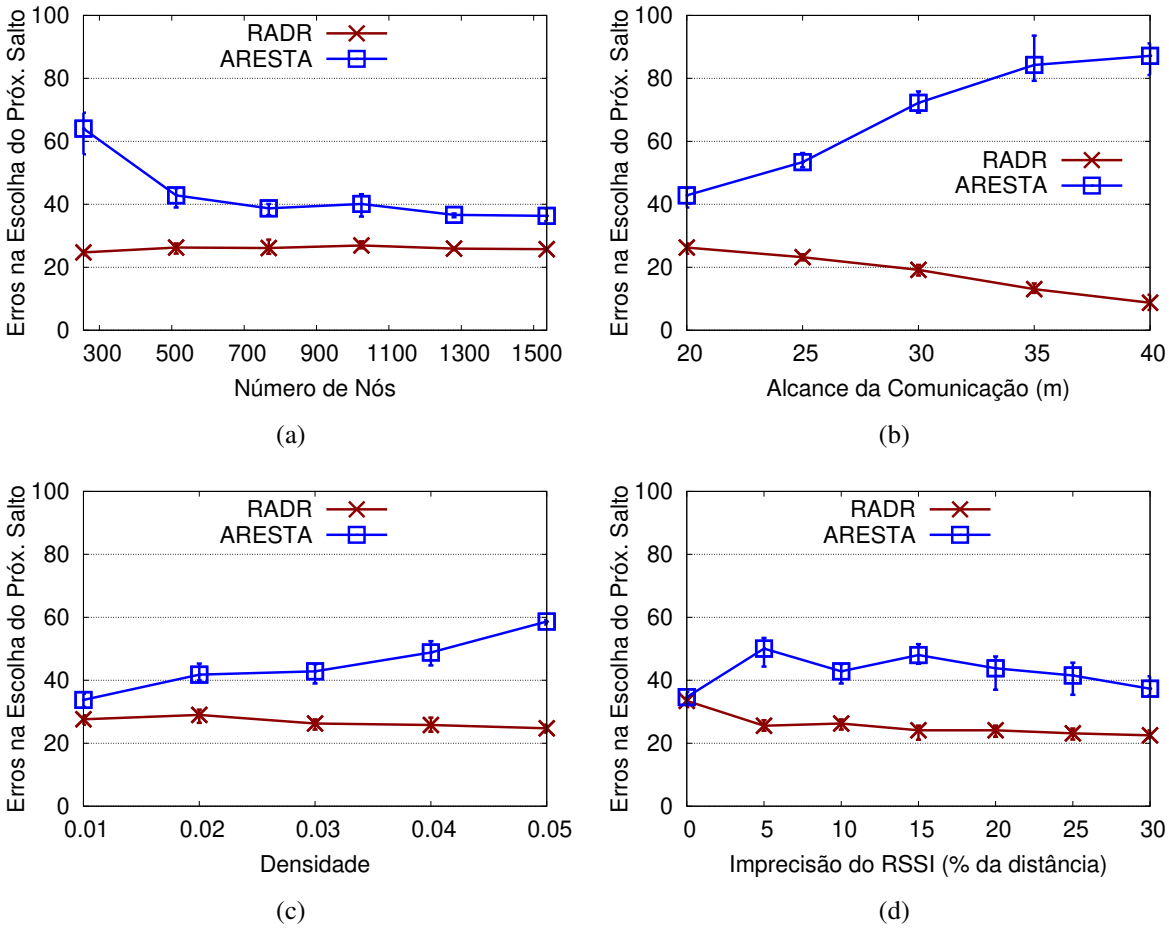


Figura 4.4. Resultados obtidos ao se avaliar as escolhas de próximo salto.

4.2.6 Impacto da Regra de Desvio de Buracos na Escolha do Próximo Salto.

O impacto da solução de desvio de buracos em relação à escolha do próximo salto também foi avaliado. Neste sentido, comparamos os resultados obtidos através da execução

dos algoritmos RADR e ARESTA na presença de um buraco de roteamento. A grade foi configurada com a inserção de um polígono côncavo irregular em forma de *L*, ao centro, de forma que os nós de um lado do buraco não se comunicassem com os nós do outro lado, portanto a parte oposta do buraco. Esta configuração pode ser considerada como o pior cenário, já que um buraco baseado em uma circunferência não cria um buraco de roteamento nem mesmo denota o fenômeno dos mínimos locais. Outros tipos de polígonos não podem criar cenários de pior caso para o problema explorado neste trabalho.

O número de escolhas erradas em relação ao próximo salto no algoritmo ARESTA é um pouco maior do que no algoritmo RADR com poucos nós, como mostram as Figuras 4.4(a), 4.4(b), 4.4(c) e 4.4(d), pois no desvio de buracos os pacotes são encaminhados a vizinhos que não estão geograficamente mais próximos do *sink* e sim na borda do buraco. Apesar disto, o algoritmo ARESTA é capaz de gerar árvores de escoamento eficientes enquanto em direção *sink*, contornando a borda do buraco.

4.3 Considerações finais

Neste capítulo, propusemos e apresentamos um algoritmo de roteamento geográfico que provê agregação de dados para redes de sensores sem fios, que chamamos de ARESTA (Agregação de Dados com desvio de buracos para Redes de Sensores Sem Fio com *Sinks* de alto alcance). O aspecto essencial desta abordagem é equipar o nó *sink* com um potente dispositivo de comunicação de forma que em um único um pacote de anúncio todos os nós da rede sejam alcançados. Com base apenas nos valores de RSSI obtidos pelos nós, propomos uma nova e eficiente abordagem o roteamento e agregação de dados, com a capacidade de lidar com buracos de roteamento.

O algoritmo ARESTA, é capaz de detectar e superar buracos em uma RSSF através da construção de uma rota alternativa até o vizinho mais próximo do *sink*. A agregação de dados ocorre durante o processo de encaminhamento dos pacotes e é possível graças à utilização de um temporizador configurado em cada nó regular após a consulta inicial recebida pelo nó *sink*. Uma série de experimentos foi realizada para avaliar a solução proposta. Os resultados demonstram claramente os benefícios introduzidos pela técnica de desvio de buracos, somada a abordagem simples de encaminhamento guloso e agregação de dados.

Capítulo 5

Roteamento Geográfico Oportunista e Agregação de Dados com Contorno de Buracos em Redes de Sensores Sem Fio

O mecanismo de encaminhamento guloso (*greedy forwarding*), utilizado pela maior parte das abordagens geográficas envolvendo comunicação em múltiplos saltos, permite que os nós escolhidos como próximo salto em direção ao destino final dos pacotes sejam eleitos de acordo com seu posicionamento geográfico em relação a esse destino Karp [2000]. Tal esquema de roteamento pode se adaptar facilmente às mudanças relacionadas à topologia da rede durante o processo de roteamento. Entretanto, os algoritmos de roteamento geográfico geralmente enfrentam duas desvantagens. A primeira é a necessidade de informações sobre a posição dos nós, que pode ser dispendiosa e ainda promover erros nas RSSFs. A segunda será a ocorrência do fenômeno dos mínimos locais (*local minimum*), uma vez que sua ocorrência compromete a entrega de pacotes ao *sink*.

Neste capítulo, propomos um algoritmo de roteamento geográfico que é capaz de detectar e superar regiões de buracos em RSSF, enquanto provê agregação de dados durante o processo de encaminhamento de pacotes ao *sink*. O algoritmo REACT considera que o *sink* utilizado na grade de sensoreamento possui grande capacidade de comunicação e alcança a toda a rede em um único salto.

5.1 Algoritmo REACT

Nesta seção, apresentamos o algoritmo REACT. Neste algoritmo de roteamento geográfico oportunista, uma nova técnica de encaminhamento às cegas, capaz de detectar buracos de roteamento em RSSF através de um mecanismo de auto-eleição do próximo salto é implementada. Através de nossa técnica de encaminhamento, o REACT é capaz de detectar

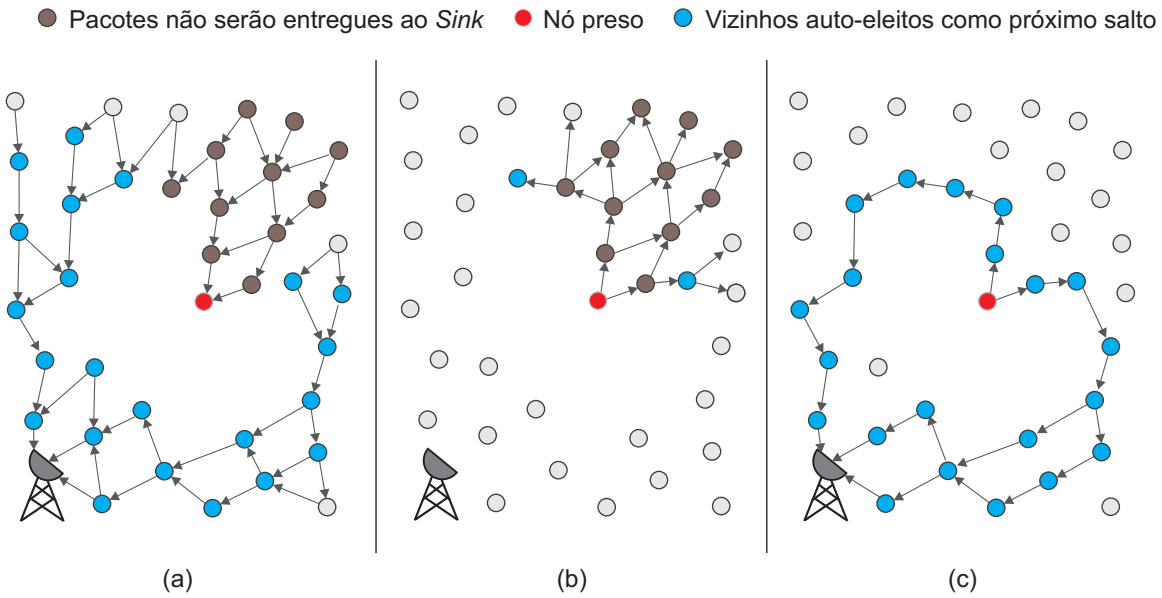


Figura 5.1. (a) Cada nó envia uma mensagem de *broadcast* após temporizador expirar. O nó preso (em vermelho) detecta que está em uma região de buracos, pois não possui vizinhos mais próximos ao *sink*. (b) Início do processo de transmissão às cegas após detecção de buraco. (c) Fluxo gerado a partir de nossa abordagem oportunista.

lacunas de comunicação e ainda superar tais regiões através de mensagens de anúncio (*broadcast*) que são tratadas pelos nós vizinhos. Nosso mecanismo de auto-eleição permite que cada nó seja capaz de perceber se é um candidato elegível a próximo salto, tomando como base os valores de RSSI divulgados pelos vizinhos imediatos na etapa de estimativa de distância. O diferencial nesta abordagem é a possibilidade de que um ou mais nós se elejam a próximo salto, permitido a configuração de rotas alternativa em direção ao *sink*, aumentando assim as chances de entrega dos pacotes.

Conforme já discutido em seções anteriores, nossos algoritmos consideram que o nó *sink* está equipado com um dispositivo de comunicação de longo alcance, de tal forma que em uma única consulta seja possível alcançar a todos os nós da rede. Os nós da grade de sensoreamento irão receber a consulta com diferentes valores de RSSI, sendo que nós mais distantes do *sink* vão receber a mensagem com valores referentes à potência do sinal menores, de acordo com a propagação. Os valores do RSSI disseminados pelo *sink* serão utilizados para estimar as distâncias entre os nós. A potência do sinal recebido pode ser facilmente traduzida em distâncias estimadas. Neste sentido, a aferição de distâncias através dos valores do RSSI tem sido amplamente discutida por Ferreira et al. [2014], Holland et al. [2006], Jacinto [2012]. O parâmetro utilizado para o cálculo estimado da distância foi dado em metros e o tempo utilizado na agregação de dados foi dado em segundos. O temporizador considera que cada metro corresponde a um segundo. Desta forma, cada nó irá esperar um

tempo em segundos antes de encaminhar suas informações.

Como forma de facilitar o entendimento sobre as abordagens aqui explanadas, neste trabalho utilizamos o termo "distância estimada" ao invés de "força do sinal recebido"

Algoritmo 2 REACT

VARIÁVEIS:
1: $dadosAgregados_i = \emptyset$; {Dados agregados}
2: $dadosConsulta_i$; {Consulta recebida pelo *sink*}
3: $tempoEnvio_i$; {Envio de dados agregados}
4: $temporizador_i$; {Temporizador}
5: $dist_i$; {Distância em relação ao *sink*}
6: $idBuraco_k = \emptyset$; {Mensagem com identificador do nó preso}
7: $tabelaElegiveis_i$; {Lista de vizinhos elegíveis}

ENTRADA:
8: *Sink*: recebe da central de monitoramento uma requisição $consulta(consultaId_k, consulta_k)$;

AÇÃO:
9: Envia $consulta(consultaId_k, consulta_k)$; a todos os nós da rede.

ENTRADA:
10: $msg_i = consulta(consultaId_k, consulta_k)$;
11: $dist_i = estimaDistancia(msg_i)$;
12: **SE** $dadosConsulta_i = recRetornoConsulta(consulta_k)$ **ENTÃO**
13: $dadosAgregados_i := dadosAgregados_i \cup dadosConsulta_i$;
14: **FIM SE**
15: $tempoEnvio_i = raioMaxComunicSink - dist_i$; {Tempo de agregação}
16: $temporizador_i.inicio(tempoEnvio_i)$;

ENTRADA:
17: $temporizador_i.expirou()$; {Temporizador para a agregação expira}
AÇÃO:
18: Divulga $msg(dadosAgregados_i, dist_i)$;
19: $enviaDado_i = 1$;

ENTRADA:
20: $msg_i = msg(dadosAgregados_k, dist_k)$; {Recebendo dado agregado de vizinho}
21: **SE** $n_i \neq 0$ **ENTÃO** {Sou um nó regular?}
AÇÃO:
22: **SE** $temporizador_i.inativo()$ **ENTÃO** {Tenho tempo para adicionar mais alguma informação?}
23: **SE** $dist_i < dist_k$ **ENTÃO** {Sou vizinho do *sink*?}
24: $dadosAgregados_i = dadosAgregados_i \cup dadosAgregados_k$;
25: **FIM SE**
26: $tabelaElegiveis_i = n_i$;
27: **FIM SE**
28: **FIM SE**
29: **SE** $enviaDado_i = 1$ **ENTÃO** {Meu pacote foi enviado?}
30: **SE** $tabelaElegiveis_i = 0$ **ENTÃO** {Estou preso em uma região de buracos?}
31: $idBuraco_k = 1$;
32: $msg_i = msg_i \cup idBuraco_k$;
33: Divulga $msg(msg_i)$;
34: **FIM SE**
35: **SE** $n_i \neq 0$ **ENTÃO** {Sou um nó regular?}
36: Divulga $msg(dadosAgregados_i, dist_i, idBuraco_k)$;
37: **FIM SE**
38: **FIM SE**
39: **FIM SE** {Eu sou o nó *sink*}
40: Envia pacote de resposta à Central de Monitoramento.
41: **FIM SE**

O funcionamento do algoritmo é apresentado no Algoritmo 2. Seu início ocorre quando o nó *sink* envia uma consulta para toda a rede (linha 9) , em um único salto. Cada nó

que recebe a consulta, estima sua distância até o *sink* através da técnica do RSSI (linhas 10 e 11), como ilustrado na Figura 5.1(a). Logo em seguida, se um nó possui qualquer dado relacionado a esta consulta do nó *sink*, este agrega esse dado à sua lista de dados agregados (linhas 12–14). Um temporizador é iniciado em cada nó, afim de realizar a agregação de dados envolvendo os pacotes recebidos (linhas 15 e 16) e ainda, um tempo de espera para o envio das informações, definido de acordo com os valores de RSSI recebidos do *sink*.

Quando o temporizador expira (linha 17), um pacote de resposta é gerado contendo tanto o dado agregado quando a distância do nó até o *sink*. Em seguida um pacote é enviado a todos os vizinhos através de uma mensagem de *broadcast* (linha 18) e também armazenado em uma lista de pacotes encaminhados, como mostrado na Figura 5.1(b). Os nós próximos ao *sink* irão iniciar o processo de auto-eleição antes dos nós mais distantes. Após consultar sua própria distância em relação ao *sink*, o nó que recebe o pacote elege a si próprio como o salto mais próximo do *sink* (linha 23). O mecanismo de auto-eleição considera que os vizinhos do nó que está enviando a mensagem de anuncio, geograficamente próximos ao *sink*, irão eleger a si como próximo salto. O nó que recebe a mensagem e se enquadra na regra de proximidade irá agregar o dado recebido (linha 24) e aguardar o tempo de agregação para só então enviar o pacote com os dados agregados via *broadcast* aos vizinhos a um salto de distância. O tempo de espera para o envio dos pacotes via mensagem de *broadcast* será proporcional à distância estimada no inicio.

Após o envio do pacote, cada nó checa se está em uma região de buracos e se seu pacote de dados já foi enviado (linha 29). Para esta checagem, o algoritmo utiliza uma lista de nós elegíveis criada quando um nó, geograficamente próximo ao *sink*, para divulgar suas informações (linhas 25–27). Se um nó não possui vizinhos próximos ao *sink* ou ainda se um de seus vizinhos é o próprio *sink*, sua lista de nós elegíveis está limpa. Neste caso, o algoritmo considera que o nó está em uma região de buracos. O nó que está "preso" nesta lacuna de comunicação agrega um identificador ao seu pacote de dados que será utilizado para re-enviar este pacote via mensagem de *broadcast* (linhas 30–34).

Quando um nó que está geograficamente distante do *sink* recebe um pacote enviado pelo vizinho da região de buracos, ele checa o identificador (linha 31) configurado e inicia a etapa de encaminhamento às cegas, que nada mais é do que uma mensagem de *broadcast* enviada por este nó com destino a vizinhos localizados na borda do buraco. A auto-eleição dos nós localizados na borda do buraco considera que, em relação ao *sink*, os nós mais distantes do nó preso não serão considerados. Neste caso, o nós próximos ao *sink* serão capazes de eleger-se (linhas 35–37) e continuar o processo de encaminhamento até encontrar um nó geograficamente próximo ao destino final, confirme ilustrado na Figura 5.1(c).

É importante observar que quando o nó preso inicia o encaminhamento a partir da região de buracos, a maioria dos candidatos elegíveis a próximo salto não estarão geogra-

ficamente próximos ao *sink* mas poderão eleger-se tomando como base as informações dos vizinhos, armazenadas nos pacotes que foram enviados ao nó que está na região de buracos.

5.2 Avaliação de Desempenho

Na presente seção avaliamos o algoritmo REACT e comparamos sua performance à dos algoritmos RSSR Enhanced [Oliveira et al., 2015] e ARESTA [Lima et al., 2016].

5.2.1 Metodologia

A avaliação de performance foi realizada através de simulações usando o simulador Sinalgo ETH-Zurich [2014], que provê um ambiente completo para simulação de algoritmos distribuídos. As simulações foram realizadas em um campo de sensoreamento de 130 m $\times$ 130 m^2. Assim como no algoritmo ARESTA, apresentado na seção 4, os nós foram posicionados em uma grade perturbada, em que a densidade escolhida para os testes foi de 0.03 nós/m^2. No tocante aos resultados apresentados, as curvas representam valores médios, enquanto as barras de erro representam intervalos de confiança para 95 % de confiança a partir de 33 instâncias independentes (sementes aleatórias).

Os parâmetros de simulação foram baseados no sensor *MICAz* e os valores utilizados são mostrados na Tabela 5.1. Os intervalos utilizados para a execução dos experimentos já foram discutidos na Seção 4.2.1. Assim como no algoritmo ARESTA, consideramos que a implantação dos nós obedece a estrutura de uma grade perturbada em que a localização de cada nó sofre um pequeno distúrbio gerado por um erro gaussiano aleatório. O nó *sink* foi posicionado no canto inferior esquerdo enquanto que os nós regulares ocupam a grade uniformemente.

Assim como no algoritmo ARESTA, nossa grade foi configurada com a inserção de um polígono irregular côncavo em forma de *L*, localizado no centro desta grade, de tal forma que os nós de um lado do buraco não consigam se comunicar com os nós do outro lado, portanto a parte oposta do buraco.

Parâmetro	Valor utilizado
Área monitorada	130 m $\times$ 130 m^2
Quantidade de nós	512 nós (grade perturbada)
Posição do Sink	Canto inferior esquerdo
Densidade	0.03 nós/m^2
Alcance da comunicação	30 m
Imprecisão do RSSI a curta distância	15 % da dist. real

Tabela 5.1. Configuração do cenário de simulação do algoritmo REACT.

Para avaliar o desempenho do algoritmo REACT utilizamos quatro métricas, descritas a seguir:

- **Pacotes transmitidos:** número total de pacotes de dados transmitidos pelos nós sensores;
- **Pacotes entregues:** razão entre o número total de pacotes de dados gerados pelos nós sensores e a quantidade de pacotes de dados recebidos pelo nó *sink*;
- **Economia de transmissões:** razão entre o número de transmissões que não foram necessárias devido à agregação de dados e o número de transmissões que seriam efetuadas sem agregar dados;
- **Acertos na escolha do próximo salto:** um acerto ocorre quando um pacote sai de um nó em direção ao seu vizinho, geograficamente mais próximo do nó *sink*, enquanto que um erro se dá quando este pacote foi enviado para um vizinho que não é o mais próximo do *sink*, ou seja, ele poderia ter sido enviado para um vizinho melhor.

5.2.2 Impacto da Escalabilidade da Rede

A escalabilidade é avaliada através do aumento do número de nós na rede de 256 para 1536 nós, aumentando-se o tamanho da área monitorada de modo a manter a densidade dos nós em 0.03 nós/m^2 e o alcance da comunicação em 30 m. Na Figura 5.2(a), podemos observar que o algoritmo RSSR consegue entregar pouco mais de 80% de suas informações quando o número de nós está 256 , uma vez que não é capaz de lidar com buracos de roteamento. Neste sentido, os nós em uma região de buracos possuem poucos vizinhos aptos à auto-eleição e consequentemente, menos opções para a criação de uma árvore de escoamento até o *sink*. Por outro lado, o algoritmo REACT apresenta taxas de entrega acima de 95 % com poucos vizinhos e ainda consegue manter um desempenho ainda superior ao do algoritmo ARESTA, mesmo com o aumento do número de vizinhos. Como afirmamos, os nós presentes na região de buraco possuem poucos vizinhos elegíveis, o que reduz a chance do processo de auto-eleição de um vizinho mais próximo do *sink*. No tocante aos pacotes transmitidos através da rede, observamos que os três algoritmos avaliados apresentam um comportamento esperado quando aumentamos a quantidade de nós na grade. Neste sentido, o número de transmissões aumenta de acordo com o número de saltos percorridos pelos pacotes de dados até o *sink*. Contudo, o número de transmissões experimentadas pelo algoritmo REACT é menor do que nas duas abordagens, já que o algoritmo implementa agregação de dados em cada salto de comunicação. Por fim, apesar do algoritmo ARESTA também implementar agregação de dados, os pacotes localizados na região do buraco trocam um número

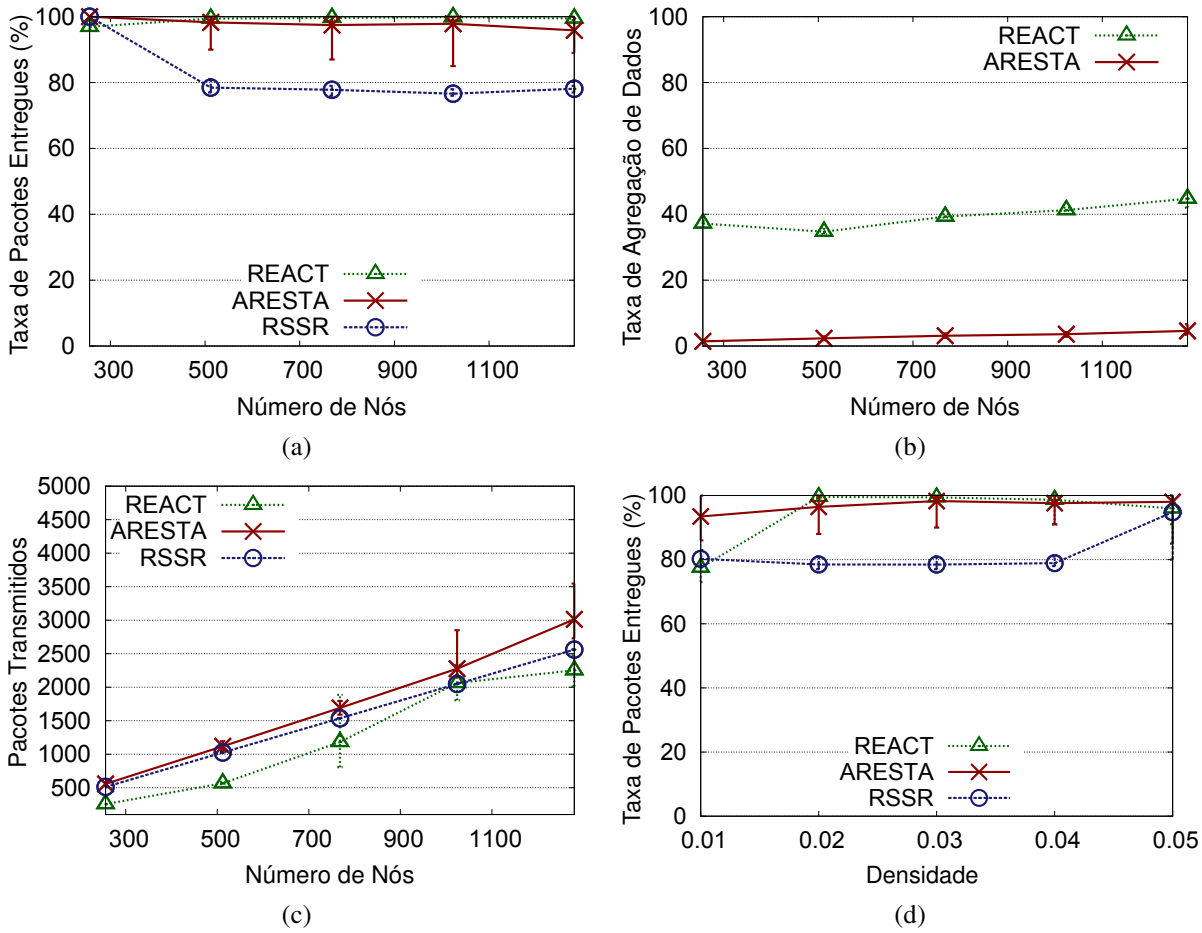


Figura 5.2. Resultados obtidos ao se avaliar a taxa de entrega de pacotes e quantidade de pacotes transmitidos em relação ao alcance da comunicação

maior de informações entre vizinhos. Os resultados confirmam que solução é capaz de operar na presença de buracos, até mesmo quando aumentamos o número de nós.

5.2.3 Impacto do Raio de Comunicação

Para avaliarmos o impacto do raio de comunicação de curto alcance dos nós em nosso algoritmo, aumentamos este parâmetro de 20 m até 40 m. O número de nós vizinhos é reduzido com raios de comunicação de curto alcance, o que consequentemente reduz o número de possíveis candidatos à próximo salto durante o processo de auto-eleição Boukerche et al. [2008]. Observamos na Figura 5.3(c), que o número de pacotes entregues com sucesso pelo algoritmo REACT é de cerca de 95 % quando o raio de comunicação está em 20 m, ficando acima de 95 % quando o alcance da comunicação aumenta para 40 m. Neste caso, conforme citado anteriormente, os nós presentes em uma área de buraco possuem poucos candidatos a próximo salto, o que reduz as oportunidades para a auto-eleição. Ainda assim, o RE-

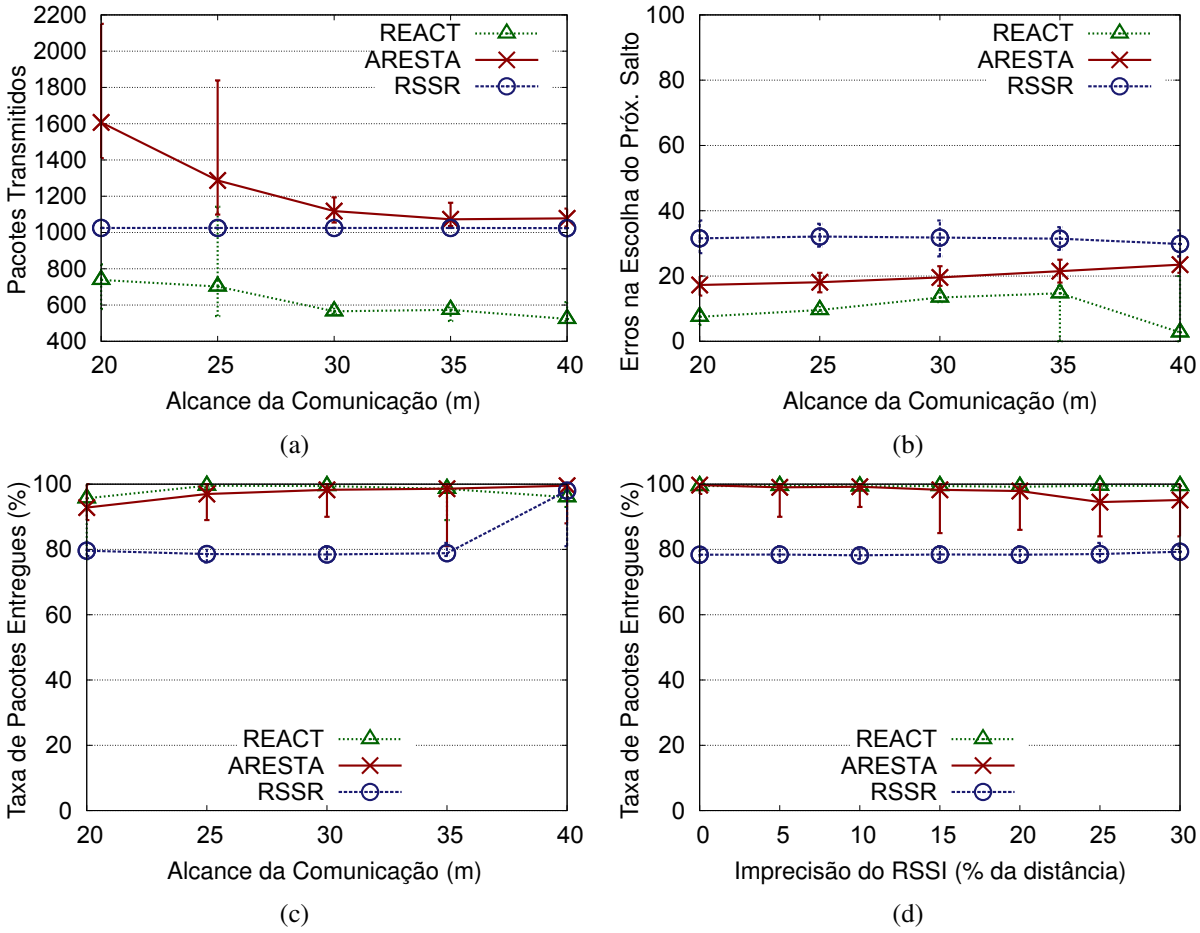


Figura 5.3. Resultados obtidos ao se avaliar as decisões geográficas de próximo salto, a taxa de entrega em relação ao alcance da comunicação e à imprecisão do RSSI, além dos pacotes transmitidos em relação ao alcance da comunicação

ACT opera de forma eficaz na presença de buracos, mantendo taxas de entrega satisfatórias, bem como o algoritmo ARESTA. Por outro lado, é possível observar que o algoritmo RSSR consegue entregar cerca de 80 % de suas informações quando o raio de comunicação varia entre 20 e 35 , em nosso cenário padrão.

No que se refere aos pacotes transmitidos, a Figura 5.3(b) mostra que o REACT supera as duas outras abordagens avaliadas pois fazer as melhores escolhas de próximo salto em direção ao *sink*, além de manter um número baixo de transmissões através da rede, conforme ilustra a Figura 5.3(a).

5.2.4 Impacto da Densidade da Rede

O impacto da densidade da rede em nosso algoritmo foi avaliado começando com uma densidade relativamente baixa de 0.01 nós/m^2 e aumentando tal densidade até 0.05 nós/m^2.

Um nó regular terá poucos vizinhos em densidades baixas, dificultando a agregação e a auto-eleição, enquanto que ao se aumentar a densidade, o número de vizinhos de cada nó aumentará. Como é possível observar na Figura 7.4(a), o algoritmo REACT tem sua taxa de entrega reduzida para 80 % em baixas densidades, chegando a alcançar taxas de sucesso acima 95 % em densidades altas. É fácil observar que o algoritmo ARESTA é capaz de entregar pacotes com altas taxas entrega, mantendo a confiabilidade da solução mesmo em baixas densidades. Contudo, o algoritmo REACT continua funcionando corretamente pois é capaz de operar de forma eficiente mesmo em baixas densidades. Por outro lado, o número de pacotes entregues com sucesso do algoritmo RSSR é de 80 % em baixas densidades, chegando a pouco mais de 95 % quando a densidade está em 0.05 nós/m^2. Essa pequena melhora na taxa de pacotes entregues com sucesso do algoritmo RSSR, foi ocasionada pelo aumento na quantidade de vizinhos. Neste caso, a auto-eleição de múltiplos caminhos aumenta a chance de entrega do pacote.

5.2.5 Impacto da Imprecisão do RSSI a Curta Distância

Nesta seção, avaliamos o impacto da imprecisão do RSSI a curta distância (estimativa de distância entre os nós regulares) aumentando essa imprecisão de 0 % para 30 % da distância real entre os nós. Conforme ilustrado na Figura 7.4(b), as abordagens avaliadas são minimamente afetadas pelo aumento da impressão do RSSI. O efeito desse aumento sobre o processo de estimativa de distância entre os vizinhos influencia as decisões de próximo salto a partir de uma área com a presença de um buraco. Apesar disso, a taxa de sucesso na entrega de pacotes no algoritmo REACT é superior à 99 % em todos os casos.

5.3 Considerações finais

Neste capítulo, apresentamos um novo algoritmo de roteamento geográfico oportunista que chamamos de REACT. O algoritmo é capaz de superar buracos de roteamento através de um novo mecanismo de transmissão às cegas em múltiplos saltos, enquanto provê agregação de dados a partir dos nós que participam do encaminhamento. O algoritmo REACT aproveita os valores de RSSI divulgados por um *sink* de grande capacidade de comunicação de tal forma que a percepção destes valores seja tratada e utilizada como métrica para a construção dos caminhos para o escoamento de pacotes até o *sink*. O algoritmo é capaz de detectar um buraco de roteamento utilizando informações obtidas dos vizinhos de um salto de distância. Por fim, nosso mecanismo cria rotas alternativas para o escoamento de pacotes até o *sink* oriundas da região de buracos através da auto-eleição do próximo salto. De acordo com os resultados apresentamos nesta seção, nosso algoritmo é capaz de atingir os mesmos

benefícios encontrados em abordagens clássicas de roteamento geográfico, tais como altas taxas de entrega e economia de energia enquanto provê agregação de dados e supera buracos de roteamento.

Capítulo 6

Usando *Sinks* de Longo Alcance para contornar Buracos de Roteamento em Redes de Sensores Sem Fio

A técnica de encaminhamento guloso (*greedy forwarding*) tem sido amplamente usada pela maioria dos algoritmos de roteamento geográfico. O mecanismo utiliza o conhecimento prévio sobre os vizinhos de um salto de distância para encaminhar pacotes ao nó geograficamente mais próximo destino. Entretanto, essa estratégia requer que todos os nós sensores conheçam sua localização física, o que nem sempre é possível em cenários de floresta. Além disso, na presença de buracos as abordagens geográficas são obrigadas a abandonar o mecanismo de GF (Greedy Forwarding) para abandonar estratégias de reconhecimento e contorno de buracos através do roteamento por perímetro (borda do buraco), que na maioria dos casos leva ao esgotamento precoce dos nós localizados na borda do buraco .

Neste capítulo, propomos um Algoritmo baseado capaz de superar buracos de roteamento em RSSF que chamamos de LRS (Long-Range SINK). Nosso algoritmo considera a existência de um *sink* de longo alcance para que, em um único salto, seja possível alcançar toda a rede. Nosso algoritmo aproveita o RSSI (Received Signal Strength Indicator) dos nós para permitir a construção do caminho de roteamento em direção ao *sink*. Nossos resultados mostram claramente que o LRS é capaz de proporcionar altas taxas de entrega em cenários com a existência de um buraco de roteamento e ainda mantém os benefícios da técnica de encaminhamento guloso tradicional.

6.1 Algoritmo LRS

Nesta seção apresentamos nossa terceira solução de roteamento geográfico, que chamamos de LRS. Este algoritmo, descrito no Algoritmo 3, também tira proveito da grande capacidade de um *sink* capaz de enviar uma consulta que alcança a todos os nós da rede, em um único salto (linha 9). Através de um aprimoramento implementado para ser executado durante a etapa de estimativa de distâncias entre vizinhos de uma salto de ditância (linhas 10–12), o LRS armazena informações oriundas de vizinhos de um salto do *sink*. É importante observar que, após a consulta única a partir do *sink*, os nós regulares inciam a etapa de estimativa de distâncias entre si e o nó *sink*. Nessa etapa, os nós divulgam suas distâncias em relação ao *sink* e em relação a seus vizinhos de um salto de distância através de uma mensagem de difusão, (*broadcast*). Em seguida, os nós regulares que são vizinhos de um salto de distância do *sink*, ou seja, nós que alcançam o nó *sink* durante a etapa de encaminhamento de pacotes, irão agregar em seu pacotes de anúncio uma informação extra que irá permitir a construção de um caminho de retorno ao *sink*, a partir do nó preso. Desta forma, um nó regular pode escolher com segurança o próximo salto em direção ao *sink*, ainda que esteja preso em uma região de buraco. Assim, o algoritmo LRS alterar o modo de eleição do próximo salto tendo como base as informações sobre os saltos válidos até o *sink*, já armazenadas em etapas anteriores. Se um nó tiver algum dado relacionado à consulta do *sink*, este irá agrega esses dados à lista de dados agregados (linhas 13–15).

O nó preso na região de buraco irá receber pacotes encaminhados por seus vizinhos e em seguida verificar se há uma rota válida para até o *sink*, com base nas mensagens de difusão iniciadas pelos vizinhos do *sink*. Se houver um caminho configurado através do vizinho que transmite a mensagem, o nó armazena essas informações (linhas 16–18). Um temporizador de inicialização é iniciado em cada nó para proporcionar a agregação dos dados recebidos por vizinhos de um salto de distância (linhas 20 e 21). Esse tempo de espera é definido de acordo com os valores de RSSI percebidos durante a consulta inicial do *sink*. Os parâmetros de configuração envolvendo o tempo de espera para o envio de pacotes já foi comentado na seção anterior.

Para limitar a quantidade de nós que participam da escolha da rota alternativa, excluímos todos os vizinhos que recebem valores de RSSI menores que o nó que transmite a mensagem de anúncio aos vizinhos. Quando o temporizador expira (linha 22), um pacote de resposta é montado contendo os dados agregados e a distância do nó ao coletor. Assim, o nó envia o pacote com os dados agregados ao vizinho mais próximo ao *sink* (linha 23). Este pacote é recebido por outro vizinho mais próximo do *sink*), que provavelmente não terá ainda seu temporizador expirado, já que tal nó estaria mais próximo do *sink*. Os dados recebidos são então agregados à lista de dados já agregados no salto anterior (linha 25) para

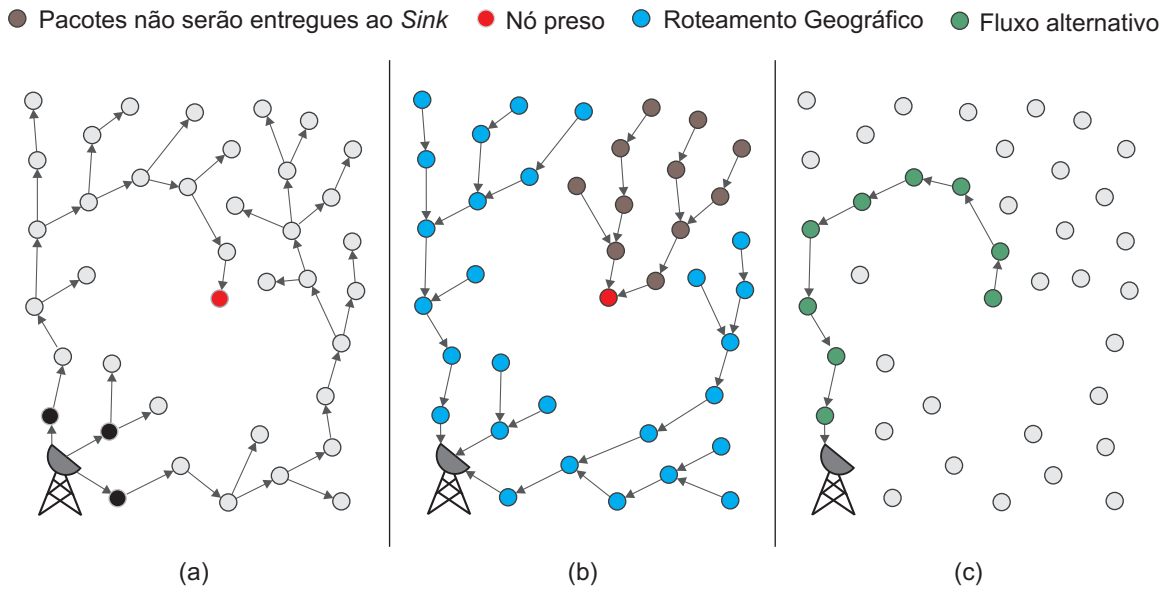


Figura 6.1. (a) Após a consulta única do *sink*, nós próximos ao *sink* (vizinhos de um salto de distância) elegem a si como salto geográfico (*nós em cor preta*). Durante a etapa de estimativa de distância, essa informação de vizinho real do *sink* é divulgada. (b) Pacotes que tendem ao buraco serão encaminhados ao nó preso. (c) Rota alternativa gera a a partir da divulgação de informações dos nós vizinhos ao *sink*.

ser encaminhado após o temporizador expirar. Nós que receberam valores de RSSI maiores (próximos ao *sink*) aguardarão um tempo muito pequeno antes de preparar outro pacote de resposta para enviar. Esse processo é repetido até que o pacote atinja o nó *sink* e, finalmente, serem enviados à central de monitoramento (linha 35).

Um nó regular irá executa o novo esquema de escolha do próximo salto que permite a seleção de um novo vizinho elegível em direção ao sink (linhas 26–34). Nosso mecanismo de encaminhamento permite que o próximo salto seja eleito de acordo com a distância do *sink* em relação a cada vizinho, considerando como elegível o vizinho geograficamente próximo so sink. Cada nó possui uma tabela de vizinhos elegíveis, criada na etapa de estimativa de distâncias entre vizinhos. Se um nós não possível vizinhos elegíveis, ou seja, vizinhos geograficamente próximos ao sink, este nó está preso em um buraco de roteamento (linha 29). Esse nó que está preso vai encaminhar seu dado agregado ao *sink*, mas utilizando a informação de rota segura obtida durante o processo de estimativa de distâncias (linha 30). Essa informação extra utilizada pelo nó do buraco irá considerar elegível o vizinho que conhece a rota alternativa até o *sink*, conforme ilustrado na Figura 6.1.

Algoritmo 3 LRS

VARIÁVEIS:
1: $dadosAgregados_i = \emptyset$; {Dados Agregados}
2: $dadosConsulta_i$; {Consulta recebida pelo *sink*}
3: $tempoEnvio_i$; {Tempo para o envio de dados gregados}
4: $temporizador_i$; {Temporizador}
5: $dist_i$; {Distância em relação ao *sink*}
6: $rotaSink_i = false$; {Salto válido em direção ao *sink*}
7: $tabElegiveis_i$; {Lista de vizinhos elegíveis}

ENTRADA:
8: *Sink*: recebe da central de monitoramento uma requisição do tipo $consulta(idConsultaSink_k, consultaSink_k)$;

AÇÃO:
9: *Sink* envia uma $consulta(idConsultaSink_k, consultaSink_k)$; a todos os nós da rede.

ENTRADA:
10: $msg_i = consulta(idConsultaSink_k, consultaSink_k)$;
11: $dist_i = estimaDist(msg_i)$;

AÇÃO:
12: *Nó* Envia $adv(n_i, dist_i)\ \forall\ n_j \in neig_i$;

AÇÃO:
13: **SE** $dadosConsulta_i == getRetornoConsulta(consultaSink_k)$ **ENTÃO**
14: $dadosAgregados_i := dadosAgregados_i \cup dadosConsulta_i$;
15: $msg_i := msg_i \cup rotaSink_i$;
16: **SE** $dist_i <= alcanceNo_i$ **ENTÃO**
17: $rotaSink_i = true$;
18: **FIM SE**
19: **FIM SE**

AÇÃO:
20: $tempoEnvio_i = alcanceSink - dist_i$; {Tempo de agregação}
21: $temporizador_i.inicio(tempoEnvio_i)$;

AÇÃO:
ENTRADA:
22: $temporizador_i.expira()$; {Temporizador para a agregação expira}

AÇÃO:
23: Envia $consulta(dadosAgregados_i)$ para $proxSalto_i$;
24: $dadoEnviado_i = 1$;

ENTRADA:
25: $msg_i = consulta(dadosAgregados_k)$;

AÇÃO:
26: **SE** $n_i \neq 0$ **ENTÃO** {Sou um nó regular?}
27: $dadosAgregados_i := dadosAgregados_i \cup dadosAgregados_k$;
28: **SE** $dadoEnviado == 1$ **ENTÃO** {Meu dado foi enviado?}
29: **SE** $tabElegiveis_i == 0$) **ENTÃO** {Sou um nó preso?}
30: Envia $consulta(dadosAgregados_i)$ para $rotaSink_i$;
31: **FIM SE**
32: $dadosAgregados_i = \emptyset$;
33: **FIM SE**
34: **FIM SE** {Sou o nó *sink*}
35: Envia pacote de resposta à Central de Monitoramento.
36: **FIM SE**

6.2 Avaliação de Desempenho

Nesta seção, avaliamos nosso algoritmo LRS proposto e comparamos seu desempenho com nosso trabalho anterior chamado REACT, disponível em Lima et al. [2017] e com o algoritmo de roteamento geográfico e desvio de buracos chamado RSSR, apresentado e discutido em Oliveira et al. [2015]. Nos algoritmos LRS e REACT, a agregação de dados é

provida durante do processo de encaminhamento de pacotes ao *sink*.

6.2.1 Metodologia

A avaliação de performance do algoritmo LRS também foi realizada através de simulações executadas no simulador Sinalgo ETH-Zurich [2014], que provê um ambiente completo para simulação de algoritmos distribuídos. As simulações foram realizadas em um campo de sensoreamento de $200\,\text{m} \times 200\,\text{m}^2$. Assim como no algoritmo ARESTA, apresentado na seção 4, os nós foram posicionados em uma grade perturbada, em que a densidade escolhida para os testes foi de $0.03\,\text{nós/m}^2$. No tocante aos resultados apresentados, as curvas representam valores médios, enquanto as barras de erro representam intervalos de confiança para 95 % de confiança a partir de 33 instâncias independentes (sementes aleatórias).

Os parâmetros de simulação foram baseados no sensor *MICAz* e os valores utilizados são mostrados na Tabela 6.1. Quanto à topologia da rede, nós assumimos que a implantação dos nós obedece a estrutura de uma grade perturbada em que a localização de cada nó sofre um pequeno distúrbio gerado por um erro gaussiano aleatório. O nó *sink* foi posicionado no canto inferior esquerdo enquanto que os nós regulares ocupam a grade uniformemente. Com relação à tabela que expressa o cenário de execução dos experimentos envolvendo o algoritmo LRS, a maneira de como os intervalos utilizados interagem com as simulações já foi discutidos na Seção 4.2.1.

Assim como nos algoritmos ARESTA e REACT, inserimos na grade de sensoreamento um polígono irregular côncavo em forma de *L*, localizado no centro da grade, de tal forma que os nós de um lado do buraco não consigam se comunicar com os nós do outro lado, portanto a parte oposta do buraco. Durante as simulações, cada nó sensor gera um pacote de dados com eventos detectados.

As seguintes métricas foram utilizadas para avaliar o algoritmo LRS:

- **Pacotes transmitidos:** número total de pacotes de dados transmitidos pelos nós sensores;

Parâmetro	Valor utilizado
Área monitorada	$200\,\text{m} \times 200\,\text{m}^2$
Quantidade de nós	512 nós (grade perturbada)
Posição do Sink	Canto inferior esquerdo
Densidade	$0.03\,\text{nós/m}^2$
Alcance da comunicação	30 m
Imprecisão do RSSI a curta distância	15 % da dist. real
Imprecisão do RSSI a longa distância	1 % da dist. real

Tabela 6.1. Configuração do cenário de simulação do algoritmo LRS.

- **Pacotes entregues:** razão entre o número total de pacotes de dados gerados pelos nós sensores e a quantidade de pacotes de dados recebidos pelo nó *sink*;

- **Acertos na escolha do próximo salto:** um acerto ocorre quando um pacote sai de um nó em direção ao seu vizinho, geograficamente mais próximo do nó *sink*, enquanto que um erro se dá quando este pacote foi enviado para um vizinho que não é o mais próximo do *sink*, ou seja, ele poderia ter sido enviado para um vizinho melhor.

- **Agregação de dados:** razão entre o número total de pacotes de dados enviados pelos nós sensores e a quantidade de pacotes agregados em nós intermediários durante o processo de encaminhamento.

6.2.2 Impacto da Densidade da Rede

O impacto da densidade da rede foi avaliado aumentado os valores da densidade de 0.03 para 0.09 nodes/m^2. Como é possível observar pela Figura 6.2(a), os algoritmos LRS e REACT praticamente não são afetados pelo aumento na densidade da rede, sendo capazes de entregar mais de 95 % de suas informações ao *sink*. Por outro lado, é possível constatar que o RSSR não foi capaz de entregar mais do que 80 % em todos os casos observados. Em baixas densidades e mantendo-se o mesmo alcance de comunicação, as oportunidades para a agregação de dados são reduzidas. neste sentido, comparamos os algoritmos LRS e REACT, já que o algoritmo RSSR não implementa agregação de dados. Assim, observamos pela Figura 6.4(a), que a porcentagem de pacotes agregados pelo algoritmo REACT é de cerca de 35 % enquanto no LRS essa mesma taxa fica em torno de 15 %. A esse respeito, é importante salientar que o algoritmo LRS escolhe somente um vizinho por vez durante a etapa de seleção do próximo salto. Já o algoritmo REACT, implementa a técnica de transmissão as cegas com auto-eleição do próximo salto. Neste sentido, mais de um vizinho será eleito para o próximo salto em direção ao *sink*.

6.2.3 Impacto da Imprecisão do RSSI a Curta Distância

Avaliamos o impacto da Imprecisão do RSSI percebida pelos vizinhos de um salto de distância. Esta análise foi realizada variando-se essa da imprecisão de 5 para 30 % das distâncias reais entre os nós. Como ilustrado pela Figura 7.5(a), tanto o REACT quanto o RSSR não são afetados pela imprecisão, especialmente o primeiro algoritmo que realiza um número menor de transmissões por conta de sua característica singular de envio de pacotes através de uma única mensagem de anúncio. Por outro lado, o algoritmo LRS é afetado pela imprecisão do RSSI durante o processo inicial de estimativa de distâncias entre os vizinhos

e o *sink*. Observamos que o efeito de se aumentar a imprecisão do RSSI afeta diretamente a escolha do próximo salto, principalmente para os nós de região do buraco. É importante frisar que as decisões de próximo salto do nó preso não serão realizadas a partir do mecanismo tradicional de encaminhamento *greedy forwardind*, conforme já ilustrado na Figura 6.1 e reforçado no Algoritmo 3. Apesar disso, ainda observamos que, de acordo com a Figura 6.2(b) a taxa de sucesso na entrega de pacotes do algoritmo LRS é um pouco superior à do algoritmo REACT. Isso se deve ao mecanismo de eleição do próximo salto do algoritmo LRS, já que nesta abordagem sempre há uma rota segura até o *sink*, previamente configurada.

6.2.4 Impacto da Escalabilidade da Rede

A escalabilidade da rede foi avaliada aumentando-se o tamanho da rede de 256 para 1280 , mantendo a densidade constante de 0,03 nós/m^2 e o alcance de comunicação em 30 m. Dessa forma, a grade é redimensionada de acordo com o número de

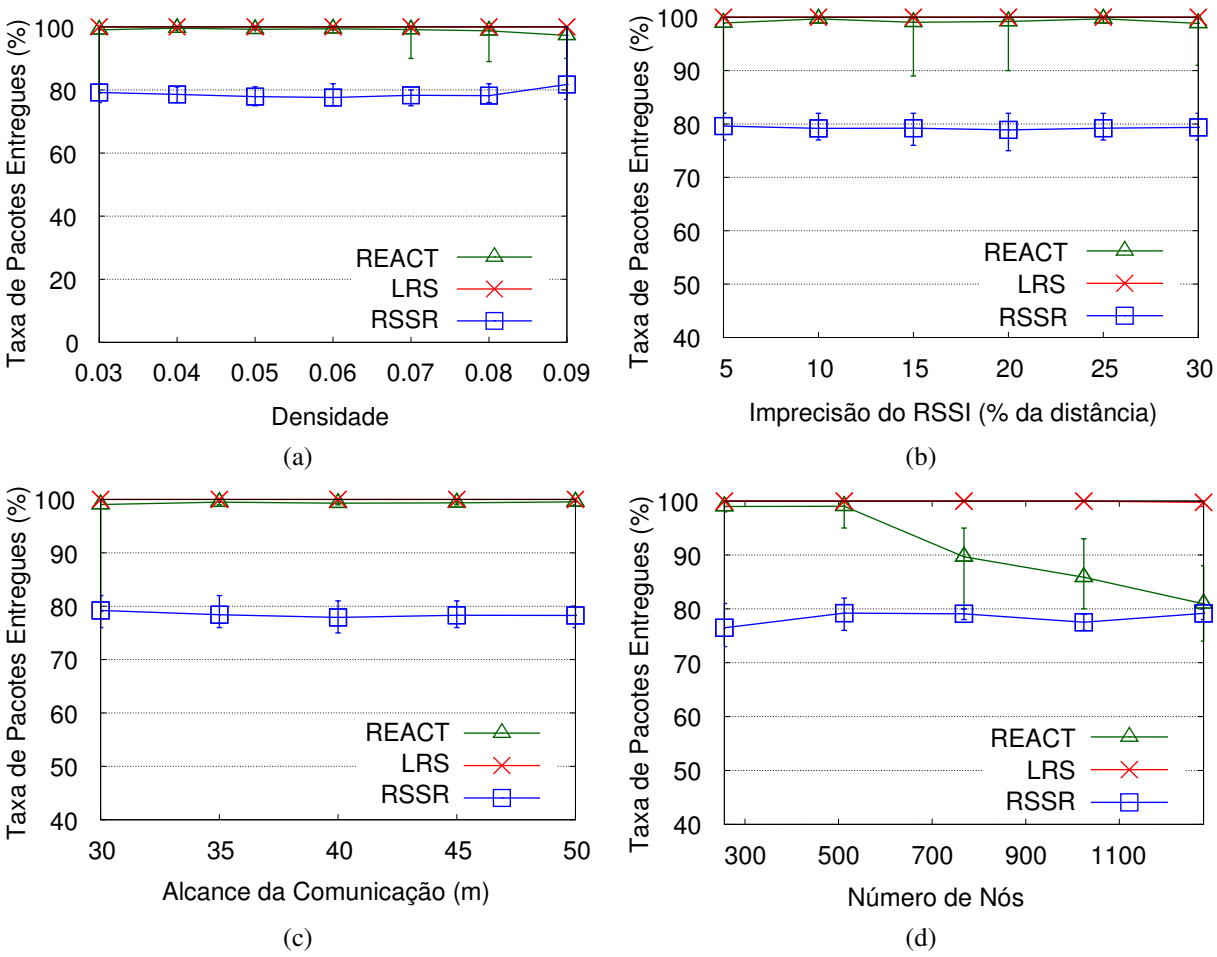


Figura 6.2. Resultados obtidos ao se avaliar a taxa de entrega.

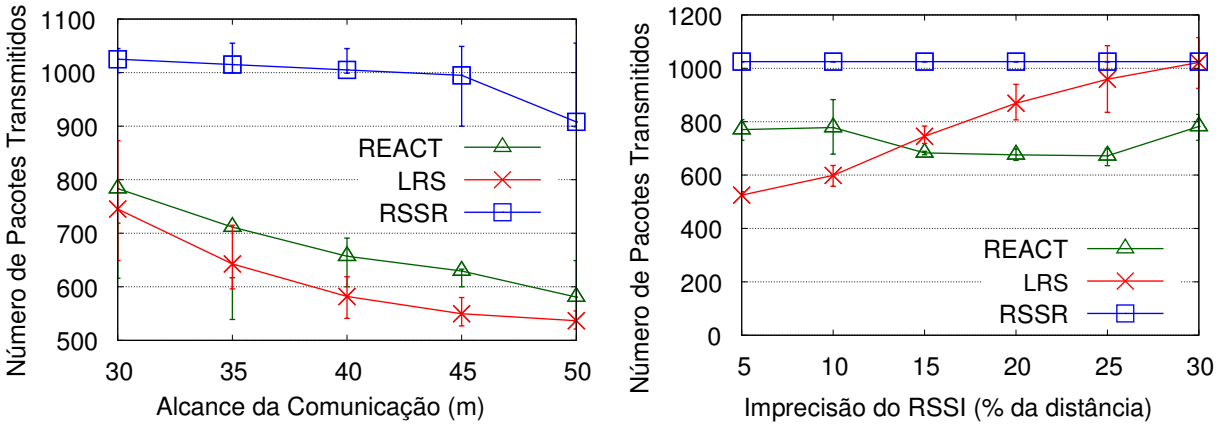


Figura 6.3. Resultados obtidos ao se avaliar a quantidade de pacotes transmitidos.

sensores. A Figura 6.2(d), mostra que tanto o LRS quanto o REACT podem entregar cerca de 98 % de suas informações quando o número de nós na grade está entre 256 e 512 . Já o RSSR conseguiu entregar mais do que 80 % de suas informações em todos os casos observados. Foi possível observar ainda que o algoritmo LRS é capaz de operar na presença de buracos de roteamento, mesmo quando o número de nós está acima de 512 , ao contrário do RSSR. Em relação ao decréscimo na taxa de entrega do algoritmo REACT, salientamos que nesta abordagem o algoritmo depende da densidade da rede durante a etapa de auto-eleição do próximo salto. Neste sentido, tanto o RSSR quanto o REACT descartarão alguns vizinhos do nó preso na etapa de detecção do buraco. Em todos os casos, a taxa de entrega de pacotes no algoritmo LRS esteve sempre acima de 98 %.

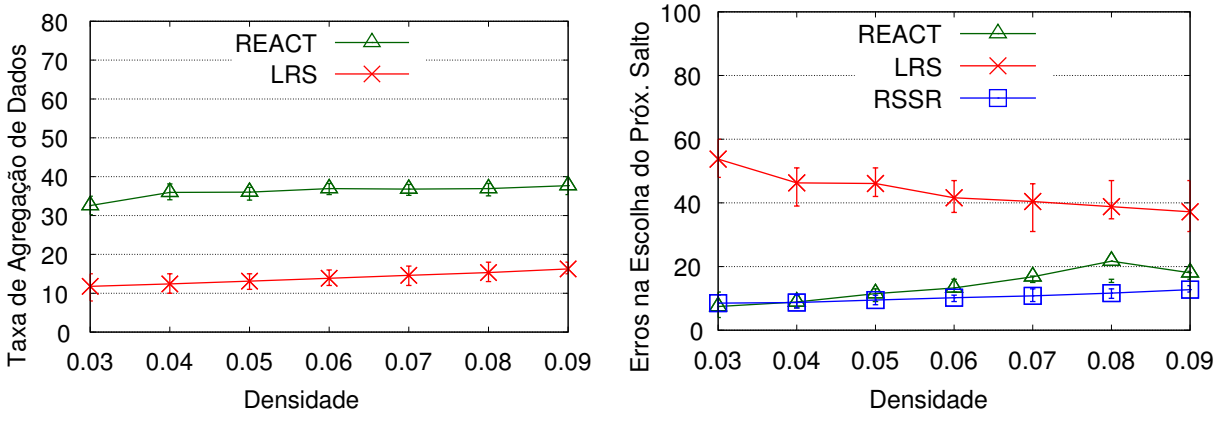


Figura 6.4. Resultados obtidos ao se avaliar a taxa de agregação de dados, os erros na escolha de próximo salto e a quantidade de transmissões na rede.

6.2.5 Impacto do Raio de Comunicação dos Nós

A última condição avaliada foi o impacto do alcance da comunicação dos nós. Para tanto, variamos esse parâmetro de 30 m para 50 m. Quando o alcance de comunicação é curto, um nó regular terá menos vizinhos. Neste sentido, menos vizinhos promovem uma redução no número de escolhas elegíveis a próximo salto de comunicação. Como observado na Figura 6.2(c), a taxa de sucesso na entrega de pacotes pelos algoritmos LRS e REACT é de cerca de 98 % em todos os casos. Entretanto, essa mesma taxa observada no algoritmo RSSR foi de cerca de 80 %. A Figura 6.3(a) ilustra o número de transmissões nos algoritmos avaliados. Na Figura é possível observar que o número de transmissões do algoritmo RSSR é consideravelmente maior que nos algoritmos LRS e REACT. Ainda assim, é possível constatar que mesmo quando o raio de comunicação está em 30 m, a quantidade de pacotes transmitidos ainda é menor no algoritmo LRS. Isso se deve à técnica de escolha do salto elegível pelo nó preso reforçada pela agregação de dados em cada salto de comunicação. Nestes termos, apesar do alto número de erros experimentado pelo algoritmo LRS, ilustrado na Figura 6.4(b) e do desempenho inferior no tange à agregação de dados (Figura 6.4(a)), o algoritmo LRS é capaz de entregar praticamente todas as suas informações ao *sink* mantendo um baixo número de mensagens transmitidas à rede.

6.3 Considerações finais

Neste capítulo, propusemos e analisamos um novo algoritmo de roteamento geográfico, chamado LRS. A ideia chave desta abordagem é considerar que o *sink* está equipado com um dispositivo de comunicação de longo alcance de comunicação, de tal forma que em um único salto de comunicação todos os nós dispostos na grade de sensoreamento são alcançados. Nós mais distantes do *sink* receberão valores de RSSI menores dos os nós mais próximos. Neste sentido, o RSSI percebido pelos nós irá habilitar a construção de fluxos de escoamento de pacotes seguros em relação ao sink. Um nó preso em um buraco de roteamento será capaz de escolher o caminho seguro até o nó, abandonando o roteamento geográfico tradicional e adotando temporariamente nosso mecanismo de roteamento alternativo. o LRS se mostrou eficaz na detecção e contorno de buracos, mantendo altas taxas de entrega de pacotes enquanto provê agregação de dados durante o processo de encaminhamento.

Capítulo 7

Evitando Buracos de Roteamento em Redes de Sensores Sem Fio com *Sinks* de Alto Alcance

O fenômeno dos mínimos locais (local minimum), comumente atribuído a obstáculos físicos, esgotamento de energia dos nós, falhas na comunicação entre vizinhos ou até mesmo a implantação incorreta dos nós na grade de sensoreamento, tem sido tratado por grande parte das abordagens geográficas como um problema de cobertura de rede. Tal fenômeno, contribui para a ocorrência de um buraco de roteamento e é a causa do insucesso das abordagens geográficas no que se refere à transmissão de mensagens através da rede. Para lidar com um buraco de roteamento, se faz necessária a utilização de mecanismos que exploram a borda do buraco e consequentemente a coleta de dados de todos os nós implantados (alocados) ao longo da borda. No entanto, esse mecanismo não satisfaz o requisito de balanceamento de carga, pois impõe muito tráfego aos nós da borda tornando-o o mais utilizados que outros nós da grade. Assim sendo, neste capítulo propomos uma nova técnica de roteamento geográfico que evita buracos de roteamento em RSSF, chamada BYPATH. A estratégia desta solução é evitar regiões com a ocorrência de buracos e criar caminhos para o escoamento de pacotes que circundem o buraco enquanto repassam informações em múltiplos saltos. Consideramos que o *sink* utilizado em nossa solução também possui um alto alcance de comunicação que permitem que todos os nós sejam alcançados em um único salto de comunicação. Nossos resultados mostram a eficiência alcançada pelo nosso mecanismo em cenários com regiões de buraco, pois evita que pacotes sejam encaminhados à região de buracos a ainda mantém altas taxas de entrega de pacotes.

7.1 Algoritmo BYPATH

Nesta seção, apresentamos uma nova abordagem de roteamento geográfico livre de sistemas de coordenadas e capaz de evitar buracos de roteamento em RSSF, através de nossa técnica de escolha de caminhos elegíveis que considera os valores de RSSI do *sink* e dos nós regulares para criar caminhos alternativos para o fluxo de pacotes de dados. O algoritmo BYPATH é capaz de criar rotas em torno de uma região de furos sem a necessidade de cálculos dispendiosos a partir do nó preso. Os nós em uma região de buraco não serão ativados, uma vez que os valores de RSSI propagados no estágio inicial da operação do algoritmo serão a base de conhecimento para a criação dos fluxos para o escoamento de pacotes ao redor do buraco. A solução foi projetada para ser tão confiável quanto um algoritmo geográfico clássico, incluindo altas taxas de entrega e economia de energia durante o processo de encaminhamento de pacotes, sem exigir qualquer atribuição de coordenadas virtuais ou quaisquer outros sistemas de coordenadas.

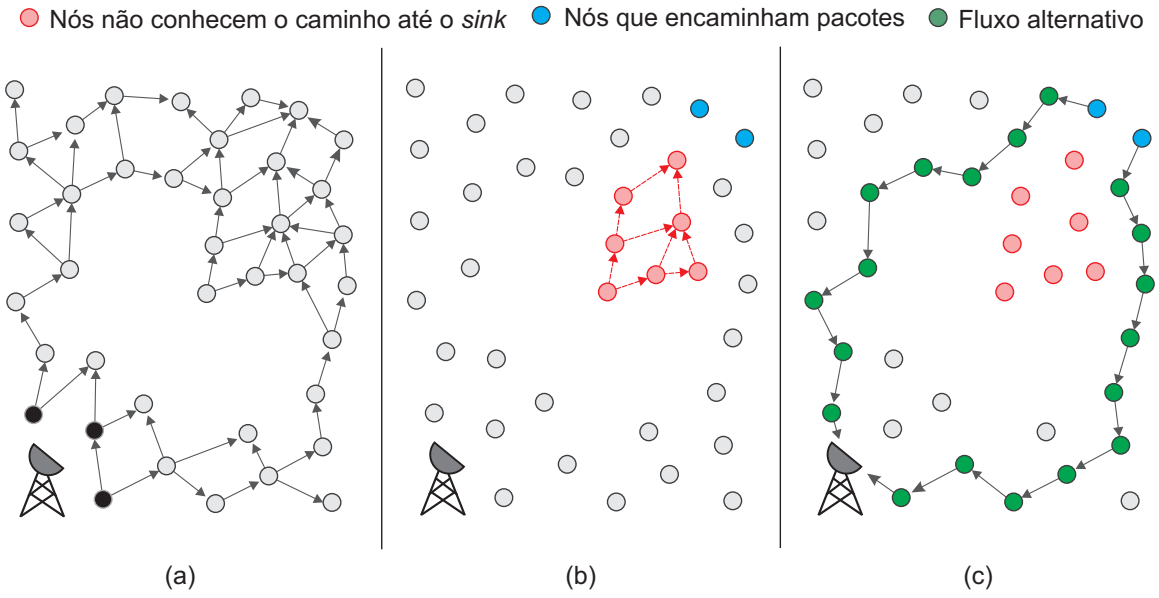


Figura 7.1. (a) Nós próximos ao *sink* se auto-elegem como salto geográfico (*nós em cor preta*) no momento em que recebem a consulta do *sink*. Durante a etapa de estimativa de distância, essa informação de vizinho real do *sink* é divulgada. (b) Nó da região de buracos também divulga sua distância estimada aos vizinhos, mas sem a informação de rota até o *sink*. (c) Nós irão escolher o vizinho mais próximo ao destino, optando pela rota gerada a partir dos vizinhos do *sink*.

Durante a execução do BYPATH, os valores RSSI disseminados pelo sink, são usados para gerar fluxos de retorno dos pacotes de dados e estimar a distância entre um transmissor e receptor, conforme ilustrado nas Figura 7.1(a) e 7.1(b). Neste sentido, no mecanismo circunjacente do BYPATH cada nó seleciona um vizinho fora da região do buracos, geogra-

ficamente mais próximo ao *sink*, conforme ilustrado na Figura 7.1(c). Essa característica só é possível porque, após a propagação da primeira consulta do *sink*, os nós regulares estimam suas distâncias para os vizinhos e para o *sink*. Assim, a escolha do vizinho capaz de encaminhar o pacote é sempre baseada aas informações obtidas nesta etapa de disseminação. Entretanto, um campo adicional foi adicionado ao pacote divulgado durante a etapa de disseminação das distâncias estimadas. Tais informações serão usadas para permitir que os vizinhos reais do *sink*, ou seja, nós localizados a um salto de distância do *sink*, sejam utilizados como referência para criar caminhos seguros de fluxo dos pacotes. Esta informação adicional será usada para encontrar uma rota válida até o *sink*.

A implementação completa do BYPATH é mostrada no Algoritmo 4. O algoritmo inicia quando o *sink* propaga uma consulta para a rede inteira (linha 8) em um único salto, como ilustrado na Figura 7.1(a). Ao receber a consulta, cada nó estima sua distância em relação ao *sink* usando os valores de RSSI percebidos nesta etapa. Em seguida, o nó verifica se é um vizinho real *sink*,ou seja, se está a um salto de distância do *sink* (linhas 9–13), ilustrado pela Figura 7.1(b). Em caso afirmativo, esse nó adiciona uma informação extra em seu pacote de anúncio que será usada durante a escolha do próximo salto de encaminhamento (linha 10). Em seguida, o nó envia um pacote de anúncio aos seus vizinhos, contendo sua distância estimada para o *sink* e as informações de rota que serão utilizadas para criar a árvore de fluxos em direção ao *sink* (linha 14). Um temporizador de inicialização é executado e iniciado nos nós que participarão no envio de dados (linhas 15–16). O tempo de espera é definido de acordo com os valores de RSSI recebidos por cada nó. Nesse sentido, o parâmetro para o cálculo da distância foi dado em metros e o tempo de espera utilizado para o envio de mensagens dado em segundos. O temporizador considera que cada metro corresponde a um segundo e, portanto, cada nó aguardará um tempo em segundos antes de encaminhar seu pacote. Os nós que receberam um RSSI maior (mais próximo do *sink*) aguardarão um tempo muito pequeno antes de preparar seu pacote de retorno para o envio.

Quando o temporizador de encaminhamento expira, um pacote contendo as informações do salto seguro e a distância em relação ao *sink* é preparado (linhas 20–25). A informação de salto seguro (linhas 9–11 e 17–19), é utilizada na criação dos fluxos de escoamento de pacotes que evitam a região do buraco. A abordagem geográfica utilizada neste trabalho considera a árvore de caminhos que foi construída durante o processo de estimativa de distâncias, a partir da mensagem única do *sink*. Nosso esquema de roteamento foi avaliado e comparado ao mecanismo de encaminhamento guloso clássico.

No processo de envio de pacotes, o nó envia o pacote já preparado para o vizinho que está mais próximo ao *sink* (linha 23), mas obedecendo aos critérios de salto seguro e não à abordagem de encaminhamento clássica, ilustrado pela Figura 7.1(c). O processo se repete até que o pacote oriundo do nó escolhido para o envio das mensagens alcance o

sink (linhas 27–31). Por fim, o pacote recebido é encaminhado à Central de Monitoramento.

Diferentemente das soluções atuais, no BYPATH os nós dispostos na região do buraco não precisam criar tabelas de roteamento ou fazer cálculos dispendiosos para detectar a área do buraco, pois os fluxos gerados a partir da troca de mensagens entre os vizinhos irão naturalmente circundar o buraco. Não é assim no algoritmo GPSR, pois os nós dispostos na borda do buraco são usados em excesso. Esse comportamento repetitivo pode levar ao esgotamento de energia de tais nós e falhas decorrentes. Nossa solução de roteamento geográfico livre de sistemas de coordenadas, provou ser eficaz em evitar regiões de buraco e fornecer rotas geográficas de roteamento compatíveis ao mecanismo de encaminhamento de guloso clássico.

7.2 Avaliação de Desempenho

Nesta seção, avaliamos e comparamos o desempenho do algoritmo BYPATH à nossa proposta anterior chamada LRS, discutida nesta tese e disponível em [Lima et al., 2018] e à abordagem clássica de detecção e desvio de buracos de roteamento, chamada GPSR (Greedy Perimeter Stateless Routing) [Karp & Kung, 2000]. Este último algoritmo utilizada a técnica de reconhecimento de perímetro para encaminhar pacotes ao longo da região de buracos. Neste sentido, pretendemos validar nosso esquema de encaminhamento que supera regiões de buraco e é livre de sistemas de coordenadas virtuais ou quaisquer outros sistemas de coordenadas.

Durante a operação do algoritmo BYPATH, cada nó (previamente escolhido) monta um único pacote de dados, contendo sua distância estimada até o *sink*. Isso é feito logo após a propagação da consulta única do *sink*. Quando o temporizador do nó expira, este envia seu pacote de dados apenas uma vez e, após isso, os pacotes gerados e enviados pelos vizinhos seguem o esquema geográfico proposto, livre de sistemas coordenadas. Para avaliar a criação de fluxos que evita o buraco de roteamento, um conjunto de nós localizados no canto superior direito da grade foi selecionado em cada ciclo de simulação. O *sink* foi implantado no lado oposto, ou seja, no canto inferior esquerdo da grade. A ideia principal era observar e analisar o fluxo de pacotes produzido a partir dos nós regulares escolhidos, enquanto as mensagens eram transmitidas através da rede em múltiplos saltos. Desta forma, foi possível comprovar a viabilidade de nossa abordagem em relação aos fluxos alternativos que evitam a região de buracos.

Visando comparar os fluxos gerados pelas três abordagens, as Figuras 7.2 e 7.2 mostram os gráficos de fluxo gerados durante as simulações. Nas figuras, é possível ver claramente que os caminhos de roteamento gerados pelo BYPATH evitam o buraco enquanto

Algoritmo 4 BYPATH

VARIÁVEIS:
1: $dadosConsulta_i$; {Conjunto de consultas recebidas}
2: $tempoEnvio_i$; {Tempo para o inicio do encaminhamento}
3: $temporizador_i$; {temporizador}
4: $dist_i$; {Distância em relação ao *sink*}
5: $rotaSegura_i = false$; {Salto válido em direção ao *sink*}

ENTRADA:
6: *Sink*: recebe da central de monitoramento uma requisição do tipo $consulta(idConsultaSink_k, consultaSink_k, dest_k)$;

AÇÃO:
7: $src_i = n_i$;
8: *Sink* envia uma $consulta(src_i, idConsultaSink_k, consultaSink_k, dest_k)$; to todos os nós da rede.

AÇÃO:
9: **SE** $dist_i <= alcanceNos_i$ **ENTÃO**
10: $rotaSegura_i = true$;
11: **FIM SE**

ENTRADA:
12: $msg_i = consulta(idConsultaSink_k, consultaSink_k, dest_k, rotaSegura_i)$;
13: $dist_i = estimaDist(msg_i)$;

AÇÃO:
14: Envia $adv(dist_i, rotaSegura_i)\ \forall\ n_j \in vizinhos_i$;

ENTRADA:
15: $tempoEnvio_i = (alcanceSink - dist_i)$; {Configura tempo para o encaminhamento}

AÇÃO:
16: $temporizador_i.start(tempoEnvio_i)$; {Temporizador inicia}
17: **SE** $(rotaSegura_i == true)$ **ENTÃO**
18: $proxSalto_i = noOrigem_i$;
19: **FIM SE**
20: **SE** $(this.temporizador_i == temporizador_i.expira())$ **ENTÃO**
21: $dest_i = src_k$;
22: $dataPckt_i = consulta(dest_i, rotaSegura_i)$;
23: Envia $dataPckt_i$ para $proxSalto_i$;
24: $dadoEnviado_i = 1$;
25: **FIM SE**

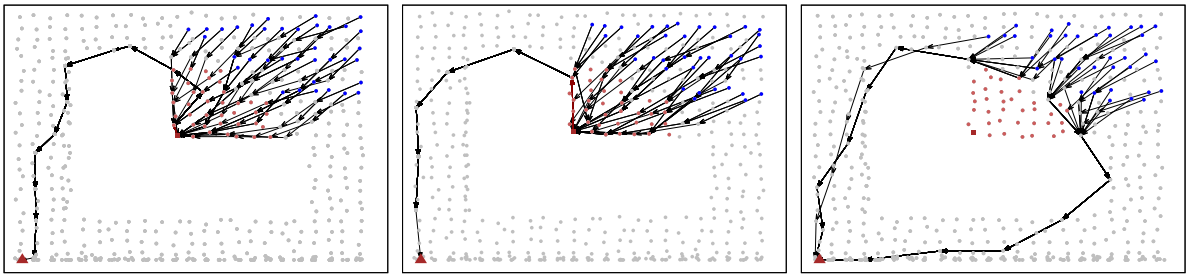
ENTRADA:
26: $msg_i = reply(dataPckt_i)$;

AÇÃO:
27: **SE** $n_i \neq 0$ **ENTÃO** {Sou um nó regular?}
28: Envia $dataPckt_i$ para $proxSalto_i$;
29: **FIM SE** {Eu sou o nó *sink*}
30: Envia pacote de resposta à Central de Monitoramento.
31: **FIM SE**

encaminham os pacotes em direção ao coletor.

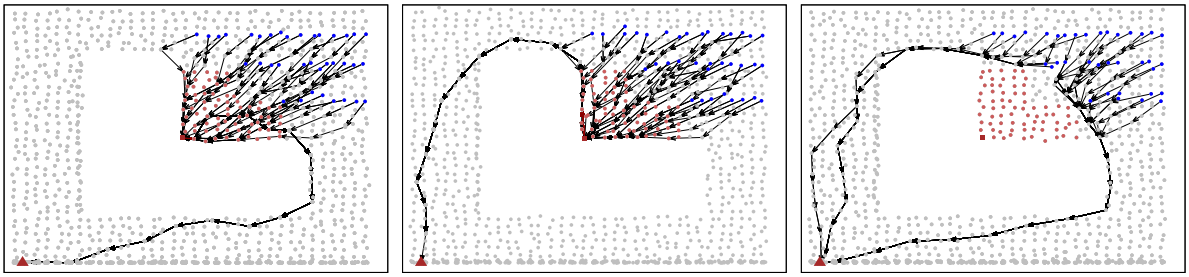
7.2.1 Metodologia

O desempenho do algoritmo BYPATH foi avaliado através de simulações executadas no simulador Sinalgo ETH-Zurich [2014]. A grade de sensoreamento básica para os experimentos gerais é de $242\,\text{m} \times 242\,\text{m}^2$. Os nós foram posicionados em uma grade perturbada, mantendo a densidade da rede em $0.03\,\text{nós/m}^2$. Em nossos resultados, as curvas representam valores médios, enquanto as barras de erro representam intervalos de confiança para 95 % de



(a) LRS (b) GPSR (c) BYPATH

Figura 7.2. Fluxo de Pacotes: Grade com 512 nós.



(a) LRS (b) GPSR (c) BYPATH

Figura 7.3. Fluxo de Pacotes: Grade com 1152 nós.

confiança a partir de 33 instâncias independentes (sementes aleatórias).

A Tabela 7.1 mostra o conjunto de parâmetros de simulação utilizados na simulação. Quanto à topologia da rede, consideramos que a disposição dos nós foi feita a partir de uma grade perturbada em que a localização de cada nó sofre um pequeno distúrbio gerado por um erro gaussiano aleatório. O nó *sink* foi posicionado no canto inferior esquerdo enquanto que os nós regulares ocupam a grade uniformemente. Os intervalos utilizados para a execução dos experimentos foram discutidos na Seção 4.2.1.

Parâmetro	Valor utilizado
Área monitorada	$200\,\text{m} \times 200\,\text{m}^2$
Quantidade de nós	512 nós (grade perturbada)
Posição do Sink	Canto inferior esquerdo
Densidade	$0.03\,\text{nós/m}^2$
Alcance da comunicação	30 m
Imprecisão do RSSI a curta distância	10 % da dist. real
Imprecisão do RSSI a longa distância	1 % da dist. real

Tabela 7.1. Configuração do cenário de simulação do algoritmo BYPATH.

Um polígono irregular côncavo em forma de *L*, localizado no centro da grade de sensoreamento foi inserido para simular um buraco de roteamento, conforme ilustrado pelas

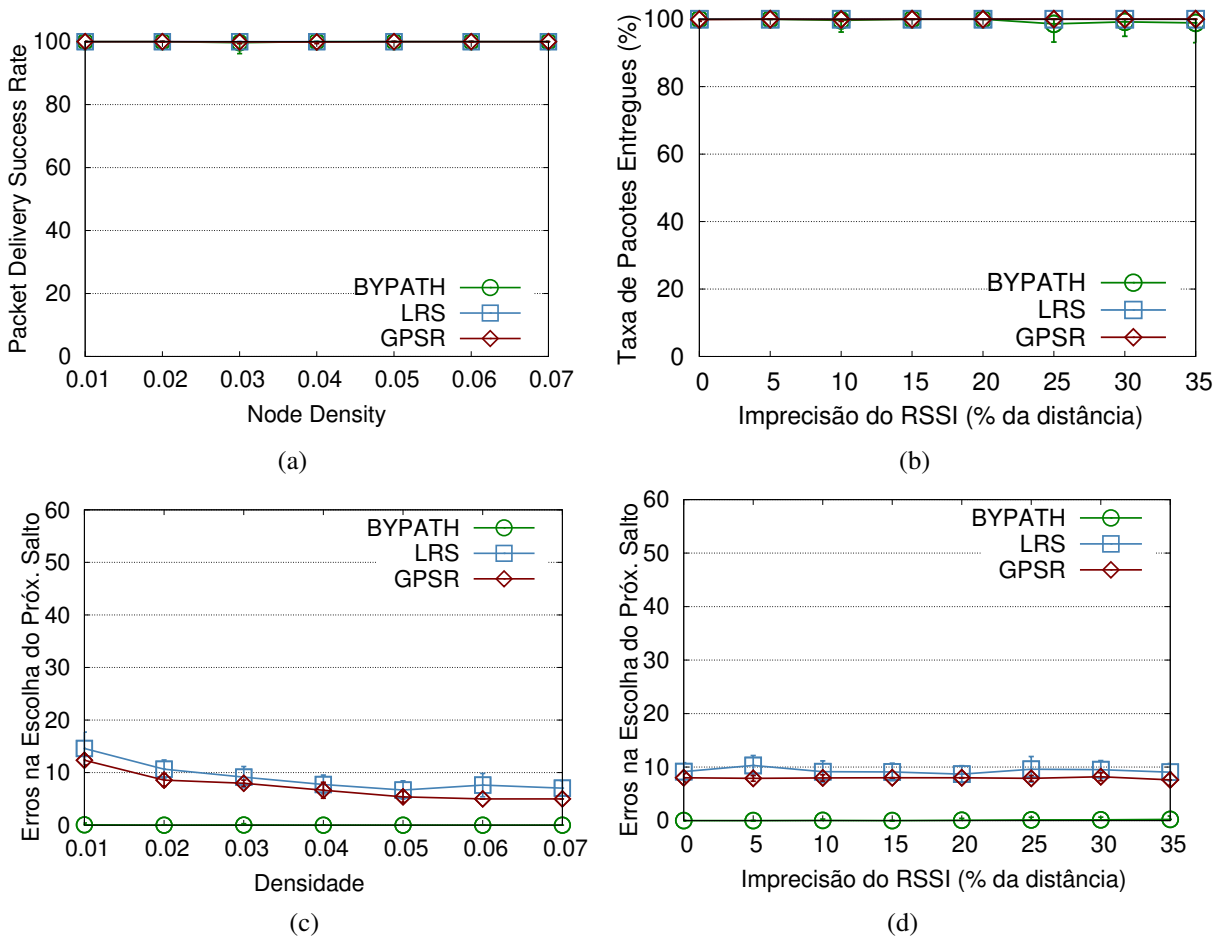


Figura 7.4. Resultados obtidos ao se avaliar a taxa de entrega e a os erros na escolha do próximo salto.

Figuras 7.1, 7.2 e 7.2. Para a validação do experimento, os nós de um lado do buraco não eram capazes se comunicar com os nós do outro lado, portanto a parte oposta do buraco. Durante as simulações, um conjunto limitado de nós estrategicamente posicionados no canto superior direito vai gerar pacotes de dados com eventos detectados.

7.2.2 Impacto da Densidade da Rede

O impacto da densidade da rede foi avaliado variando-se esse parâmetro de 0.01 nós/m^2 até 0.07 nós/m^2. Como é possível observar pelas Figuras 7.4(a) e 7.4(b), todos os algoritmos conseguem entregar acima de 98 % de suas informações ao *sink*, mesmo em densidades altas. Observamos também que o algoritmo BYPATH é capaz de fazer as melhores escolhas geográficas em direção ao *sink* relação ao *sink*, pois além de evitar a região de buracos o algoritmo continua a fazer escolhas gulosas em todos os casos experimentados, conforme ilustrado nas Figuras 7.4(c) e 7.4(d). Já nos algoritmos LRS e GPSR, os pacotes

são encaminhados ao nó preso no buraco e posteriormente encaminhados ao *sink* através de rotas geradas ao longo do buraco. É comum que em densidades menores e mantendo-se a mesma faixa de comunicação, abordagens geográficas enfrentem o problema da baixa quantidade de vizinhos. No entanto, o algoritmo BYPATH ainda opera de forma eficaz na presença do buraco de roteamento, sem a necessidade cálculos adicionais ou *loops* extras a partir do nó preso no buraco, uma vez que o fluxo gerado na abordagem não entra na região do buraco, como já ilustrado e discutido através das Figuras 7.2 e 7.2.

7.2.3 Impacto da Imprecisão do RSSI a Curta Distância

Avaliamos a imprecisão do RSSI variando este parâmetro de 0 % até 35 % da distância real entre os nós regulares. É facil constatar que, conforme ilustra as Figuras 7.4(b) e 7.4(d), a taxa de entrega de pacotes as escolhas geográficas de próximo salto não são afetadas por esta imprecisão. Neste sentido, as características inerentes a cada abordagem contribuem para tais métricas (impressão do RSSI e escolhas geográficas de próximo salto) não sofram interferências importantes. Contudo, ao avaliarmos a quantidade de pacotes transmitida pelos nós percebemos que o algoritmo BYPATH é afetado, como ilustrado na Figura 7.5(a). É importante lembrar que o processo de estimativa de distância entre os vizinhos de um salto e o *sink* é feito a partir dos valores de RSSI propagados pelo *sink* no estágio inicial de funcionamento da abordagem. Assim, a montagem dos fluxos de escoamento de pacotes será influenciada pelo aumento na imprecisão do RSSI já que um nó percebe os valores acrescidos do erro de localização. Desta forma, um nó pode preferir caminhos mais longos devido à imprecisão aplicada à estimativa de distância dos nós em relação ao *sink*.

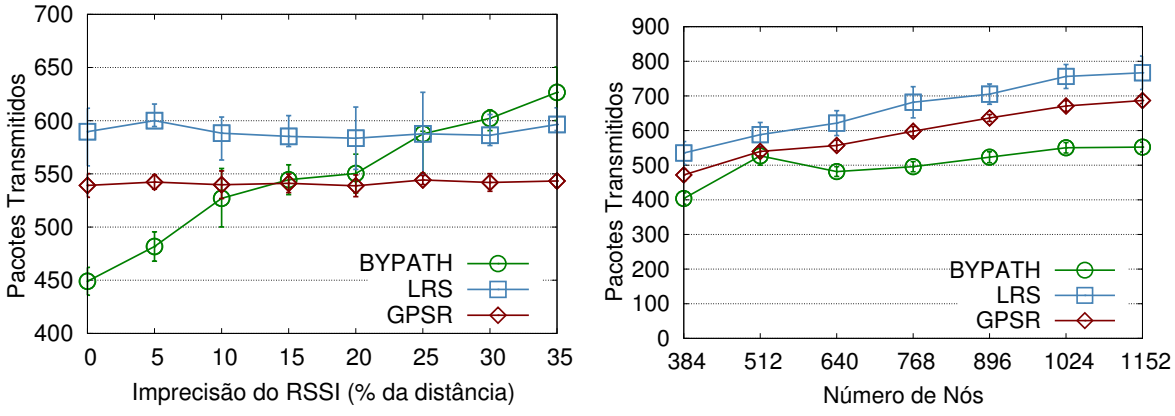


Figura 7.5. Resultados obtidos ao se avaliar a quantidade de pacotes transmitidos.

7.2.4 Impacto da Escalabilidade da Rede

Avaliamos o impacto de escalar a quantidade de nós na rede aumentando o número de nós 384 até 1152 , com uma densidade constante de 0.03 nós/m^2. Além disso, mantivemos o alcance da comunicação entre os vizinhos imediatos em 30 m. Como é possível verificar pela Figura 7.6(c), as três abordagens apresentam comportamento regular, pois o aumento no número de saltos ao se aumentar a quantidade de nós na rede se justifica por existirem mais nós intermediários entre a origem dos pacotes e o destino final. Pelo que ilustra a Figura 7.6(d), é possível constatar que o mesmo comportamento é observado em relação ao tamanho médio dos caminhos de roteamento gerados pela abordagem, já que em nossa abordagem o aumento da quantidade de nós está diretamente relacionado ao aumento de tamanho da grade de sensoreamento. Ainda assim, é possível observar que mesmo quando a quantidade de nós está em 1152 , o algoritmo BYPATH está apto a construir caminhos geográficos de melhor qualidade, menores do que nas duas outras abordagens, mantendo altas taxas de entrega de pacotes. Por fim, observamos que o mecanismo que evita o buraco de roteamento contribui para a redução na quantidade de pacotes transmitidos pelo algoritmo BYPATH, conforme ilustrado na Figura 7.5(b). Neste sentido, os algoritmos LRS e GPSR precisam trocar uma quantidade maior de mensagens entre os vizinhos afim de prover a detecção e o desvio do buraco de roteamento.

7.2.5 Impacto do Raio de Comunicação dos Nós

Nesta seção, avaliamos o impacto do alcance de comunicação dos nós aumentando este parâmetro de 18 m para 42 m. Quando o alcance da comunicação é curto, o número de vizinhos percebidos por um nó será menor. Nesse caso, a escolha do próximo salto será menos eficiente podendo gerar caminhos de roteamento maiores até o *sink*. Como podemos ver nas Figura 7.6(b) e 7.6(a), todas as abordagens apresentam comportamento esperado já que ao aumentarmos o alcance de comunicação dos nós também aumentamos as chances de um nó perceber uma quantidade maior vizinhos. Neste sentido, observa-se uma redução na quantidade de saltos e nos caminhos gerados pelo fluxo natural dos pacotes até o *sink*. Contudo, quando examinamos o gráfico produzido pela Figura 7.6(b), é fácil observar que o algoritmo BYPATH é capaz de entregar cerca de 75 pacotes a menos do que o GPSR, quando o alcance de comunicação dos nós está em 18 m. Esse resultado confirma a eficácia do algoritmo BYPATH, mesmo quando o número de vizinhos é reduzido.

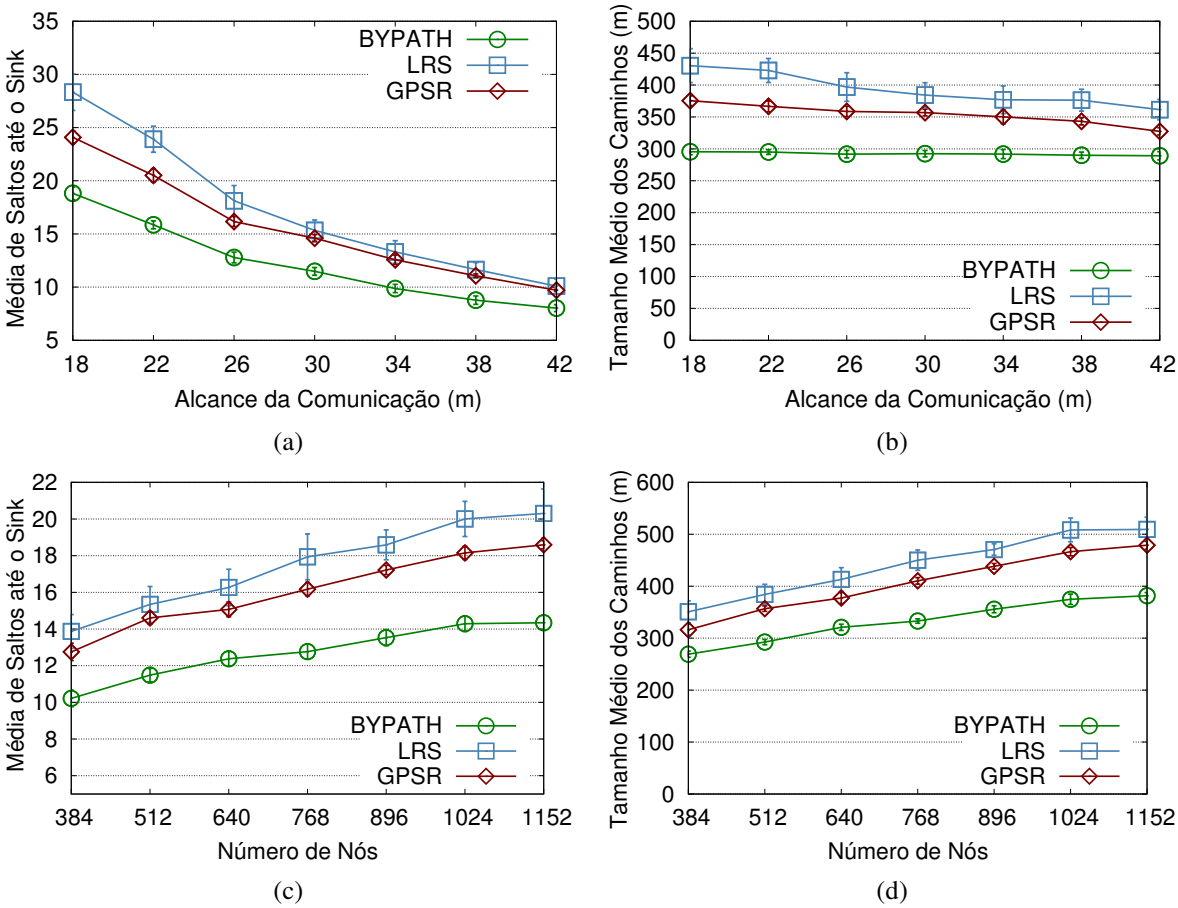


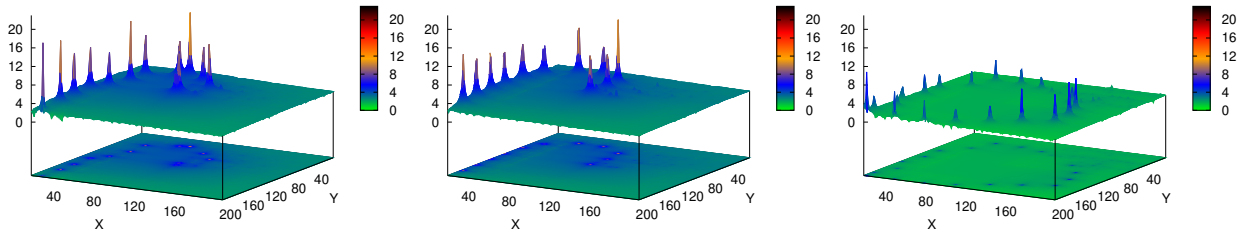
Figura 7.6. Resultados obtidos ao se avaliar a quantidade de saltos e o tamanho médio dos fluxos gerados até o *sink*.

7.2.6 Qualidade dos Fluxos de Escoamento de Pacotes

Nesta seção avaliamos a qualidade dos caminhos gerados entre a origem dos pacotes e o *sink*, computando os saltos de comunicação do trajeto dos pacotes, bem como as distâncias percorridas em múltiplos saltos. Um dos problemas enfrentados pelas abordagens que consideram a existência de buracos (ou lacunas de comunicação) é o aumento no tamanho dos caminhos até o destino final dos pacotes. Na maioria dos casos, os pacotes de dados são encaminhados a um único nó preso para posteriormente serem reencaminhados em múltiplos saltos a partir da adoção de mecanismos que exploram a borda do buraco. Entretanto, os mecanismos que exploram a borda ao longo do buraco são em sua grande maioria dispendiosos às abordagens geográficas, já que podem levar os nós da borda a um esgotamento energético precoce, aumentando o tamanho do buraco e consequentemente aumentando o tamanho dos caminhos até o destino.

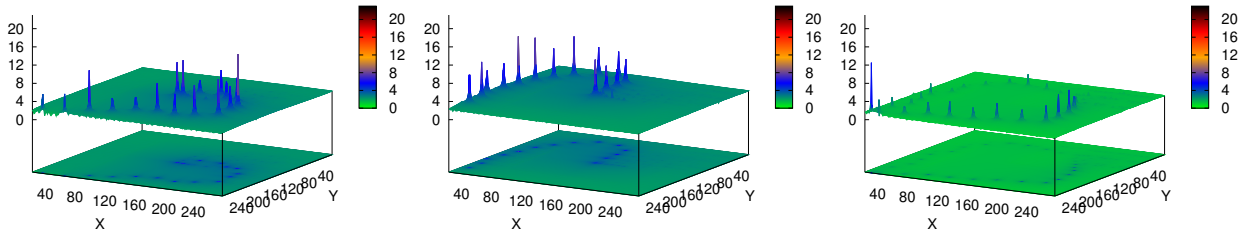
As Figuras 7.6(a) e 7.6(c) mostram que a média de saltos no algoritmo BYPATH é sempre menor que a observada nas duas outras abordagens. Isto se deve ao mecanismo que

evita regiões de buracos sem a necessidade de cálculos ou *loops* extras a partir do nó preso. O tamanho médio dos caminhos gerados até o *sink* também foi avaliado. Conforme ilustrado pelas Figuras 7.6(b) e 7.6(d), o tamanho médio dos fluxos de escoamento de pacotes do algoritmo BYPATH é cerca de 40 % menor do que o observado na abordagem clássica GPSR e 60 % menor do que no LRS. Neste sentido, podemos afirmar que há uma significativa melhora no desempenho da rede em termos de consumo de energia, já que menos transmissões serão efetuadas no BYPATH.



(a) LRS (b) GPSR (c) BYPATH

Figura 7.7. Qualidade dos caminhos de fluxo gerados pelas três abordgens, em uma grade com 512 nós.



(a) LRS (b) GPSR (c) BYPATH

Figura 7.8. Qualidade dos caminhos de fluxo gerados pelas três abordgens, em uma grade com 1152 nós.

Como forma de ilustrar e comparar graficamente os caminhos de fluxo gerados pelas três abordagens, as Figuras 7.7 e 7.8 ilustram a quantidade de transmissões efetuadas por cada em nó, em sua respectiva posição na grade, gerados durante as simulações. Nas figuras é possível ver claramente que os caminhos de roteamento gerados pelo BYPATH evitam o buraco enquanto encaminham os pacotes em direção ao coletor. Também é fácil observar que o BYPATH gera mais uma rota para o escoamento de pacotes em torno do buraco, ao

contrário do que se vê nos algoritmos LRS e GPSR que sempre irão escolher um dos lados para a retransmissão de mensagens a partir do buraco. É possível observar ainda que o número de transmissões do algoritmo BYPATH é significativamente menor do que o observado as outras abordagens, já que não há necessidade de encaminhar pacotes ao buraco.

7.3 Considerações finais

Neste capítulo, propusemos e presentamos um novo algoritmo de roteamento geográfico livre de sistemas de coordenadas, capaz de evitar regiões de buraco em RSSF, chamado BYPATH. Na abordagem, o *sink* é equipado com um dispositivo de comunicação de grande potência para que seja possível alcançar a todos os nós da rede em um salto único. Os valores de RSSI propagados pelo *sink* servem de métrica para a escolha do vizinho geograficamente mais próximo ao *sink*, além da criação de fluxos válidos de escoamento de pacotes que evitam o buraco de roteamento. Neste sentido, o BYPATH é apto a encaminhar pacotes ao redor de uma região de buraco mas sem encaminhar pacotes de dados na direção do buraco.

Um conjunto de experimentos foi realizado para avaliar o algoritmo BYPATH. Os experimentos demonstraram que este novo mecanismo de roteamento geográfico supera a abordagem clássica de detecção e desvio de buracos, GPSR (*Greedy Perimeter Stateless Routing*). O BYPATH não faz uso de atribuição de coordenadas virtuais ou quaisquer outros sistemas de coordenadas e/ou atribuição de posição de nós. Ainda sim, é capaz de criar caminhos eficientes para até o nó *sink*, enquanto circunda a região de buraco. Mostramos que é possível evitar que pacotes de dados sejam encaminhados à regiões com a presença de buracos. Com essa melhoria, nós sensores presos ou localizados ao longo da região de buracos não serão usados intensivamente durante o encaminhamento de mensagens. Essa melhoria otimiza o gasto de energia dos nós, promovendo o balanceamento de carga entre os nós intermediários. Nossos resultados mostram os benefícios introduzidos pelo BYPATH no tocante a evitar regiões com buracos de roteamento em RSSFs.

Capítulo 8

Conclusões

Este capítulo conclui e sumariza a presente tese de doutorado, além de apontar para as futuras direções da pesquisa atual. Na seção 8.1, apresentamos nossas considerações finais sobre as soluções apresentadas e na seção 8.2, apresentamos os direcionamentos futuros pretendidos para esta pesquisa.

8.1 Considerações Finais

Nesta tese, abordamos o problema da ocorrência de regiões de buraco ou lacunas de comunicação em Redes de Sensores Sem Fio como parte de pesquisa de doutorado em Informática, em andamento. As regiões de buraco são um desafio para a utilização eficaz de protocolos de roteamento geográfico no tocante à transmissão de mensagens através da rede. Neste sentido, contribuímos com um conjunto soluções capazes de detectar, desviar e por último evitar buracos de roteamento em Redes de Sensores Sem Fio, a partir do desenvolvimento e implementação dos algoritmos ARESTA, REACT, LRS e BYPATH. Um dos aspectos essenciais deste trabalho é considerar que o nó *sink* está equipado com um dispositivo de comunicação de grande potência, de tal forma que em um único salto todos os nós da rede sejam alcançados.

Afim aproveitar essa arquitetura singular, nossos algoritmos tiram proveito dos valores de RSSI percebidos pelos nós regulares, afim de implementar mecanismos de encaminhamento de pacotes que proporcionam agregação de dados, capazes de contornar buracos de roteamento e por último evitar regiões de buraco. Nossos algoritmos exploram as seguintes estratégias para lidar com regiões de buraco: (a) escolha de um candidato elegível a próximo salto com base em informações locais (tabela de vizinhos); (b) auto-eleição de candidatos à próximo salto, a partir de mensagens de anúncio (*broadcast*) oportunistas que também considera os vizinhos de um salto de distância; (c) contorno de um buraco de roteamento a partir do conhecimento sobre os saltos em direção ao *sink* e; (d) criação de fluxos alternativos

que evitam um buraco de roteamento aproveitando as informações propagadas em múltiplos saltos pelos vizinhos.

Por fim, comprovamos através de experimentos que as soluções presentes nesta tese são capazes de lidar com o fenômeno dos Mínimos Locais (*local minimum*), mantendo altas taxas de entrega de pacotes e ainda provendo agregação de dados durante o processo de encaminhamento de pacotes. Demonstramos ainda que, nossas soluções de contorno de buraco livre de sistemas de coordenadas, superam a abordagem clássica de roteamento por perímetro e reconhecimento GPSR[Karp & Kung, 2000], além de promover balanceamento de carga aos nós em torno do buraco de roteamento.

8.2 Trabalhos Futuros

A partir dos resultados parciais obtidos, pretendemos continuar nossos experimentos, buscando aprimorar o mecanismo de detecção de regiões de buraco. Ainda, pretendemos adequar a última contribuição, que chamamos de BYPATH, de tal forma que o *sink* possa as repassar informações aos nós regulares em múltiplos saltos, preservando sua característica de não depender de sistemas de coordenadas virtuais ou quaisquer outros sistemas de atribuição de coordenadas. No que tange às soluções de contorno utilizadas em nossos algoritmos, as seguintes atividades ainda estão em aberto:

1. Comparar e avaliar nossos algoritmos com outros mecanismos de detecção e desvio de buraco, como: GEAR [Yu et al., 2001], BOUNDHOLE [Fang et al., 2004] e EL-BAR [Le et al., 2012]

2. Aprimorar nosso mecanismo de detecção e contorno de buracos para uso em cenário com outros tipos de buracos (lacunas de comunicação);

3. Avaliar ganhos e desvantagens da utilização de um *sink* de longo alcance nas abordagens de desvio de buracos disponíveis na literatura;

4. Adaptar a metodologia de reconhecimento de regiões de buraco utilizada em nossos algoritmos de forma que seja possível a adoção de um *sink* comum. Neste sentido, ao invés do *sink* alcançar a rede em um único salto, os nós estariam aptos a receber a mesma consulta a partir de múltiplos saltos; e

5. Ampliar o escopo de nossa abordagem para utilização em redes veiculares, com foco em regiões com ausência de conectividade.

Referências Bibliográficas

Abbasi, A. A. & Younis, M. (2007). A survey on clustering algorithms for wireless sensor networks. *Computer Communications*, 30(14-15):2826 – 2841.

Abdelgawad, A. & Bayoumi, M. (2012). *Resource-Aware Data Fusion Algorithms for Wireless Sensor Networks*, volume 118 of *Lecture Notes in Electrical Engineering*. Springer.

Ahmed, N.; Kanhere, S. S. & Jha., S. (2005). The holes problem in wireless sensor networks: A survey. *ACM SIGMOBILE Mobile Computing*.

Akkaya, K. & Younis, M. (2005). A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3:325 – 349.

Akyildiz, I.; Su, W.; Sankarasubramaniam, Y. & Cayirci, E. (2002a). A survey on sensor networks. *IEEE, Communications Magazine*, 40:102 – 114.

Akyildiz, I. F.; Su, W.; Sankarasubramaniam, Y. & Cayirci, E. (2002b). Wireless sensor networks: a survey. *Computer Networks*, 38:393 – 422.

Al Alawi, R. (2011). Rssi based location estimation in wireless sensors networks. In *17th IEEE International Conference on Networks (ICON), 2011*, pp. 118–122.

Al-karaki, J. N. & Kamal, A. E. (2004). Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, 11:6 – 28.

Antil, P. & Malik, A. (2014). Hole detection for quantifying connectivity in wireless sensor networks: A survey,. *Journal of Computer Networks and Communications,*, 2014:11.

Benkic, K.; Malajner, M.; Planinsic, P. & Cucej, Z. (2008). Using rssi value for distance estimation in wireless sensor networks based on zigbee. pp. 303 –306.

Biswas, S. & Morris, R. (2004). Opportunistic routing in multi-hop wireless networks. *SIGCOMM Comput. Commun. Rev.*, 34(1):69 – 74.

Boukerche, A.; Chatzigiannakis, I. & Nikoletseas, S. (2005). Power-efficient data propagation protocols for wireless sensor networks. *Simulation*, 81(6):399 – 411.

Boukerche, A.; Oliveira, H.; Nakamura, E. & Loureiro, A. (2007). Localization Systems for Wireless Sensor Networks. *IEEE Wireless Communications, 2007*, 14(6):6 –12.

Boukerche, A.; Oliveira, H.; Nakamura, E. & Loureiro, A. (2008). A Novel Location-Free Greedy Forward Algorithm for Wireless Sensor Networks. In *IEEE International Conference on Communications (ICC), 2008*.

Caruso, A.; Chessa, S.; De, S. & Urpi, R. (2005). Gps free coordinate assignment and routing in wireless sensor networks. In *In IEEE INFOCOM*, pp. 150 – 160.

Chen, D. & Varshney, P. (2007). A survey of void handling techniques for geographic routing in wireless networks. *Communications Surveys Tutorials, IEEE*, 9(1):50 – 67.

Clark, B. N.; Colbourn, C. J. & Johnson, D. S. (1991). Unit disk graphs. *Discrete Math.*, 86(1-3):165 – 177.

Cohen, N. H.; Cohen, N. H.; Purakayastha, A.; Purakayastha, A.; Turek, J.; Turek, J.; Wong, L.; Wong, L.; Yeh, D. & Yeh, D. (2001). Challenges in flexible aggregation of pervasive data. Technical report, IBM.

Dan, A.; Halder, S. & DasBit, S. (2011). Localization with enhanced location accuracy using rssi in wsn. In *2011 Fifth IEEE International Conference on Advanced Telecommunication Systems and Networks (ANTS)*, pp. 1 – 6.

Di Francesco, M.; Das, S. K. & Anastasi, G. (2011). Data collection in wireless sensor networks with mobile elements: A survey. *ACM Trans. Sen. Netw.*, 8(1):7:1 – 7:31.

Éfren Souza; Nakamura, E.; Oliveira, H. & Figueiredo, C. M. S. (2012). Reducing the impact of location errors for target tracking in wireless sensor networks. *Journal of the Brazilian Computer Society*.

ETH-Zurich, D. C. G. a. (2014). Sinalgo - simulator for network algorithms. In *http://dcg.ethz.ch/projects/sinalgo/*.

Eu, Z. A. & Tan, H.-P. (2012). Adaptive opportunistic routing protocol for energy harvesting wireless sensor networks. In *IEEE International Conference on Communications (ICC), 2012*, pp. 318–322.

Fang, Q.; Gao, J. & Guibas, L. (2004). Locating and bypassing routing holes in sensor networks. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 4, pp. 2458 – 2468.

Ferreira, L. C. B.; Cardieri, P.; Branquinho, O. C. & de Faria, T. T. (2014). Geographic routing by using location algorithm based in signal measurements. In *2014 International Telecommunications Symposium (ITS)*, pp. 1 – 5.

Ghosh, P.; Gao, J.; Gasparri, A. & Krishnamachari, B. (2014). Distributed hole detection algorithms for wireless sensor networks. In *IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), 2014*, pp. 257 – 261.

Goyal, D. & Tripathy, M. (2012). Routing protocols in wireless sensor networks: A survey. pp. 474 – 480.

Holland, M.; Aures, R. & Heinzelman, W. (2006). Experimental investigation of radio performance in wireless sensor networks. In *Wireless Mesh Networks, 2006. WiMesh 2006. 2nd IEEE Workshop on*, pp. 140 –150.

Huo, G. & Wang, X. (2008). An opportunistic routing for mobile wireless sensor networks based on rssi. In *4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08.*, pp. 1 – 4.

Jabeur, N.; Sahli, N. & Khan, I. M. (2013). Survey on sensor holes: A cause-effect-solution perspective. *Procedia Computer Science*, 19:1074 – 1080.

Jacinto, R. M. P. (2012). Modelação da Propagação numa Rede de Sensores sem Fios. Master's thesis, Universidade Nova de Lisboa.

Kaimal, J. C. & Gaynor, J. E. (1983). The boulder atmospheric observatory. *Journal of Climate and Applied Meteorology*, 22(5):863–880.

Kanno, J.; Buchart, J. G.; Selmic, R. R. & Phoha, V. (2009). Detecting coverage holes in wireless sensor networks. In *2009 17th Mediterranean Conference on Control and Automation*, pp. 452–457.

Karp, B. & Kung, H. T. (2000). GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *MOBICOM*, pp. 243 – 254.

Karp, B. N. (2000). *Geographic Routing for Wireless Networks*. PhD thesis, Cambridge, MA, USA. AAI9988566.

Khan, I. F. & Javed, M. Y. (2008). A survey on routing protocols and challenge of holes in wireless sensor networks. In *2008 International Conference on Advanced Computer Theory and Engineering*, pp. 161 – 165.

Kim, Y.-J.; Govindan, R.; Karp, B. & Shenker, S. (2005). Geographic routing made practical. In *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2*, pp. 217 – 230.

Krishnamachari, B.; Estrin, D. & Wicker, S. B. (2002). The impact of data aggregation in wireless sensor networks. pp. 575 – 578.

Kuhn, F.; Moscibroda, T. & Wattenhofer, R. (2004). Unit disk graph approximation. In *Proceedings of the 2004 Joint Workshop on Foundations of Mobile Computing*, DIALM-POMC '04, pp. 17--23, New York, NY, USA. ACM.

Le, N. P.; Hieu, N. T. & Van, N. K. (2012). ELBAR: Efficient Load Balanced Routing Scheme for Wireless Sensor Networks with Holes. In *Proceedings of the Third Symposium on Information and Communication Technology*, SoICT '12, pp. 190 – 199. ACM.

Leong, B.; Liskov, B. & Morris, R. (2007). Greedy virtual coordinates for geographic routing. In *IEEE International Conference on Network Protocols (ICNP), 2007*, pp. 71 – 80.

Li, F.; Zhang, B. & Zheng, J. (2011). Geographic hole-bypassing forwarding protocol for wireless sensor networks. *IET Communications*, 5(6):737 – 744.

Lima, M.; Oliveira, H.; Nakamura, E.; Balico, L. & Loureiro, A. (2013a). Greedy routing and data aggregation in wireless sensor networks. In *Computers and Communications (ISCC), 2013 IEEE Symposium on*, pp. 342 – 347.

Lima, M.; Oliveira, H.; Nakamura, E.; Barreto, R. & Guidoni, D. (2016). Agregação de dados com desvio de buracos para redes de sensores sem fio com sinks de alto alcance. In 34^o *Simposio Brasileiro de Redes de Computadores e Sistemas Distribuidos (SBRC 2016)*, Salvador, Bahia.

Lima, M.; Oliveira, H.; Nakamura, E. & Loureiro, A. (2013b). Roteamento e Agregação de Dados baseado no RSSI em Redes de Sensores sem Fio. In 31^o *Simposio Brasileiro de Redes de Computadores e Sistemas Distribuidos (SBRC 2013)*, Brasilia, DF.

Lima, M. M.; Oliveira, H. & Pazzi, R. W. (2018). A novel RSSI-based algorithm for detect and bypass routing holes in wireless sensor networks. In *2018 IEEE Symposium on Computers and Communications (ISCC) (ISCC 2018)*, Natal, Brazil.

Lima, M. M.; Oliveira, H. A.; Guidoni, D. L. & Loureiro, A. A. (2017). Geographic routing and hole bypass using long range sinks for wireless sensor networks. *Ad Hoc Networks*, pp. 1 – 10.

Lin, S.; Liqin, W. & Zhengwei, Z. (2010). A clustering algorithm based geographic location information for wireless sensor networks. pp. 2588 – 2592.

Loureiro, A. A. F.; Nogueira, J. M. S.; Ruiz, L. B.; Mini, R. A.; Nakamura, E. F. & Figueiredo, C. M. S. (2003). Redes de sensores sem fio. *Simposio Brasileiro de Redes de Computadores e Sistemas Distribuidos (SBRC 2003)*, 21:179 – 226.

Nakamura, E. F.; Loureiro, A. A. F. & Frery, A. C. (2007). Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Computing Surveys*, 39(3).

Nati, M. (2008). *Geographic Routing: Mission Possible*. PhD thesis, University of Rome.

Nguyen, K.-V.; Nguyen, P. L.; Vu, Q. H. & Do, T. V. (2017). An energy efficient and load balanced distributed routing scheme for wireless sensor networks with holes. *Journal of Systems and Software*, 123:92 – 105.

Oliveira, H. A.; Boukerche, A.; Guidoni, D. L.; Nakamura, E. F.; Mini, R. A. & Loureiro, A. A. (2015). An enhanced location-free greedy forward algorithm with hole bypass capability in wireless sensor networks. *J. Parallel Distrib. Comput.*, 77(C):1 – 10.

Park, E.; Bae, D. & Choo, H. (2010). Energy efficient geographic routing for prolonging network lifetime in wireless sensor networks. In *Proceedings of the 2010 International Conference on Computational Science and Its Applications*, ICCSA '10, pp. 285 – 288, Washington, DC, USA. IEEE Computer Society.

Piotrowski, K.; Sojka, A. & Langendoerfer, P. (2012). Long-range wireless sensor nodes – lessons learned. In *2012 6th European Conference on Antennas and Propagation (EUCAP)*, pp. 57 – 61.

Rappaport, T. (2002). *Wireless communications: principles and practice*. Prentice Hall communications engineering and emerging technologies series. Prentice Hall PTR.

Revathi, A. & Malathy, N. (2016). VAGR – Void aware in geographic routing for wireless sensor networks. In *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, pp. 1– 6.

Sahoo, P. K. & Liao, W. C. (2015). HORA: A Distributed Coverage Hole Repair Algorithm for Wireless Sensor Networks. *IEEE Transactions on Mobile Computing*, 14(7):1397–1410.

Santi, P. (2005). Topology control in wireless ad hoc and sensor networks. *ACM Comput. Surv.*, 37(2):164 – 194.

Seada, K.; Zuniga, M.; Helmy, A. & Krishnamachari, B. (2004). Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, pp. 108 – 121, New York, NY, USA. ACM.

Singh, S. K.; Singh, M. P. & Singh, D. K. (2010). Routing Protocols in Wireless Sensor Networks – A Survey. *International Journal of Computer Science & Engineering Survey (IJCSES)*, 1(2):63 – 83.

Takagi, H. & Kleinrock, L. (1984). Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, 32(3):246 – 257.

Tollefson, J. (2010). A towering experiment an ambitious project to track greenhouse gases from a perch high above the amazon forest will provide crucial data - but only if scientists can get it built. *Nature*, 467(386).

Villalba, L. J.; Orozco, A. L.; Cabrera, A. & Barenco, C. J. (2009). Routing protocols in wireless sensor networks. *Sensors*, 9(11):8399 – 8421.

Wang, B. (2011). Coverage problems in sensor networks: A survey. *ACM Comput. Surv.*, 43(4):1 – 32.

Wang, X.; Sun, F. & Kong, X. (2009). Research on optimal coverage problem of wireless sensor networks. In *2009 WRI International Conference on Communications and Mobile Computing*, volume 1, pp. 548 – 551.

Werth, D.; Buckley, R.; Zhang, G.; Kurzeja, R.; Leclerc, M.; Ferro Duarte, H.; Parker, M. & Watson, T. (2015). Quantifying the local influence at a tall tower site in nocturnal conditions. *Theoretical and Applied Climatology*, 127.

Willis, S. & Kikkert, C. (2005). Radio propagation model for long-range ad hoc wireless sensor network. In *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on*, volume 1, pp. 826 – 832.

Xing-Sheng, L.; Gaynor, J. E. & Kaimal, J. C. (1983). A study of multiple stable layers in the nocturnal lower atmosphere. *Boundary-Layer Meteorology*, 26(2):157--168.

Yang, J. & Fei, Z. (2010). HDAR: Hole Detection and Adaptive Geographic Routing for Ad Hoc Networks. In *2010 Proceedings of 19th International Conference on Computer Communications and Networks*, pp. 1 – 6.

Yi, C.; Davis, K. J.; Bakwin, P. S.; Berger, B. W. & Marr, L. C. (2000). Influence of advection on measurements of the net ecosystem-atmosphere exchange of co2 from a very tall tower. *Journal of Geophysical Research: Atmospheres*, 105(D8):9991 – 9999.

Yoneki, E. & Bacon, J. (2005). A survey of Wireless Sensor Network technologies: research trends and middleware's role. Technical Report UCAM-CL-TR-646, University of Cambridge, Computer Laboratory.

You, J.; Lieckfeldt, D.; Han, Q.; Salzmann, J. & Timmermann, D. (2009a). Look-ahead geographic routing for sensor networks. In *IEEE International Conference on Pervasive Computing and Communications (PerCom), 2009*, pp. 1 – 6.

You, J.; Lieckfeldt, D.; Reichenbach, F. & Timmermann, D. (2009b). Context-aware geographic routing for sensor networks with routing holes. In *2009 IEEE Wireless Communications and Networking Conference*, pp. 1 – 6.

Yu, F.; lee, E.; Choi, Y.; Park, S.; Lee, D.; tian, Y. & Kim, S.-H. (2007). A modeling for hole problem in wireless sensor networks. In *Proceedings of the 2007 International Conference on Wireless Communications and Mobile Computing*, IWCMC '07, pp. 370 – 375, New York, NY, USA. ACM.

Yu, Y.; Govindan, R. & Estrin, D. (2001). Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks. Technical report.

Zennaro, M.; Bagula, A.; Gascon, D. & Noveleta, A. B. (2010). *Planning and Deploying Long Distance Wireless Sensor Networks: The Integration of Simulation and Experimentation*, pp. 191 – 204. Springer Berlin Heidelberg, Berlin, Heidelberg.

Zhao, L.; Liu, G.; Chen, J. & Zhang, Z. (2009). Flooding and directed diffusion routing algorithm in wireless sensor networks. In *Ninth International Conference on Hybrid Intelligent Systems, 2009. HIS '09*, volume 2, pp. 235 – 239.

Zhu, Y.; Zhang, B.; Yu, F. & Ning, S. (2009). A rssi based localization algorithm using a mobile anchor node for wireless sensor networks. In *International Joint Conference on Computational Sciences and Optimization, 2009. CSO 2009.*, volume 1, pp. 123 – 126.