



Universidade Estadual de Feira de Santana
Programa de Pós-Graduação em Computação Aplicada

Posicionamento de Múltiplos Sinks em Redes de Sensores Sem Fio com Priorização

João Paulo Just Peixoto

Feira de Santana

2016



Universidade Estadual de Feira de Santana
Programa de Pós-Graduação em Computação Aplicada

João Paulo Just Peixoto

Posicionamento de Múltiplos Sinks em Redes de Sensores Sem Fio com Priorização

Dissertação apresentada à Universidade Estadual de Feira de Santana como parte dos requisitos para a obtenção do título de Mestre em Computação Aplicada.

Orientador: Dr. Daniel G. Costa

Feira de Santana

2016

Ficha Catalográfica – Biblioteca Central Julieta Carteado

Peixoto, João Paulo Just
P43p Posicionamento de múltiplos Sinks em redes de sensores sem fio com
priorização/João Paulo Just Peixoto. - Feira de Santana, 2016.

56 f.: il.

Orientador: Daniel G. Costa

Dissertação (Mestrado) – Universidade Estadual de Feira de Santana,
Programa de Pós-graduação em Computação Aplicada, 2016.

1. Computação. 2. Redes de sensores sem fio - princípios. I. Costa
Daniel G., orient. II. Universidade Estadual de Feira de Santana. III.
Título.

CDU: 681.3

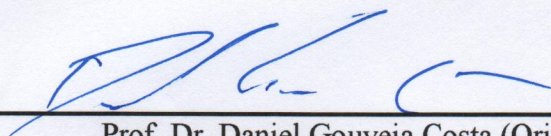
João Paulo Just Peixoto

**Posicionamento de Múltiplos Sinks em Redes de Sensores Sem Fio
com Priorização**

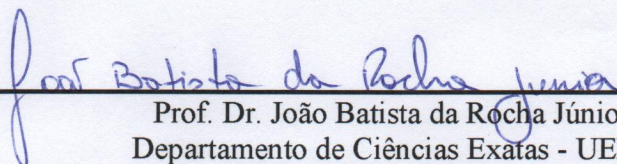
Dissertação apresentada à Universidade Estadual de Feira de Santana como parte dos requisitos para a obtenção do título de Mestre em Computação Aplicada.

Feira de Santana, 15 de dezembro de 2015

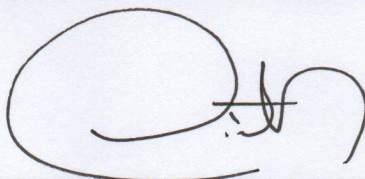
BANCA EXAMINADORA



Prof. Dr. Daniel Gouveia Costa (Orientador)
Departamento de Tecnologia - UEFS



Prof. Dr. João Batista da Rocha Júnior
Departamento de Ciências Exatas - UEFS



Prof. Dr. Cristian Duran-Faundez
Universidade del Bio-Bio (Chile)

Abstract

Wireless Sensor Networks can be used for environment monitoring, which may be in indoor scenarios such as in industries, warehouses and buildings, or in outdoor scenarios, just like forests, volcanos and cities. Due to its energy supply limitations, researchers have been developing several approaches to reduce energy consumption in sensors and extend their lifetime. One of the ways to improve energy usage in a Wireless Sensor Network is by using multiple mobile sinks. In this work, a method to position multiple sinks in a relevance-based Wireless Sensor Network is proposed. After several simulations, it was observed that in applications where sensors have different importance levels, the network lifetime can be extended by using this approach.

Keywords: sink mobility, wireless sensor networks, energy consumption, QoE

Resumo

Redes de Sensores Sem Fios podem ser utilizadas para o monitoramento de ambientes, sejam eles internos como indústrias, galpões, construções, ou externos como florestas, vulcões, cidades, entre outros. Devido à sua limitação no fornecimento de energia, pesquisadores tem desenvolvido diversas abordagens para reduzir o consumo energético nos sensores e prolongar seu tempo de vida. Uma das formas de melhorar o consumo de energia em uma Rede de Sensores Sem Fios é através do uso de múltiplos *sinks* móveis. Neste trabalho, é proposto um método para posicionamento de múltiplos *sinks* em uma Rede de Sensores Sem Fios baseada em relevâncias. Após diversas simulações, foi observado que em aplicações onde sensores possuem diferentes níveis de importância, o uso dessa abordagem aumenta o tempo de vida da rede.

Palavras-chave: mobilidade de sinks, redes de sensores sem fios, consumo de energia, QoE

Prefácio

Esta dissertação de mestrado foi submetida à Universidade Estadual de Feira de Santana (UEFS) como requisito parcial para obtenção do grau de Mestre em Computação Aplicada.

A dissertação foi desenvolvida dentro do Programa de Pós-Graduação em Computação Aplicada (PGCA), tendo como orientador o Dr. **Daniel G. Costa**.

Agradecimentos

Agradeço a Deus por ter me dado forças e sabedoria para terminar mais essa etapa da minha vida. À minha esposa e meu filho, por ter me dado momentos de alegria e distração durante esta caminhada, evitando que eu virasse um robô. A meu pai, pelo incentivo e ter possibilitado minha ida ao congresso em Paris. E ao meu orientador, por ter feito um trabalho excelente durante esses dois anos de pesquisa e desenvolvimento.

A todos, meu muito obrigado!

Dedico este trabalho à minha falecida avó, Fausta Eduardina de Oliveira Peixoto,
por ter incentivado meus estudos desde a infância.

Sumário

Abstract	i
Resumo	ii
Prefácio	iii
Agradecimentos	iv
Sumário	vii
Lista de Publicações	viii
Lista de Tabelas	ix
Lista de Figuras	x
Lista de Abreviações	xi
1 Introdução	1
2 Fundamentação Teórica	4
2.1 Princípios das RSSF	6
2.1.1 Aplicações	6
2.1.2 Protocolos	8
2.1.3 Consumo de energia	11
2.1.4 Transmissão de dados multimídia	13
2.1.5 Mobilidade de <i>sinks</i>	15
2.2 Múltiplos sinks	16
2.3 Otimização com QoE	19
3 Metodologia	21
3.1 Formalização do problema	22
3.2 Procedimentos	22
3.3 Ferramentas	23

4	Posicionamento de <i>sinks</i> usando priorização	25
4.1	Posicionamento de um único <i>sink</i>	25
4.2	Setorização de RSSF	27
4.3	Setorização baseada em relevâncias	30
4.3.1	Método proposto	31
4.4	Posicionamento de múltiplos <i>sinks</i>	32
4.5	Restrições de posicionamento	34
4.5.1	Posicionamento com ZPs pré-determinadas	34
4.5.2	Posicionamento com vias de movimentação	35
4.5.3	Zonas Não Conectadas	36
4.5.4	Rotas para movimentação	37
5	Resultados Experimentais	39
5.1	Comparação dos métodos de inicialização <i>Forgy</i> , <i>Random partition</i> e <i>Relevance-based</i>	39
5.2	Simulação	40
5.2.1	Simulação sem restrições de posicionamento	42
5.2.2	Simulação com restrições de posicionamento	44
5.3	Discussão dos resultados	49
6	Considerações Finais	50
	Referências Bibliográficas	52

Lista de Publicações

PEIXOTO, J. P. J.; COSTA, D. G. *Wireless Visual Sensor Networks for Smart City Applications: A Relevance-based Approach for Multiple Sinks Mobility*. Future Generation Computer Systems. 19 p. (Submetido para publicação)

PEIXOTO, J. P. J.; COSTA, D. G. *QoE-Aware Multiple Sinks Mobility in Wireless Sensor Networks*. IEEE/IFIP 7th International Conference on New Technologies, Mobility and Security, 2015, Paris. IEEE/IFIP NTMS 2015.

PEIXOTO, J. P. J.; COTIAS, B.; COSTA, D. G. *A Simulation Tool to Support Teaching of Image Transmissions over Wireless Sensor Networks using RAIT Protocol*. WEIBASE - Workshop de Educação em Computação, 2015, Salvador. XV Escola Regional de Computação Bahia, Alagoas e Sergipe, 2015.

Lista de Tabelas

5.1	Taxa de geração de pacotes de acordo com níveis de relevância. . . .	43
5.2	Programação de atividades dos sensores.	45

Lista de Figuras

2.1	Exemplos de dispositivos de uma RSSF.	4
2.2	Exemplo de uma RSSF.	5
2.3	Exemplo de uma RSMSF.	14
2.4	Exemplo de disposição de múltiplos <i>sinks</i> em uma RSSF.	17
4.1	Movimentação do <i>sink</i> com uso de relevâncias.	26
4.2	Quatro <i>sinks</i> em grade.	32
4.3	Correção da posição de um <i>sink</i> em uma ZP.	34
4.4	Projeção da nova posição de um <i>sink</i> em uma via.	36
4.5	Exemplos de ZNC e ZC.	37
4.6	Exemplo de uma rota calculada com o algoritmo de Dijkstra.	38
5.1	Posicionamento dos <i>sinks</i> usando métodos de inicialização convencionais.	40
5.2	Posicionamento dos <i>sinks</i> usando o método de inicialização <i>Relevance-based</i>	41
5.3	Posicionamento dos sensores na simulação, com destaque para sensores com $RL > 0$	43
5.4	Posições iniciais dos <i>sinks</i> nas simulações.	43
5.5	Consumo de energia nas simulações sem restrições de posicionamento.	44
5.6	Configuração da Rede 1.	45
5.7	Configuração da Rede 2.	46
5.8	Consumo de energia na Rede 1.	47
5.9	Consumo de energia na Rede 2.	47
5.10	Número médio de saltos na Rede 2.	48

Lista de Abreviações

Abreviação	Descrição
ACT	Adaptive Compression-based Congestion Control Technique
ALS	Anchor Location Service
CH	Cluster Head
CLDP	Cross-Link Detection Protocol
CSV	Comma-separated Values
DCA	Direct Communication Area
DS-MAC	Sensor Medium Access Control Protocol with Dynamic Duty Cycle
ELBR	Energy Level Based Routing Algorithm
IEEE	Institute of Electrical and Electronics Engineers
LEACH	Low Energy Adaptive Clustering Hierarchy
MAC	Medium Access Control
MCA	Multi-hop Communication Area
MPMPS	Multi-Priority Multi-Path Selection
PBR	Primary Based Routing Algorithm
QoE	Quality of Experience
RAIT	Reliable Asynchronous Image Transmission
RL	Relevance Level
RSSF	Rede de Sensores Sem Fios
RSMSF	Rede de Sensores Multimídia Sem Fios
RSVSF	Rede de Sensores Visuais Sem Fios
S-MAC	Sensor Medium Access Control
STCP	Sensor Transmission Control Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VANET	Veicular Ad-Hoc Network
WSN-HEAP	Wireless Sensor Network Powered by Ambient Energy Harvesting
XML	Extensible Markup Language
ZC	Zona Conectada
ZNC	Zona Não Conectada
ZP	Zona Proibida

Capítulo 1

Introdução

Redes de Sensores Sem Fios (RSSF) ainda é um tema de pesquisa em destaque. Em geral, suas aplicações permitem o monitoramento de ambientes diversos, principalmente onde o acesso por seres humanos é difícil ou impossível, como florestas densas, vulcões, ambientes subterrâneos, áreas industriais de alta periculosidade, etc [Akyildiz et al. 2002]. Este tipo de rede é composto por nós equipados com baterias, sensores, processador, memória, rádio e atuador [Yick et al. 2008], além de nós coletores (*sinks*), responsáveis por receber os dados transmitidos pelos nós sensores.

Sensores de uma RSSF podem possuir algumas restrições, sobretudo no poder de processamento, na capacidade de armazenamento e nos recursos de comunicação. Por serem alimentados com baterias e, geralmente, instalados em áreas remotas onde o acesso humano é inviável, esses sensores devem economizar o máximo de energia possível. Devido a isso, seus elementos (processador, radiotransmissor, unidade de armazenamento, etc.) possuem baixas capacidades em comparação com *hardwares* computacionais convencionais. Além disso, os sensores geralmente são equipamentos descartáveis, o que exige o uso de componentes de baixo custo [Costa 2013].

Devido às limitações das redes de sensores, abordagens específicas devem ser consideradas ao implementar protocolos de comunicação e transporte, métodos de posicionamento de sensores e *sinks*, compressão de dados, etc. Os protocolos e algoritmos comumente usados em redes de computadores convencionais, por exemplo, são inviáveis em RSSF [Akyildiz et al. 2002], o que exige o desenvolvimento de novas técnicas.

O consumo elevado de energia é um dos principais problemas em uma RSSF, o que leva os pesquisadores a constantemente pensarem em novas abordagens para economia de energia. Dentre elas, o uso de múltiplos *sinks* e o posicionamento planejado deles é algo utilizado com frequência para a solução ou minimização deste problema [Basagni et al. 2007].

Em uma RSSF, os sensores em volta do *sink* são responsáveis por encaminhar dados oriundos de outros nós, além de fazer as medições do ambiente ao seu redor. Por

ter essa dupla tarefa, seu consumo de energia é mais elevado em comparação aos outros sensores que fazem apenas o monitoramento [Saad et al. 2009]. Quando a energia desses sensores acaba e eles são desligados, os nós mais distantes perdem a comunicação com o *sink*, o que acaba por impedir o funcionamento de toda a rede, já que os dados obtidos não chegarão mais ao seu destino.

Algumas abordagens para reduzir o consumo de energia nesses nós próximos ao *sink* consideram o uso de múltiplos *sinks* e o posicionamento planejado deles [Almalkawi et al. 2010]. O uso de múltiplos *sinks* pode balancear o consumo de energia, já que haverá diversos caminhos para que os sensores enviem seus dados. Já o posicionamento planejado pode definir um local ideal para o *sink*, de forma que o consumo de energia dos sensores seja reduzido, além de permitir que o coletor se movimente pela rede de acordo com a demanda de transmissão. Porém, nenhuma das abordagens existentes leva em consideração a relevância dos sensores da rede [Costa et al. 2014]. Dessa forma, nós mais importantes para a aplicação nem sempre serão priorizados, o que poderá trazer um atraso na sua comunicação, ou ainda, permitir que sua energia se esgote rapidamente, acarretando na perda de um sensor relevante.

Com o desenvolvimento de um método de posicionamento de múltiplos *sinks* baseado em relevância, será possível aumentar o tempo de vida da rede, priorizando os sensores que mais importam para a aplicação. Novas possibilidades poderão surgir com o uso desse método, como coleta eficiente de imagens em redes de vigilância pública, detecção rápida de intrusão, menor latência em monitoramento industrial de alto risco, etc.

Ao priorizar os sensores mais importantes para a aplicação, a abordagem desenvolvida nesse trabalho poderá aumentar o tempo de vida de uma RSSF de forma significativa. Nesse caso específico, aumentar o tempo de vida significa manter os sensores que mais interessam à aplicação ativos por mais tempo; não simplesmente manter sensores quaisquer funcionando. Os objetos de monitoramento da aplicação serão priorizados com essa abordagem.

Assim, será proposto nesse trabalho um método inovador para posicionamento e movimentação de múltiplos *sinks*, levando em consideração a relevância dos sensores da rede. Esse método será capaz de definir quais as melhores posições para que os *sinks* priorizem os sensores mais importantes da rede e quais caminhos eles deverão seguir de forma a se movimentarem, considerando também restrições de posicionamento no ambiente que está sendo monitorado.

Para alcançar este objetivo, dois marcos devem ser atingidos:

1. Obtenção de um método para definição das posições de n *sinks* em uma RSSF com múltiplos *sinks*.
2. Obtenção de um método para, após a definição das posições de cada *sink*, traçar um caminho para que ele se movimente do seu ponto de origem até a

posição de destino, levando em consideração obstáculos e restrições de movimentação no ambiente, como ruas em uma cidade, por exemplo.

Com o uso dessa abordagem, além de aumentar o tempo de vida da rede, pretende-se também diminuir a latência da comunicação entre os sensores e os *sinks*, exigindo menos saltos para que os dados dos nós transmissores cheguem ao destino.

Esse texto está organizado da seguinte forma: no Capítulo 2, a fundamentação teórica das RSSF é apresentada e a metodologia do trabalho é descrita no Capítulo 3. O Capítulo 4 trata da abordagem proposta e apresenta o algoritmo desenvolvido, enquanto o Capítulo 5 descreve as simulações executadas e aborda os resultados obtidos. Por último, o Capítulo 6 conclui este trabalho.

Capítulo 2

Fundamentação Teórica

Em uma RSSF, diversos nós sensores são espalhados pelo ambiente de monitoramento alvo (uma floresta, uma área urbana, uma indústria, o topo de um vulcão, etc.). Os sensores são responsáveis por fazer medições do ambiente, coletando dados relevantes à aplicação como umidade, temperatura, luminosidade, nível de ruídos sonoros, níveis de fumaça, vibrações, etc [Yick et al. 2008]. Para fazer estas medições, o *hardware* dos nós deve ser dotado de sensores específicos, de acordo com o objetivo da aplicação. A Figura 2.1 apresenta dois exemplos de dispositivos para RSSF: PSU mote (esquerda) e Crossbow TelosB (direita).

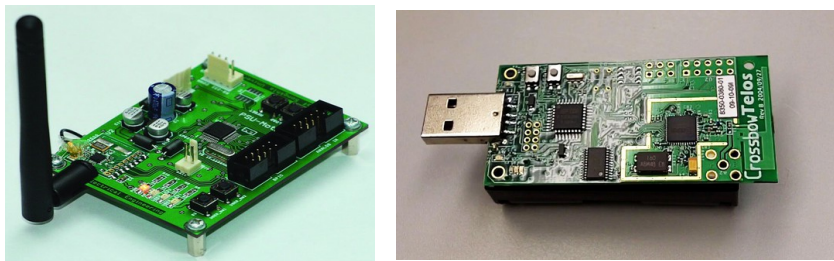


Figura 2.1: Exemplos de dispositivos de uma RSSF.

Os nós sensores podem ser espalhados em um ambiente de forma planejada ou aleatória. Em implantações planejadas, as posições de cada sensor é pré-determinada e então eles são instalados nessas posições. Já em implantações aleatórias, os sensores são espalhados sem se preocupar com seu posicionamento, simplesmente despejando os sensores pela área de interesse. Geralmente esse tipo de implantação é necessária em locais onde o acesso é impossível, inviável ou difícil, como em vulcões, por exemplo [Yick et al. 2008].

Após as medições, os sensores enviam os dados para o *sink*, que geralmente possui conexão com o mundo exterior. Esses tipos de dados correspondem a valores numéricos e são chamados de dados escalares (sensores que coletam esses tipos de dados geralmente são chamados de sensores escalares). Como as redes de sensores podem

ser bastante extensas, na maioria das vezes os sensores não terão uma comunicação direta com o *sink*, portanto, para que os dados cheguem a ele, os sensores precisam trabalhar em conjunto repassando os pacotes de rede até que estes alcancem o destinatário. Esse tipo de transmissão de dados é chamada de transmissão salto a salto (*hop-by-hop transmission*). A Figura 2.2 apresenta um exemplo de uma RSSF onde vários sensores coletam dados e os enviam até o *sink*, encaminhando os pacotes através de nós intermediários.

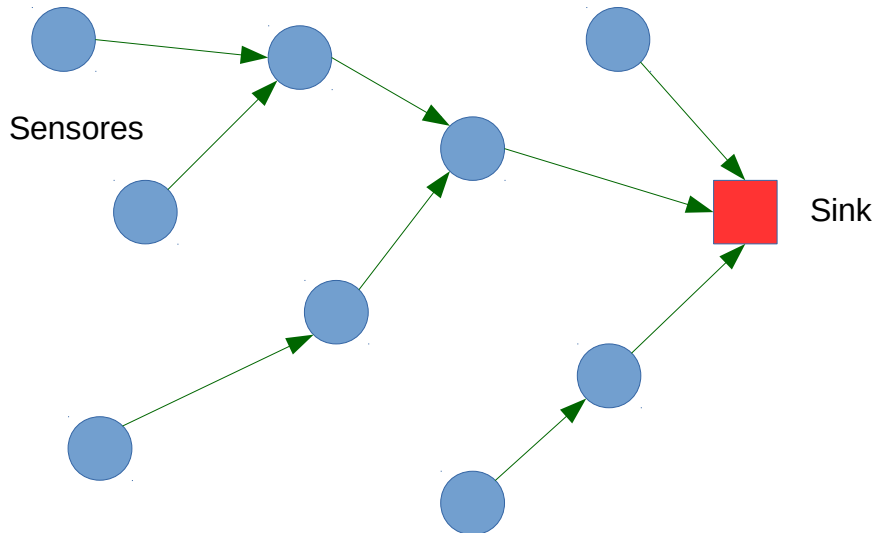


Figura 2.2: Exemplo de uma RSSF.

Em transmissões salto a salto, os sensores intermediários (que estão entre a fonte de dados e o *sink*) funcionam como encaminhadores. Cada pacote de dados que é transmitido por um nó que está distante do *sink* é repassado pelos sensores intermediários, formando um caminho sem fios, por onde os dados trafegam até o destino (de forma semelhante aos roteadores que formam a *Internet*) [Baronti et al. 2007].

Sensores multimídia formam Redes de Sensores Multimídia Sem Fio (RSMSF). Estes tipos de sensores são usados para coletar imagens estáticas, vídeos e/ou áudio [Akyildiz et al. 2002]. Algumas de suas aplicações incluem detecção de movimentos, detecção de intrusão, sistemas de controle de tráfego, telemedicina, etc [Almalkawi et al. 2010]. Tais tipos de dados (imagens, áudio e vídeo) requerem mais banda de tráfego da rede e geram pacotes de dados maiores, portanto, protocolos de transporte mais robustos são utilizados para estes tipos de redes [Almalkawi et al. 2010].

Por formarem pacotes de dados pequenos em comparação com pacotes transmitidos por sensores multimídia [Almalkawi et al. 2010], dados escalares podem ser transmitidos pela rede utilizando potencialmente menos recursos [Yick et al. 2008].

As próximas seções apresentam os princípios das RSSF além de suas aplicações e desafios. Serão tratados detalhes pertinentes à pesquisa deste trabalho, os quais

devem ser considerados ao propor abordagens em mobilidade de *sinks* nas redes de sensores.

2.1 Princípios das RSSF

As RSSF diferem das redes convencionais em vários aspectos, desde suas aplicações até às restrições que o *hardware* dos sensores impõe às técnicas de processamento de dados, transporte de rede, comunicação entre nós, dentre outros. Os próximos tópicos tratam dessas características, apresentando-as e exemplificando com aplicações e abordagens já desenvolvidas.

2.1.1 Aplicações

As aplicações possíveis em RSSF são inúmeras. Diversos pesquisadores estão continuamente desenvolvendo protocolos e algoritmos para este tipo de rede, abrindo novas possibilidades e gerando novas ideias para suas utilizações.

Podemos categorizar as aplicações em RSSF em monitoramento e rastreamento [Yick et al. 2008]. Aplicações de monitoramento são usadas para monitorar ambientes, verificando variáveis como temperaturas, umidade, luminosidade, etc., enquanto aplicações de rastreamento são usadas para localizar animais, pessoas, veículos, dentre outros.

Um exemplo de aplicação de monitoramento é o estudo de caso apresentado em [Tolle et al. 2005]. Neste trabalho, sensores foram instalados ao longo de uma sequoia localizada na cidade de Sonoma, Califórnia. Durante um período de 44 dias, os sensores monitoraram a temperatura do ar, umidade relativa e a radiação solar fotossinteticamente ativa. Com isso, foi possível observar as variações e a dinâmica do clima na área monitorada, algo que não era tão simples de ser feito, de acordo com conversas com biólogos locais.

Já a aplicação descrita em [Krishnamurthy et al. 2005] trata do uso de uma RSSF em uma fábrica de semicondutores e em um petroleiro, em ambos os casos, usada para monitorar o funcionamento de equipamentos industriais. Os sensores liam as assinaturas de vibrações dos equipamentos para prever falhas, ajudando na manutenção preventiva dos dois ambientes. Os autores concluíram que o uso de uma RSSF é bastante viável para a prática da manutenção preventiva em ambientes hostis, trazendo dados de alta qualidade a um custo relativamente baixo de instalação e operação.

Em se tratando de aplicações de rastreamento, podemos citar o trabalho em [Simon et al. 2004]. Chamado de *PinPtr*, o sistema proposto usa diversos sensores espalhados em uma área urbana para detectar atiradores de rifles. Os sensores

empregados podem detectar as ondas de choque sonoras causadas por um disparo. Ao manter uma rede bastante espalhada, os autores conseguiram obter uma precisão média de 1,3 metros. Eles ainda comparam o *PinPtr* com sistemas existentes que trabalham de forma centralizada. Por ter diversos sensores instalados em pontos distintos na área de interesse, o *PinPtr* pôde superar sistemas atuais de detecção de atiradores em ambientes urbanos.

Outro exemplo é o *SensEye* [Kulkarni et al. 2005], uma rede de sensores visuais em várias camadas para vigilância. Em cada camada, um grupo de sensores específicos é instalado para exercer uma determinada função na aplicação. Por exemplo, em uma rede de duas camadas, nós com câmeras de baixo custo ficariam na camada inferior, ativando câmeras de alta definição na camada superior, apenas após a ocorrência de algum evento específico. Dessa forma, o custo da rede é reduzido, já que não é necessário sensores mais caros em toda a rede. No caso do *SensEye*, um sistema de três camadas é capaz de realizar a detecção, o reconhecimento e o rastreamento de objetos. Para efetuar a detecção de objetos, as câmeras devem comparar os quadros e analisar qualquer mudança entre eles. Esse procedimento deve ser feito de forma a ter uma baixa latência na detecção, ou seja, o sistema deve detectar um objeto o mais rápido possível, antes que ele deixe a sua área de cobertura. A melhor forma de alcançar esse objetivo é manter as câmeras ligadas o tempo inteiro, porém, isso acarreta em um alto consumo de energia. A solução no *SensEye* foi fazer com que as câmeras revezassem o período de atividade e inatividade, com um tempo curto entre os períodos de atividade. Os autores concluíram que nesse esquema de funcionamento de várias camadas, onde sensores de funções específicas só são ativados quando necessário, o consumo de energia pode ser reduzido.

RSSF também podem ser usadas para criação de Redes *Ad Hoc* Veiculares (*Vehicular Ad Hoc Networks* - VANETs). Este é um tipo de rede onde existe uma comunicação entre veículos ou entre um veículo e uma unidade fixa na rodovia. Carros equipados com transceptores de rádio apropriados funcionam como sensores em uma RSSF, trocando informações entre si ou com as unidades fixas [Hartenstein e Laberteaux 2008]. Diversas aplicações podem fazer uso deste conceito. Carros parados em um congestionamento podem se comunicar com carros que estão chegando, avisando sobre a situação do local. Esses carros também se comunicam com outros e a informação sobre um engarrafamento se propaga pela rodovia. Isso pode evitar que o congestionamento se amplie e também acidentes que podem ocorrer quando um veículo se aproxima em alta velocidade a um congestionamento. Ao detectar um acidente, carros ou unidades fixas na rodovia podem comunicar outros veículos sobre o fato antes que eles cheguem ao local. Isso pode fazer com que os carros tomem caminhos alternativos, por exemplo. Sinais de trânsito podem também ser informados eletronicamente através de transmissões via a rede de veículos. Mesmo que as placas físicas sejam danificadas, o condutor ainda receberá informações sobre sinalização através de um computador de bordo. Além de avisos que podem evitar acidentes, esse tipo de rede pode também usar as comunicações sem fio para prover entretenimento na rodovia, como *podcasts*, rádios locais

e até acesso à *Internet*.

2.1.2 Protocolos

Por possuírem um hardware bastante reduzido em termos de recursos computacionais e energéticos, os sensores precisam de protocolos de rede específicos para que tais recursos possam ser utilizados da melhor forma possível [Costa e Guedes 2012]. Além disso, suas aplicações requerem abordagens diferentes daquelas usadas em redes convencionais. Protocolos tradicionais como o *Transmission Control Protocol* (TCP) e o *User Datagram Protocol* (UDP), apesar de já estarem bastante estáveis e sendo usados na *Internet*, não podem ser diretamente implementados em RSSF devido às particularidades desse tipo de rede [Almalkawi et al. 2010].

Alguns dos aspectos que devem ser observados na escolha ou desenvolvimento de um protocolo de rede para RSSF são:

- *Topologia*: a topologia de uma RSSF é bastante dinâmica. Diferente de redes cabeadas ou redes Wi-Fi, os nós em uma RSSF, mesmo estando em posições fixas, podem se desligar devido ao esgotamento de energia, ou até mesmo devido a eventos do ambiente. Como as transmissões de dados geralmente funcionam com vários saltos da origem até o *sink*, sensores desligados no meio do caminho não poderão repassar os pacotes. Os protocolos de transporte e roteamento adotados em RSSF devem levar isso em consideração.
- *Vazão da rede*: vários fatores podem influenciar na vazão de uma RSSF. Um deles é o fato de haver um ou poucos *sinks* responsáveis para obter dados de dezenas ou centenas de sensores. Devido a esta topologia, a transmissão de dados ocorre de vários para um (vários sensores enviam dados para um *sink*). Os nós próximos ao *sink* precisam encaminhar pacotes vindos de diversos sensores, portanto, a necessidade de vazão nesses nós é bastante alta em comparação com os outros, o que também aumenta o consumo de energia.
- *Necessidade de tráfego em tempo-real*: algumas aplicações críticas como monitoramento de caldeiras, equipamentos em usinas, ou até mesmo aplicações onde vidas humanas estão em risco como no exemplo do *PinPtr* [Simon et al. 2004], necessitam de transmissão em tempo-real. Nessas situações, os dados devem trafegar da forma mais rápida possível até o *sink* e protocolos bastante específicos são usados para satisfazer esse requisito.
- *Qualidade dos dados obtidos*: RSSF são um caso bastante comum onde a qualidade dos dados (imagens, no caso de redes visuais) é bastante crítica. Aplicações de detecção de intrusão precisam que as imagens sendo monitoradas possuam uma qualidade mínima, para que possibilite a identificação do intruso. Os protocolos tradicionais para redes de sensores escalares não são capazes de atender a estas especificações [Costa e Guedes 2012], o que exige novas abordagens que possibilitem o tráfego de imagens e vídeos.

- *Redundância de dados*: em RSSF escalares, alguns sensores podem ser posicionados muito próximos um do outro, principalmente em implantações aleatórias. Essa situação pode prover redundância na rede e, mesmo quando um dos sensores é desligado pelo esgotamento de energia, o outro sensor continuará monitorando aquela região. Porém, se ambos transmitem o mesmo dado, poderá ser feito um uso não ótimo da rede. Alguns protocolos de transporte e roteamento consideram essa situação e são capazes de tratar a redundância de forma a utilizá-la para o bem da rede, economizando energia.
- *Consumo de energia*: ainda tratando do consumo de energia em RSSF, essa característica não pode ser deixada de lado no *design* de um protocolo de rede. Os algoritmos e abordagens envolvidas devem sempre levar em consideração a dificuldade de manter uma RSSF funcionando por bastante tempo devido à energia oriunda das baterias. Retransmissões desnecessárias, *overhead*, saltos em excesso, dentre outros, devem ser evitados neste tipo de rede.

Há muitos protocolos para diversas funcionalidades em RSSF. A aplicação da rede e os objetivos desejados é que vão definir quais protocolos deverão ser utilizados.

Na camada de transporte, os protocolos envolvidos devem ser capazes de garantir a confiabilidade da transmissão através de controles de congestionamento e recuperação de pacotes perdidos [Yick et al. 2008]. Como exemplo de protocolo de transporte, podemos citar o *Sensor Transmission Control Protocol* (STCP) [Iyer et al. 2005]. Este protocolo suporta várias aplicações em RSSF, além de prover confiabilidade e detecção de congestionamento. Para iniciar uma transmissão neste protocolo, o sensor deve enviar um pacote de inicialização de sessão para a estação base, contendo informações sobre os fluxos de dados que ele deseja transmitir. Após a confirmação da estação base, o nó de origem pode iniciar sua transmissão. A estação base calcula qual o tempo necessário para a chegada de cada pacote e monitora a transmissão constantemente. Se um pacote não chega no tempo determinado, a estação base contabiliza essa perda e calcula o nível de confiabilidade da transmissão. Caso a confiabilidade fique abaixo do nível solicitado pelo sensor de origem, a estação base envia uma confirmação negativa para o transmissor, informando-o da falha. O sensor então deverá retransmitir os dados para a estação base.

Outro exemplo desenvolvido para RSSF é o *Reliable Asynchronous Image Transport Protocol* (RAIT) [Lee e Jung 2010b]. Para prover confiabilidade na transmissão de imagens em redes de sensores multimídia, o RAIT utiliza duas janelas deslizantes por nó: uma janela, responsável pelos pacotes recebidos do sensor anterior e outra para armazenar os pacotes que serão transmitidos para o próximo salto. Quando um pacote chega ao nó, ele é temporariamente armazenado no *buffer* de recepção. Caso esteja endereçado a ele, o pacote é retirado do *buffer* e processado, caso contrário, ele deve encaminhar para o próximo salto. Isso é feito retirando o pacote do *buffer* de recepção e colocando-o no de transmissão, para, em seguida, ser encaminhado. Antes de transmitir um pacote para o próximo salto, o nó precisa solicitar *tokens*. O sensor que irá receber os pacotes verifica as suas duas filas e calcula quantos pacotes

pode receber sem haver perdas. Mesmo que haja espaço em seu *buffer* de recepção, o sensor verifica o de transmissão, para evitar que receba pacotes demais e não consiga movê-los para a fila de envio. Esse número de pacotes possíveis para recepção é enviado para o nó anterior em forma de *tokens*. Dessa forma, os sensores repassam pacotes pela rede diminuindo o congestionamento e aumentando a confiabilidade da transmissão.

Na camada de acesso ao meio (*Medium Access Control* - MAC), existem também protocolos específicos para RSSF. Eles devem cuidar da comunicação entre os nós, levando em consideração a economia de energia, sincronização entre os sensores, controle de erro, entre outros [Yick et al. 2008]. Um deles é o *sensor-MAC* (S-MAC) [Ye et al. 2004]. Neste protocolo, a fim de economizar energia, os sensores usam janelas de tempo de atividade e inatividade com tamanhos fixos. Nos momentos de atividade, os transmissores de rádio dos nós são ligados e usados para transmitir dados e pacotes de controle, enquanto que nos momentos de inatividade, o consumo de energia é reduzido ao desligar os rádios. Porém, esta abordagem requer que sensores vizinhos estejam ao mesmo instante no período de atividade. Para alcançar este objetivo, o protocolo S-MAC troca pacotes de controle constantemente, os quais os sensores utilizam para fornecer dados de suas agendas de atividade, de forma a sincronizarem. Os autores do protocolo afirmam que o S-MAC supera outras abordagens baseadas no padrão IEEE 802.11 (sem períodos de inatividade). Ainda de acordo com os autores, protocolos onde o período de inatividade é fixo reduz a vazão da rede. No S-MAC, como há um acordo entre os sensores sobre a agenda de transmissões, gerando um mecanismo dinâmico de acordo com a atividade da rede, a vazão pode ser mantida.

Outra abordagem chamada DS-MAC [Lin et al. 2004] faz uma mudança dinâmica no período de inatividade dos nós dependendo da latência dos pacotes recebidos. Quando a latência está muito alta, os nós diminuem pela metade o tempo de inatividade. Quando a latência está muito baixa, eles dobram esse tempo para economizar energia. O objetivo dessa abordagem é conseguir um balanceamento entre o consumo de energia e a latência. Neste protocolo, enquanto o período de inatividade é gerenciado (aumenta ou diminui de acordo com a latência), o período de atividade é sempre o mesmo (não muda com a latência). Ou seja, o período de tempo em que os nós estão transmitindo é fixo, mas o tempo que eles permanecem inativos pode variar. Os autores afirmam que as simulações mostraram que a latência da rede foi melhorada enquanto o consumo de energia se manteve em níveis razoáveis.

Além de protocolos de transporte e de acesso ao meio específicos, RSSFs também precisam de protocolos otimizados para o roteamento do tráfego entre os nós. Eles devem ser capazes de encontrar o melhor caminho entre um nó de origem e o de destino, considerando todas as restrições dos equipamentos envolvidos [Yick et al. 2008]. Na camada de rede, temos como exemplo o *Cross-Link Detection Protocol* (CLDP) [Kim et al. 2005]. Nesta abordagem, os autores do protocolo afirmam que a maioria das propostas para protocolos de rede assumem condições ideais sobre os rádios e a conectividade entre eles, o que causa falhas em roteamento em um cenário real.

Para desenvolver o CLDP, seus idealizadores fizeram testes para identificar os problemas das outras abordagens. O protocolo funciona analisando todos os *links* entre os nós da rede e armazenando informações sobre as conexões que se cruzam (como em um grafo, considerando as arestas como os *links*). Após essa análise, o protocolo decide então quais enlaces de rádio irá desativar, mantendo uma quantidade mínima possível de *links* cruzados. Os autores concluíram que o CLDP oferece altas taxas de dados, baixa sobrecarga e rápida convergência.

Outro exemplo é o *Anchor Location Service* (ALS) [Zhang et al. 2006]. Este protocolo dá suporte a nós móveis, com uma abordagem baseada na localização de cada sensor. O ALS constrói uma rede em forma de grade e cada nó, após obter sua posição geográfica através de algum mecanismo de posicionamento, determina em qual setor ele estará inserido. Neste caso, os nós se comunicam com outros nós das grades vizinhas. Segundo os autores, o excesso de cabeçalhos de pacotes nas comunicações da rede consome muita energia, enquanto que o ALS consegue reduzir esse excesso consideravelmente.

2.1.3 Consumo de energia

Para possibilitar o funcionamento dos sensores em ambientes distantes e isolados, baterias são utilizadas como fonte de energia. Quando a carga da bateria termina, o sensor para de funcionar e a rede perde um ponto de coleta de dados e também um nó para ajudar no encaminhamento de pacotes até o *sink*. Muitos pesquisadores desenvolveram algoritmos diversos para fazer com que os sensores economizem o máximo de energia possível, aumentando seu tempo de vida. Entre tais contribuições estão melhores protocolos de roteamento e transporte, armazenamento de dados, agendamento de tarefas, etc. [Yick et al. 2008]

Sensores que permanecem próximos ao *sink* tem uma carga de trabalho maior, pois, além de monitorar dados do ambiente, funcionam como encaminhadores de tráfego de outros nós que estão mais distante, repassando os dados oriundos deles para o *sink* [Akyildiz et al. 2002]. Por causa disso, estes sensores perdem energia mais rapidamente e tendem a se desligarem antes dos outros nós, o que acarreta em um buraco de energia (*energy hole*) em volta do *sink* [Saad et al. 2009]. Quando isto acontece, a rede fica sem conectividade com o *sink* e os sensores não conseguem mais enviar dados.

Outra forma de economizar energia, ao mesmo tempo que melhora o desempenho da rede, é a compressão e agregação de dados. Ao comprimir dados, o sensor diminui o tamanho dos pacotes que serão enviados, deixando a descompressão para o destinatário. No caso de agregação de dados, pacotes oriundos de vários sensores são combinados em um só, reduzindo o custo com cabeçalhos e sobrecarga [Yick et al. 2008].

O protocolo de compressão *Adaptive Compression-based Congestion Control Technique* (ACT) proposto em [Lee e Jung 2010a] é capaz de reduzir a quantidade de

dados transmitidos em uma RSSF através do controle de congestionamento em várias etapas. A técnica usa um conjunto de métodos de compressão para diminuir a quantidade de dados, diminuir o número de pacotes e atribuir prioridades a eles. O tamanho dos dados a serem transmitidos é então reduzido e em caso de congestionamento, pacotes com menor prioridade são descartados. De acordo com os autores, experimentos demonstraram que o ACT foi capaz de aumentar a eficiência da rede e a taxa de dados no *sink* em comparação com outros métodos.

Já para agregação de dados existe um protocolo chamado *Low Energy Adaptive Clustering Hierarchy* (LEACH) [Rajagopalan e Varshney 2006]. LEACH funciona setorizando a RSSF de forma que cada setor possua um nó líder, responsável por encaminhar os dados. A abordagem se dá em duas etapas: 1) divisão da rede em setores e eleição de um líder por cada setor; 2) agregação de dados nos líderes dos setores para envio ao *sink*. Na primeira fase, os sensores eleitos enviam uma mensagem para toda a rede informando os outros nós que eles são os líderes de setor. Os nós então, após receberem as mensagens de vários líderes, podem decidir a qual setor farão parte. Sendo assim, os sensores enviam os dados coletados para o líder do seu setor, que, então, irá fazer a agregação dos dados para enviar ao *sink*.

Além de métodos para transmissão de dados de forma eficiente, os componentes dos sensores devem ser levados em consideração. De todos eles, o transmissor de rádio é o que mais gasta energia, podendo variar uma corrente elétrica de 9 mA (em modo de recepção) a 15 mA (em modo de transmissão) [Baronti et al. 2007]. Portanto, manter os dispositivos de rádio ligados o tempo inteiro, aguardando por dados que podem não chegar, é algo que consome energia desnecessariamente. O comportamento ideal para os radiotransmissores em um sensor é que eles sejam ativados apenas quando precisarem enviar ou receber dados.

Um método bastante utilizado em RSSF para aumentar o tempo de vida dos sensores é gerenciar seus ciclos de trabalho [Yick et al. 2008]. Os nós podem ligar seus transmissores de rádio apenas quando necessitarem enviar ou receber dados, mantendo-os desligados no resto do tempo, economizando, assim, energia. Um exemplo de aplicação deste conceito é a *WiseNet* [Enz et al. 2004]. Nesta aplicação, seus desenvolvedores criaram também um protocolo MAC, chamado *WiseMAC*, com alternância de ciclos de trabalho. Como o radiotransmissor irá alternar entre estados de ligado e desligado, o protocolo de acesso ao meio utilizado deverá levar em consideração este novo comportamento. Além disso, componentes internos dos sensores são ativados e desativados em sequência, de acordo com a necessidade, economizando ainda mais energia. Segundo os autores, o uso do *WiseMAC* melhorou o consumo de energia em consideração a outros protocolos existentes.

Apesar de todos esses problemas envolvendo o consumo de energia em sensores, existem outras opções. Várias formas de captação de energia podem ser utilizadas, como painéis solares acoplados nos sensores, vibrações, ruído acústico, energia cinética, etc [Yick et al. 2008]. Porém, tais soluções vão depender do ambiente onde os sensores estão instalados. Em uma fábrica, por exemplo, é mais viável posicionar os

sensores em locais que permitam o uso da rede elétrica como fonte de energia.

Em se tratando de captação de energia, um dos métodos mais disponíveis atualmente é a utilização de painéis solares [Yick et al. 2008]. Porém, não basta acoplar um painel ao sensor, pois vários fatores influenciam na eficiência do uso de energia solar em RSSF [Raghunathan et al. 2005]. Além de estar disponível somente durante o dia, o fornecimento de energia solar pode variar. Enquanto momentos de sol forte podem prover uma alta carga de energia, quando as nuvens cobrem o céu, o fornecimento de energia cai. Outro fator importante é o tipo de bateria que é utilizado na rede. Painéis solares possuem suas características específicas, como por exemplo, tensão da energia fornecida, que deve ser compatível com a tensão de carga da bateria recarregável presente no sensor.

Há também formas de se obter energia através da movimentação dos sensores, convertendo energia cinética em eletricidade [Seah et al. 2009]. Neste método, sensores podem ser instalados em árvores, aproveitando a movimentação dos galhos, embaixo de pontes ou estradas, transformando as vibrações geradas pelas passagens dos carros em energia elétrica ou, em casos de sensores ligados ao corpo, usando o próprio movimento da pessoa que os carrega. A maioria dessas abordagens visa manter as baterias do sensor carregadas, evitando que eles parem de funcionar. Porém, existem também aplicações de captação de energia sem o uso de baterias, as quais visam manter um sensor funcionando por tempo ilimitado, desde que haja uma fonte de energia que possa ser explorada. Os autores em [Seah et al. 2009] chamam esse conceito de *Wireless Sensor Network Powered by Ambient Energy Harvesting* (WSN-HEAP).

Outro exemplo de captação de energia para manutenção de uma RSSF é a aplicação descrita em [Mateu et al. 2006]. Os autores usaram termo geradores (componentes eletrônicos que podem gerar energia a partir de um gradiente de temperatura), os quais aproveitavam a diferença entre a temperatura do corpo humano e do ambiente para alimentar um sensor ligado ao corpo. Eles desenvolveram um protótipo que media a temperatura do corpo a cada um segundo e transmitia essa informação via radiofrequência, usando apenas a alimentação fornecida pelo termo gerador, sem a necessidade de baterias.

Em resumo, energia será sempre um fator crítico em RSSF, demandando uma atenção extra no desenvolvimento de protocolos, algoritmos e mecanismos para alimentação dos nós da rede.

2.1.4 Transmissão de dados multimídia

Em redes de sensores multimídia ou redes de sensores visuais, ao invés de coletar dados escalares, os sensores precisam tirar fotos, gravar áudio ou vídeo. A disposição dos nós nesse tipo de rede pode influenciar o monitoramento, pois como esse modelo de rede de sensores utiliza câmeras para obter as imagens, surge o conceito de *campo*

de visão, que indica qual a área o sensor consegue monitorar de acordo com a posição de sua câmera. Um exemplo de RSMSF pode ser visto na Figura 2.3.

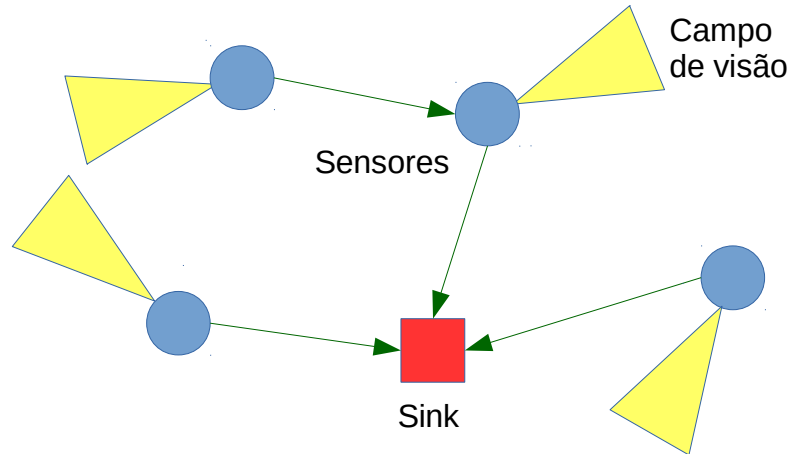


Figura 2.3: Exemplo de uma RSMSF.

Por terem recursos limitados, os sensores também sofrem de outros problemas, como alcance restrito (o que os impossibilitam de transmitir dados para longas distâncias), baixa vazão em transmissão de dados devido a limitações de consumo de energia, pouca capacidade de armazenamento e processamento e problemas de conectividade devido a obstruções [Yick et al. 2008]. Quando dados multimídia precisam ser enviados, diversos novos desafios devem ser tratados.

Diversos protocolos de transporte, roteamento e enlace foram desenvolvidos para tentar solucionar os problemas gerados pelo alto tráfego de dados em redes multimídia [Almalkawi et al. 2010]. A ideia por trás desses protocolos é viabilizar o transporte de grandes quantidades de dados dentro das limitações das redes sem fio entre os sensores. Mesmo com essas abordagens, o consumo de energia é elevado devido à natureza da transmissão salto a salto das redes de sensores, que agora possui mais dados para transmitir.

Devido a esta quantidade de dados que precisa ser transmitida na rede, a codificação de dados é um fator importante nas RSMSF. Quanto maior for a compressão dos dados multimídia, menos detalhes da imagem serão transmitidos, porém, mais eficiente será o tráfego na rede. Além de reduzir o tamanho dos pacotes, as codificações permitem a recuperação de dados perdidos no receptor, evitando retransmissões.

As abordagens de múltiplos *sinks* e de *sinks* móveis podem ajudar a solucionar os problemas de transmissão de dados em tempo real nas redes multimídia [Costa et al. 2014]. Ao posicionar os *sinks* de maneira otimizada, os sensores podem transmitir as imagens diretamente, sem a necessidade de encaminhamentos através de outros nós. Dessa forma, outros sensores não precisam consumir suas

baterias e a transmissão direto da origem de dados pode ser feita de forma mais confiável e rápida.

2.1.5 Mobilidade de *sinks*

Devido às restrições impostas pela limitação de energia das RSSF, técnicas devem ser utilizadas para reduzir o consumo energético dos sensores. O uso da mobilidade de *sinks* é uma das abordagens que pode ser utilizada.

Em RSSF com *sinks* estáticos, sensores em volta dos *sinks* trabalham encaminhando pacotes oriundos de outros sensores mais distantes. Ou seja, além de fazer o sensoriamento do ambiente, precisam também funcionar como conexões para outros nós. Dessa forma, a energia desses sensores é consumida mais rápida e eles podem deixar de funcionar, fazendo-se perder a conexão com os outros nós que dependiam deles para enviar dados para o *sink* (*energy hole*) [Peixoto e Costa 2015].

Para evitar os buracos de energia, os *sinks* devem se movimentar pela rede. Ao fazer isso, o consumo devido ao encaminhamento de pacotes é balanceado com os outros sensores. Porém, ao movimentar um *sink* e aproximá-lo de alguns nós, ele acaba se distanciando de outros. Dessa forma, o posicionamento deve ser feito de uma maneira planejada para evitar que uma movimentação ruim traga malefícios à rede [Khan et al. 2013].

O padrão de mobilidade dos *sinks* em uma RSSF pode ser dividido em três tipos: fixo, aleatório e controlado. No tipo fixo, o *sink* se movimenta por um percurso predeterminado durante todo o tempo. Esse percurso não é definido por nenhum parâmetro da rede e também não é influenciado pelas posições dos sensores. Como exemplo de uma movimentação fixa, considere um *sink* que permanece se movimentando em uma linha reta pela RSSF, indo e voltando pelo mesmo caminho. No tipo aleatório, o *sink* se movimenta de uma forma aleatória pela área da RSSF. Assim como no tipo fixo, parâmetros da rede e posições dos sensores não influenciam o percurso do *sink*. Enquanto que no tipo controlado, o *sink* utiliza algum parâmetro da rede para decidir para onde se mover. As posições dos sensores, combinadas com os parâmetros considerados na abordagem de mobilidade controlada, influenciam as novas posições do *sink* [Khan et al. 2013].

O trabalho em [Gao et al. 2009] é um exemplo de abordagem de movimentação de *sink* usando um padrão de mobilidade fixo. Nessa proposta, é definido um caminho fixo para o *sink*, no qual ele se movimenta periodicamente a uma velocidade constante enquanto os sensores são espalhados na vizinhança desse caminho. O *sink* coleta dados dos sensores na sua proximidade, que são chamados nessa abordagem de *sub-sinks*. Esses sensores formam a *Direct Communication Area* (DCA) e apenas eles se comunicam com o *sink* móvel. Sensores mais distantes formam o que os autores chamam de *Multi-hop Communication Area* (MCA). Os sensores na MCA não se comunicam com o *sink* móvel, ao invés disso, eles se comunicam com os *sub-sinks* na DCA, os quais são responsáveis em encaminhar os pacotes ao *sink* móvel.

Uma abordagem com padrão de mobilidade aleatório é apresentado em [Chatzigiannakis et al. 2006]. Os autores comparam quatro tipos diferentes de mobilidades de *sinks*, sendo duas delas de padrão aleatório. Na primeira abordagem, o *sink* se move aleatoriamente pela rede, enviando *beacons* para alertar os sensores de que está por perto. Os sensores que recebem os *beacons* se conectam com o *sink* e enviam os dados de sensoriamento que estavam armazenados. Dessa forma, os autores garantem que toda a rede é visitada, inclusive áreas desconectadas. Na segunda abordagem, os autores definem regiões de visitas para os *sinks* móveis e eles se movimentam por essas regiões de forma aleatória, coletando os dados dos sensores. O trabalho de [Chatzigiannakis et al. 2006] apresenta gráficos onde é possível perceber que, apesar de conseguir se comunicar com todos os sensores, os *sinks* móveis com padrão aleatório aumentam significativamente a latência da rede, tornando seu uso inviável em aplicações onde a latência é crucial.

Já em [Friedmann e Boukhatem 2007] podemos encontrar um exemplo de padrão de mobilidade controlado. Nesse trabalho, os autores propõem um método para movimentar múltiplos *sinks* considerando a energia dos nós que farão o encaminhamento dos pacotes. Para cada conexão entre os sensores é atribuído um peso que considera a energia do nó de destino e a distância. O algoritmo verifica os pesos para todas as rotas possíveis, considerando todos os *sinks* da rede. Para obter as melhores posições para os *sinks*, diversas configurações de posicionamento são testadas e comparadas com o estado atual. Apenas as que beneficiam o consumo de energia são consideradas. Após esta etapa, a melhor configuração é aquela que leva a ter um menor consumo de energia nas transmissões de múltiplos saltos e que também traz o menor custo para a movimentação dos *sinks*.

O artigo em [Costa et al. 2014] propôs uma abordagem para o posicionamento de um único *sink* em uma RSSF baseada nas relevâncias dos sensores da rede. Nesta abordagem, cada nó possui um nível de relevância que pode variar de 0 (nada relevante) a 15 (totalmente relevante). De acordo com os níveis de cada sensor, o algoritmo calcula a melhor posição para o *sink*. Porém, o método de Costa não leva em consideração as RSSFs com vários *sinks*. Dessa forma, este trabalho propõe usar o algoritmo de posicionamento baseado em relevância para achar a posição ótima dos múltiplos *sinks* em uma rede de sensores.

2.2 Múltiplos sinks

Abordagens que envolvem múltiplos *sinks* tem como objetivo aumentar o tempo de vida da rede evitando os buracos de energia [Marta e Cardei 2009]. Com múltiplos *sinks*, a carga de trabalho para o recolhimento de dados é distribuída. Ao invés de usar um único ponto central onde todos os pacotes convergem, consumindo energia dos sensores em volta, vários *sinks* são espalhados na rede. Dessa forma, os sensores em volta de cada *sink* recebem menos dados para encaminhar. Se colocados nas

posições corretas, múltiplos *sinks* podem diminuir o consumo de energia da rede de forma considerável [Friedmann e Boukhatem 2007]. A Figura 2.4 apresenta exemplos de RSSF com um, dois e quatro *sinks* espalhados em forma de grade, onde cada sensor é associado ao *sink* mais próximo.

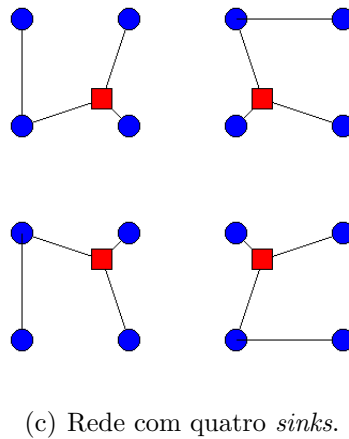
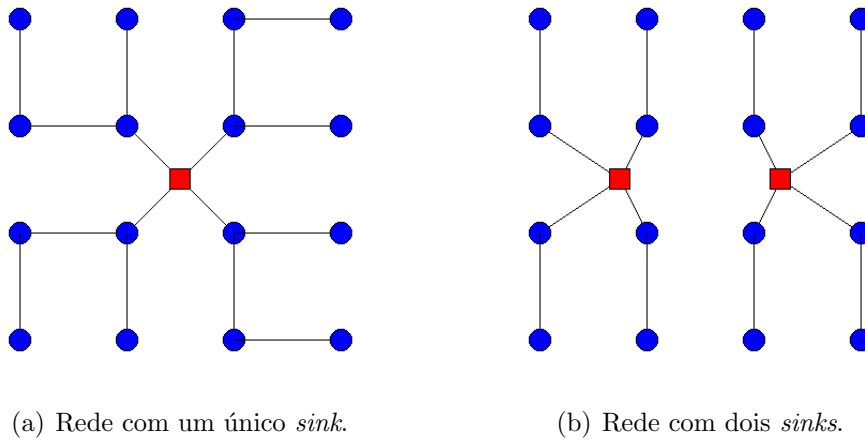


Figura 2.4: Exemplo de disposição de múltiplos *sinks* em uma RSSF.

Da mesma forma, ao usar *sinks* móveis, os sensores acabam revezando na tarefa de encaminhar os dados para recolhimento. Pois enquanto um determinado sensor está muito próximo de um *sink* em um determinado momento, após o movimento deste *sink*, o sensor que estava próximo a ele já poderá estar mais distante, não sendo mais o responsável por encaminhar dados para o nó coletor [Koç e Korpeoglu 2014].

Os autores em [Das e Dutta 2005] criaram um modelo de simulador para adaptar e testar o uso de protocolos e algoritmos de roteamento para redes com um único *sink* em ambientes com vários *sinks*. Primeiro eles modelaram um grafo G representando a RSSF, onde cada sensor e cada *sink* era um nó do grafo e cada conexão da rede era representada por uma aresta. Em seguida, era gerado um grafo G' , onde os nós

que representavam os *sinks* eram removidos, além das arestas que se conectavam a eles. Esse nós eram então substituídos por um nó s' que representava todos os *sinks* em um único *sink* virtual e cada aresta que conectava um sensor a um *sink* no grafo G era então inserida no grafo G' , conectando os nós ao *sink* virtual. Com isso, eles tinham um grafo contendo todos os sensores da rede e suas conexões com seus *sinks*, utilizando apenas um único nó virtual. Esse grafo G' era então utilizado para executar os algoritmos dos protocolos que foram desenvolvidos para redes com um único *sink*, possibilitando aos autores simularem o comportamento desses algoritmos em redes com múltiplos *sinks*. Após as simulações, foi possível concluir que a economia de energia em uma região de d -dimensões com a inserção aleatória de k *sinks* é proporcional a $k^{\frac{1}{d}}$.

Outro trabalho envolvendo múltiplos *sinks* é descrito em [Meng et al. 2007]. Nesta abordagem, os autores demonstram que ao usar mais de um *sink* é possível aumentar o tempo de vida da rede. Eles desenvolveram dois algoritmos de roteamento para RSSF com múltiplos *sinks* considerando os níveis de energia das rotas e o balanceamento do consumo na rede. O *Energy Level Based Routing Algorithm* (ELBR) considera os níveis de energia em cada rota possível quando um sensor precisa enviar dados para o *sink*. Neste caso, antes de enviar um pacote, o sensor mede os níveis de energia de todas as rotas possíveis e escolhe aquela com o maior nível. Dessa forma, rotas com menos energia são evitadas para preservar o funcionamento dos sensores que fazem parte dela. Já o *Primary Based Routing Algorithm* (PBR) considera também o custo de cada caminho possível. Antes de iniciar uma transmissão, o sensor verifica os níveis de energia das rotas e também o custo de cada uma, preferindo as que tiverem o menor custo. Dessa forma, é possível manter o consumo de energia na RSSF de forma balanceada, prolongando o tempo de vida da rede. De acordo com os autores, o uso desses algoritmos permitem enviar mais dados na rede em comparação com uma configuração com apenas um único *sink*.

Uma abordagem usando um *sink* móvel é proposta em [Saad et al. 2009]. Neste trabalho, os autores propuseram uma estratégia para planejar um traçado dentro da RSSF pelo qual o *sink* deve se movimentar. Nesta proposta, a rede é dividida em setores onde um nó principal ficará responsável por armazenar os dados medidos de cada sensor do seu setor. O *sink* então irá se mover de forma a passar próximo a cada setor, coletando os dados armazenados pelos nós principais até atravessar toda a rede e retornar ao ponto de partida. Com essa abordagem, os nós principais de cada setor podem transmitir os dados diretamente para o *sink*, minimizando o consumo de energia devido a múltiplos saltos. A estratégia de movimentação proposta é capaz também de evitar obstáculos, sejam eles fixos ou móveis. Segundo os autores, essa abordagem supera diversos outros métodos de recolhimento de dados, incluindo abordagens com *sinks* móveis.

Os autores em [Marta e Cardei 2009] propuseram um novo algoritmo de posicionamento para *sinks* móveis. Eles simularam uma RSSF dividida em setores em forma de um hexágono, na qual os *sinks* se movimentavam nas bordas dos setores. Eles mudavam de local dependendo da energia residual dos sensores dentro dos setores

da rede. Para decidirem onde ir, os sinks buscavam por zonas com maior quantidade de energia residual. Como os sensores mais próximos aos sinks tem um maior consumo, já que eles precisam encaminhar pacotes de outros nós, eles geralmente tem um tempo de vida menor comparado com os outros sensores. Então, os sinks se movem para áreas com maior energia, balanceando o consumo. Os autores afirmaram que sua abordagem aumentou o tempo de vida da rede em 4,86 vezes, comparado com um cenário de sink estático.

2.3 Otimização com QoE

Dentre todas as possibilidades de otimização de uma RSSF, o uso de priorização pode ser bastante eficaz quando há uma necessidade de diferenciar os sensores da rede [Costa et al. 2015]. Em certas aplicações, um sensor pode ser mais importante que outro, de forma que esse nó mais importante seja priorizado para que seus dados cheguem mais rapidamente ao *sink* e que seu consumo de energia seja minimizado. Ao priorizar sensores da rede de acordo com a importância que cada um tem para a aplicação, conseqüentemente estaremos dando uma melhor qualidade de experiência (*Quality of Experience* - QoE).

Para atingir esse objetivo, as características dos nós e da aplicação devem ser consideradas e utilizadas como parâmetro de priorização. Por exemplo, suponha uma RSSF que possui sensores que monitoram dados escalares como temperatura e luminosidade, além de dados multimídia como fotos. Se as fotos forem mais importantes para a aplicação do que os dados escalares, os sensores que capturam fotos do ambiente, além daqueles que estão encaminhando pacotes de dados contendo fotos, devem ser priorizados. Nesse exemplo, o parâmetro de priorização foi a capacidade de um sensor em enviar fotos. Após definir um parâmetro, a forma pela qual ele será explorado para definir as prioridades deve ser especificada. Tomando o mesmo exemplo, podemos dizer que sensores que contém pacotes de dados em seu *buffer* de transmissão devem ser priorizados, enquanto que aqueles que não possuem tais pacotes não devem ser priorizados [Costa et al. 2015].

As priorizações utilizadas nesse tipo de otimização podem ainda ser classificadas como local ou global [Costa et al. 2015]. Priorizações locais são aquelas que afetam uma transmissão específica de um sensor específico. Por exemplo, um sensor tem dados escalares e multimídia armazenados em seu *buffer* de transmissão, porém, como a aplicação requer que os dados multimídia sejam transmitidos o mais brevemente possível, o sensor os prioriza em sua transmissão. Já as priorizações globais são aquelas que afetam toda a rede. Tomando a mesma aplicação do exemplo anterior, se existe um sensor com dados escalares para serem transmitidos e um outro com dados multimídia, o segundo sensor terá prioridade sobre o primeiro no tráfego da rede.

Um exemplo de priorização local pode ser visto em [Zhang et al. 2008]. Nesse tra-

balho, os autores propuseram um esquema de múltiplas rotas para pacotes de dados chamado *Multi-priority Multi-path Selection* (MPMPS). Em uma RSMSF, pacotes de vídeo eram separados em imagem e áudio. Dependendo da aplicação a imagem pode ser mais importante (em um monitoramento de incêndio, por exemplo) ou o áudio pode ter maior relevância (exploração do fundo do mar). O fluxo mais importante para a aplicação, seja ele imagem ou áudio, é transportado pelos melhores caminhos possíveis entre a origem e o destino, enquanto que o fluxo menos importante é transportado por outros caminhos.

Em [Costa et al. 2013], os autores propuseram uma abordagem de otimização com priorização global. Nesse trabalho, nós munidos de câmeras podem ter suas relevâncias alteradas por eventos ocorridos na área de monitoramento. Sensores escalares são usados para monitorar atividades diversas no ambiente, que geram eventos, provocando uma mudança de comportamento na rede. Sensores multimídia na mesma célula onde ocorreu o evento podem ter suas prioridades aumentadas para que o comportamento da rede mude, a fim de priorizar o tráfego vindo dessas câmeras. Por exemplo, se os sensores escalares detectam um incêndio em um determinado local, os sensores multimídia que podem visualizar o incêndio são priorizados pela rede para que suas imagens cheguem mais rapidamente à aplicação.

Capítulo 3

Metodologia

O desenvolvimento desse trabalho foi fundamentado nos conceitos de RSSF, mais especificamente em mobilidade de múltiplos *sinks* e RSSF baseadas em relevância. Diversas abordagens e algoritmos que tratam sobre esse assunto foram estudados de forma a descobrir e compreender as soluções já existentes.

Inicialmente, a abordagem de posicionamento de um único *sink* baseado em relevância desenvolvido em [Costa et al. 2014] serviu como ponto de partida para este trabalho. O algoritmo já existente resolve o problema de posicionamento de um *sink*, considerando os níveis de relevância dos sensores na rede, porém, é limitado para RSSF com um único *sink*.

De forma a posicionar vários *sinks* usando a abordagem de [Costa et al. 2014], pensou-se em setorizar a RSSF, dividindo os sensores em grupos e atribuindo um *sink* para cada grupo. Ao fazer isso, foi possível posicionar todos os *sinks*, usando os níveis de relevância dos sensores em seu grupo.

A próxima etapa consistiu em simular o posicionamento de múltiplos *sinks* após setorizar a rede. Durante este processo, um simulador para *sinks* móveis em RSSF foi desenvolvido, de forma a possibilitar testar o algoritmo desenvolvido e visualizar a movimentação dos *sinks* no percorrer do tempo da simulação. Neste simulador, testes foram feitos com os métodos de setorização mais comuns em RSSF (*k-means* com *Forgy* e *Random Partition*) [Abdul Latiff et al. 2007], o que demonstrou ser ineficiente quando os níveis de relevância dos nós importam para a aplicação. Diante disto, foi necessário desenvolver um novo método de inicialização para o algoritmo *k-means*, baseado em relevâncias.

Por último, após a elaboração do método de setorização baseado em relevâncias, o simulador foi utilizado para testar o algoritmo e verificar sua eficácia. Diversos ajustes foram feitos até chegar ao resultado apresentado neste trabalho.

3.1 Formalização do problema

O problema a ser solucionado neste trabalho é como posicionar múltiplos *sinks* em uma RSSF, de forma a considerar os níveis de relevância dos sensores presentes na rede, priorizando os mais relevantes. De forma a entender o problema, o conceito de “relevância” deve ser descrito.

Em uma RSSF baseada em relevâncias, cada sensor possui um nível de importância para a aplicação. Por exemplo: em uma aplicação de detecção de intrusão, sensores posicionados em locais mais críticos do ambiente monitorado são mais importantes para a aplicação e devem ser priorizados. A relevância que cada sensor possui é determinada pela aplicação ou pelo usuário que está utilizando a RSSF (podendo ser fixa ou variável), não é definida pelo método proposto neste trabalho. Ou seja, o algoritmo desenvolvido neste trabalho utiliza os níveis de relevância que já devem estar previamente definidos pela aplicação, pelo usuário ou por algum outro agente externo, protocolo, etc.

Uma forma de priorizar esses sensores em uma RSSF com vários *sinks* é posicionar os *sinks* mais próximos desses sensores importantes. Ao fazer isso, é possível economizar energia na rede para trafegar dados mais relevantes, pois haverá menos saltos entre o sensor importante e o *sink*, o que evita o consumo com encaminhamento de pacotes. Além disso, ao reduzir o número de saltos, é possível diminuir a latência de comunicação do sensor importante [Aoun e Boutaba 2006].

Para avaliar a abordagem, consideramos o consumo de energia de todos os nós da rede. Nas simulações realizadas, sempre que um sensor transmitia ou recebia um pacote, a energia utilizada para esta tarefa era contabilizada e, no final da simulação, a energia gasta por todos os sensores era somada e com isso obtém-se o consumo total da rede. Como o objetivo do posicionamento controlado dos *sinks*, neste trabalho, é reduzir o consumo de energia, as simulações permitiam que fosse possível avaliar se o algoritmo proposto é superior ao uso de *sinks* estáticos ou ao posicionamento sem o uso de relevâncias.

Em um cenário real, diversos outros fatores influenciam no consumo de energia, como a perda de pacotes, retransmissões, interferências, falhas de *hardware*, protocolo de comunicação, etc. Porém, como pretende-se comparar diferentes métodos de posicionamento, fatores externos foram descartados e apenas o consumo por transmissão e recepção de pacotes foi considerado.

3.2 Procedimentos

Todo o processo de desenvolvimento do trabalho iniciou com a elaboração do algoritmo de setorização. A literatura disponibiliza diversos trabalhos sobre setorização em RSSF, assim como sua categorização. Ao pesquisar outras abordagens, foi descoberto que cada algoritmo de setorização tem um objetivo específico, o que levou à

construção de um método específico para RSSF com relevância. Para o desenvolvimento do algoritmo, os sensores e os *sinks* são modelados como pontos em um plano cartesiano. Suas coordenadas são utilizadas para definir suas posições e, a partir delas e dos níveis de relevância, a posição final do *sink* é calculada dentro daquele setor.

Para contemplar restrições de posicionamento na rede, foi considerada a possibilidade de existência de obstáculos e caminhos específicos (vias) para movimentação dos *sinks*. Os obstáculos impedem que um *sink* se posicione em determinada área, portanto, o algoritmo deve corrigir a posição de um *sink* caso ele ocupe a área de um obstáculo. De forma a simplificar o desenvolvimento, os obstáculos foram modelados como circunferências e equações de geometria analítica puderam ser utilizadas para a correção da posição do *sink*. Neste trabalho, não foram consideradas perdas devido à propagação dos sinais através dos obstáculos.

Na situação de caminhos para movimentação, foram utilizados segmentos de reta para a modelagem. As coordenadas de posição de um *sink* obrigatoriamente devem estar dentro de um segmento de reta. Também foram considerados os raios de alcance dos sensores, obrigando o *sink* a permanecer em uma zona onde fosse possível se conectar a um sensor. Mais uma vez, equações de geometria analítica resolveram essa etapa do desenvolvimento do algoritmo.

Em resumo, o algoritmo completo faz com que cada *sink* tenha uma posição calculada a partir dos níveis de relevância dos sensores em seu setor, evitando obstáculos (quando houver), obedecendo limites de posicionamento em vias (quando houver) e assegurando que esteja dentro do raio de alcance de algum sensor.

3.3 Ferramentas

Para avaliar o algoritmo proposto, um simulador foi desenvolvido usando a linguagem C++. Neste simulador, os procedimentos de setorização, posicionamento, correção e cálculo de rotas foram codificados de forma a simular o consumo de energia da rede e possibilitar a visualização de cada movimentação dos *sinks* durante todo o tempo da simulação. Também foram codificados o comportamento da rede com *sinks* estáticos, com movimentos fixos (horizontal e vertical) e com setorização utilizando os métodos de inicialização *Forgy* e *Random Partition*.

O simulador permite que o comportamento dos sensores sejam programados através de um arquivo XML (*Extensible Markup Language*) que contém a configuração de toda a rede. Neste arquivo, é possível especificar o tamanho da rede, o raio de alcance dos transmissores, as posições e tamanhos dos obstáculos e vias, além das posições e níveis de relevância dos sensores. Para cada sensor, é possível também especificar mudanças de comportamento por tempo. Por exemplo, através deste arquivo, é possível dizer que após x segundos, o sensor vai mudar seu nível de relevância, ligar

ou desligar. Não há limites para o número de mudanças que pode ser programadas para cada sensor.

Sempre que há uma mudança na configuração dos sensores, o simulador recalcula as posições dos *sinks*, obedecendo os obstáculos e restrições, quando houver. Após definir a nova posição, a rota utilizada para movimentar o *sink* da posição anterior até a posição recalculada é obtida através do algoritmo de menor caminho de Dijkstra [Dijkstra 1959] e o caminho é destacado na interface gráfica, possibilitando uma visualização em tempo de execução das rotas percorridas pelos *sinks*.

Após as simulações, é possível gerar um arquivo CSV (*Comma-separated Values*) contendo o consumo de energia e o número de sensores ativos a cada segundo da simulação. Outros valores contidos nesse arquivo são: o total de energia consumida, os momentos onde ocorreram mudanças nas posições dos *sinks* e suas rotas (caminhos pelos quais se movimentaram), número de pacotes recebidos pelos *sinks*, dentre outros.

Os dados contidos no arquivo CSV de saída foram utilizados para gerar os gráficos apresentados neste trabalho e analisar os resultados obtidos com a abordagem proposta.

Capítulo 4

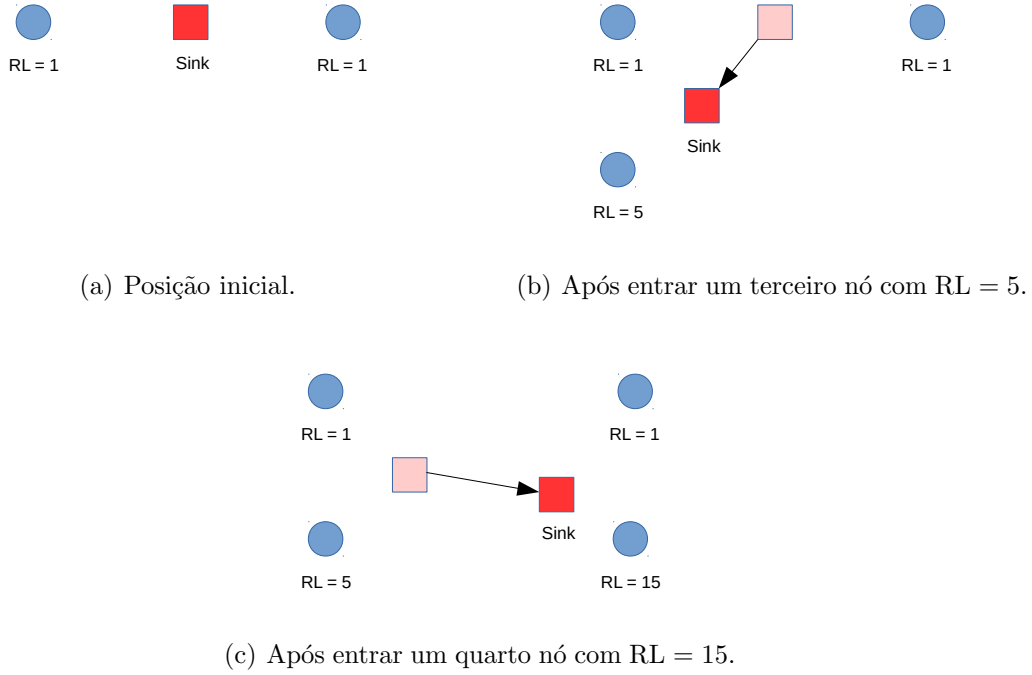
Posicionamento de *sinks* usando priorização

Este capítulo apresenta a abordagem proposta, descrevendo seu funcionamento, modelo matemático utilizado e o algoritmo. São descritos também métodos de posicionamento de um único *sink* com o uso de relevâncias, taxonomias de algoritmos de setorização e o funcionamento de uma abordagem desenvolvida para a setorização dos sensores considerando seus níveis de relevância.

4.1 Posicionamento de um único *sink*

O posicionamento do *sink* é feito de forma a considerar apenas os níveis de relevância (RL) dos sensores da RSSF, mantendo-o mais próximo daqueles nós mais relevantes. Se existem vários sensores com níveis de RL diferentes e em posições espalhadas, o *sink* irá considerar a importância de cada um relacionada com a média dos níveis de relevância, fazendo assim uma abordagem de priorização global, permitindo que os nós mais importantes sejam priorizados. Neste cálculo, o raio de alcance dos transmissores não são considerados, o que pode permitir que sensores menos importantes fiquem fora da rede.

A Figura 4.1 apresenta o comportamento de um *sink* ao considerar os níveis de relevância dos sensores em uma RSSF. Na Figura 4.1(a), dois sensores com $RL = 1$ estão na rede e o *sink* se posiciona entre os dois. Como possuem a mesma prioridade, o *sink* não os distingue quanto à importância. Na Figura 4.1(b), um terceiro sensor com $RL = 5$ é inserido na rede. Por possuir relevância maior que os dois nós iniciais, o *sink* se posiciona mais próximo dele, de forma proporcional à sua relevância, comparada com as relevâncias dos outros sensores, priorizando seu tráfego. Por último, na Figura 4.1(c), um quarto sensor com $RL = 15$ entra na rede, novamente, fazendo com que o *sink* se movimente.


 Figura 4.1: Movimentação do *sink* com uso de relevâncias.

Para obter esse comportamento, as coordenadas (x, y) de um *sink* k são calculadas a cada vez que a topologia da rede se altera. Nesse cálculo, são consideradas as coordenadas de cada sensor, multiplicadas pelo seu nível de RL. Esse valor é então dividido pela soma das relevâncias de todos os nós da rede. As distâncias entre os sensores, junto com seus níveis de relevância, vão influenciar na nova posição do *sink*.

No modelo matemático utilizado para o cálculo da posição do *sink*, é considerado que a RSSF é formada por um conjunto $S = \{s_0, s_1, \dots, s_n\}$ de sensores, onde cada um possui uma coordenada no eixo horizontal denominada $x(s_i)$ e outra no eixo vertical denominada $y(s_i)$, além de seu nível de relevância, definido como $RL(s_i)$. As Equações 4.1 e 4.2 definem o cálculo das coordenadas (x, y) de um *sink* k em uma RSSF baseada em relevância [Costa et al. 2014].

$$x(k) = \left[\frac{\sum_{i=0}^n (x(s_i) \cdot RL(s_i))}{\sum_{i=0}^n RL(s_i)} \right] \quad (4.1)$$

$$y(k) = \left[\frac{\sum_{i=0}^n (y(s_i) \cdot RL(s_i))}{\sum_{i=0}^n RL(s_i)} \right] \quad (4.2)$$

Após definidas as novas coordenadas do *sink*, ele deve ser movido para a nova posição e lá permanecer até uma nova mudança de topologia (um novo sensor entra na rede, um sensor já existente deixa de funcionar, etc) ou caso o padrão de transmissão da rede se altere (sensores ociosos começam a transmitir, um sensor ativo se torna ocioso, os níveis de RL são alterados, etc). O mecanismo de movimentação física do *sink* pela RSSF está fora do escopo deste trabalho.

4.2 Setorização de RSSF

A abordagem de posicionamento descrita na Seção 4.1 falha em uma RSSF com múltiplos *sinks*. Se o cálculo das coordenadas forem repetidos para todos os *sinks* da rede, eles ficarão posicionados no mesmo local.

Para alcançar uma abordagem baseada em relevância que funcionasse com vários *sinks*, foi proposta a metodologia de setorização das RSSF de acordo com o número de *sinks*, onde cada um é atribuído para cada setor. Esse esquema traz um novo desafio: como dividir a rede em setores considerando os níveis de relevância dos sensores?

Em [Abbasi e Younis 2007], vários algoritmos de setorização podem ser encontrados. Cada um possui um objetivo específico e atua em um tipo de RSSF. De acordo com os autores, os tipos de abordagens de setorização podem ser classificados quanto às suas características e apresentam uma taxonomia que os divide por tipo de rede, objetivos e atributos.

Quanto ao tipo de rede, os algoritmos são categorizados pela RSSF a qual eles atuam, considerando suas peculiaridades e restrições. São definidos da seguinte forma:

- *Dinâmica da rede.* O comportamento dos sensores da RSSF influenciam no tipo de algoritmo de setorização a ser utilizado. Sensores podem ser móveis ou estáticos e seu comportamento pode ser constante ou variável. Em caso de sensores móveis, a abordagem de setorização será bastante complexa, já que ao se movimentarem, os sensores podem trocar de setor. Quanto ao tráfego da rede, em determinadas aplicações, os nós da rede apenas transmitem dados quando há algum evento (comportamento variável), fazendo com que a rede fique ociosa por um tempo. Em outros casos, os sensores podem enviar dados continuamente (comportamento constante).

- *Processamento de dados na rede.* Alguns pacotes de dados podem ser gerados de forma redundante, onde mais de um sensor monitora o mesmo alvo ou quando um único nó replica o mesmo pacote por mais de uma rota. Neste tipo de rede, os pacotes redundantes podem ser agregados para garantir a recepção do dado, ou podem ser descartados para diminuir o tráfego na rede. O mecanismo usado para dividir a rede em setores deve considerar essa característica para evitar que pacotes desnecessários cheguem à aplicação ou que sejam descartados de forma incorreta, provocando a perda de dados.
- *Disposição e capacidade dos sensores.* Os nós de uma RSSF podem ser implantados de forma planejada ou aleatória. Em redes com disposição planejada de sensores, os setores também podem ser pré-determinados, dispensando a necessidade de algoritmos de setorização, ou em alguns casos, não há a necessidade de setorização. Porém, em implantações aleatórias, o uso de um algoritmo de setorização se faz necessário. Além disso, os sensores podem ter papéis diferentes dentro da rede. Em algumas aplicações, um nó pode ser usado para monitoramento de dados escalares, disparando eventos, e outros para coleta de dados multimídia. Sensores podem funcionar de forma ativa, sensoreando o ambiente, enquanto outros atuam apenas como intermediários em uma comunicação de múltiplos saltos. Nestes casos, os setores devem ser definidos de forma a manter a heterogeneidade da rede.

Os objetivos da aplicação da RSSF também definem o algoritmo de setorização a ser utilizado. Novamente, os autores em [Abbasi e Younis 2007] definem uma taxonomia dos mecanismos de setorização quanto ao objetivo da rede.

- *Balanceamento de carga.* Dependendo da aplicação, pode ser necessário o balanceamento de carga entre os sensores e os *sinks*. Neste caso, um algoritmo de setorização que consiga dividir a rede em partes iguais pode beneficiar a aplicação [Younis et al. 2003]. Agindo de forma correta, os sensores podem se comunicar de forma mais homogênea, equalizando o consumo de energia e a latência do tráfego de dados na rede. Esse cenário pode evitar que uma parte da rede se desligue precocemente devido ao esgotamento de energia.
- *Tolerância a falhas.* Em aplicações onde os sensores são instalados em ambientes hostis, muitos deles podem falhar permanentemente durante o tempo de vida da rede. Nesses casos, os setores vão perdendo seus membros e a partir de um certo ponto, a divisão dos nós pode não fazer mais sentido para a aplicação. Alguns algoritmos propõem refazer a setorização, o que pode ser custoso para os elementos da rede. Outros elaboram mecanismos de tolerância a falhas, adaptando os setores à nova configuração da rede [Heinzelman et al. 2002].
- *Conectividade aumentada e latência reduzida.* A conectividade entre os sensores é muito importante em uma RSSF. Existem aplicações onde os setores devem se comunicar, o que geralmente é feito através de um sensor principal chamado de *cluster head* (CH). Nessas situações, o algoritmo de setorização

deve considerar a conectividade entre os CH, a fim de manter os setores interligados. Em outros casos, a comunicação entre os sensores de um setor deve ser otimizada, o que requer algoritmos de setorização capazes de manter os nós a poucos saltos do seu destino, o que também reduz a latência. Em determinadas abordagens, isso também inclui limitar o número de saltos dentro de um setor [Dai e Wu 2005].

- *Número mínimo de setores.* Esse tipo de abordagem tenta minimizar a quantidade de setores em uma RSSF. Isso geralmente é necessário quando há a necessidade do uso de *cluster heads* caros ou proibitivos. Dependendo da capacidade de um CH, ele pode ser muito custoso, ou até mesmo, muito grande, penalizando sua instalação. Além disso, algumas aplicações de RSSF requerem que os elementos da rede fiquem ocultos (por exemplo, no monitoramento de um campo de batalha), o que torna o uso de muitos CH visíveis prejudicial ao objetivo da rede.
- *Máxima longevidade da rede.* Como a restrição de energia dos sensores é um problema inerente em uma RSSF, seu uso deve ser otimizado ao máximo. Em casos onde o objetivo da setorização é maximizar o tempo de vida da rede, a divisão dos sensores deve ser feita de forma a reduzir o consumo desnecessário de energia devido a transmissões através de muitos saltos. Nessas situações, o uso de *cluster heads* próximos aos sensores é eficaz e pode trazer benefícios. Caso os *cluster heads* sejam escolhidos a partir de sensores instalados na rede, sugere-se que eles não sejam utilizados para monitoramento a fim de economizar o uso de energia.

Por último, a taxonomia apresentada em [Abbasi e Younis 2007] define os atributos dos setores.

- *Propriedades dos setores.* Os algoritmos de setorização tem suas propriedades de acordo com o esquema escolhido. Algumas são definidas como *contagem de setores* (o número de setores da rede pode ser pré-determinado pela aplicação ou variável), *estabilidade* (adaptativa, quando o número de setores muda durante o tempo de vida da rede ou fixa, quando os elementos sempre executam o mesmo papel e o número de setores permanece inalterado), *topologia intra-setor* (comunicação direta dos sensores com o CH ou comunicação de múltiplos saltos) e *conectividade entre setores* (quando os *cluster heads* não atuam como *sinks*, eles precisam fazer uma comunicação de múltiplos saltos entre eles para chegar ao destino).
- *Capacidades do cluster head.* As necessidades da aplicação e o tipo de RSSF utilizada podem determinar as capacidades exigidas do *cluster head*, o que inclui *mobilidade* (um CH pode ser móvel, alterando dinamicamente o padrão de setorização da rede, como um *sink* móvel, ou pode ser fixo), *tipo* (um CH pode ser um sensor da rede ou um nó especializado) e *papel* (um CH pode atuar apenas como intermediário em uma comunicação de múltiplos saltos para o

destino, pode fazer agregação de dados como também pode agir como um *sink* da rede).

- *Processo de setorização.* A forma como a setorização é feita por cada abordagem varia de acordo com a finalidade, tipo de RSSF, etc. São definidos atributos do processo de setorização quanto à *metodologia* (pode ser feita de forma distribuída entre os sensores, de forma centralizada pela aplicação ou de forma híbrida), *objetivo* (tolerância a falhas, longevidade da rede, etc.), *escolha do cluster head* (pode ser pré-determinado ou selecionado dinamicamente a partir dos sensores da rede) e *complexidade* (pode ser constante ou depender do número de CH e sensores).

Para a abordagem apresentada neste trabalho, o algoritmo de setorização utilizado tem como objetivos aumentar a conectividade dentro dos setores, reduzir a latência dos sensores mais relevantes e aumentar a longevidade da rede. Os *cluster heads* atuarão como *sinks*, portanto, serão móveis e o número de setores será pré-determinado e inalterado durante o tempo de vida da RSSF. Os sensores dentro de cada setor poderão fazer comunicações de múltiplos saltos com os *sinks/cluster heads* caso necessário e a comunicação entre setores não será utilizada. O processo de setorização será centralizado com *cluster heads* pré-determinados.

4.3 Setorização baseada em relevâncias

De forma a setorizar a RSSF e agrupar os sensores, considerando seus níveis de relevância (RL), a proposta deste trabalho é utilizar o algoritmo de setorização *k-means* modificado.

O algoritmo *k-means* é a abordagem mais utilizada para setorização [Abdul Latiff et al. 2007]. Ele consiste em encontrar o centroide (ponto central) de um setor, de forma que as distâncias de todos os nós que fazem parte do mesmo grupo ao centro calculado pelo algoritmo seja as menores possíveis [Hartigan 1975]. Esse algoritmo exige que o número de setores seja pré-estabelecido, o que será feito a partir do número de *sinks* que serão utilizados na RSSF (para cada *sink*, um setor).

O processo de setorização por *k-means* é feito em dois procedimentos que se repetem: atribuição e atualização. Porém, antes da primeira atribuição, os centroides devem ter uma posição inicial. Essa etapa, denominada inicialização, tem muita influência no resultado da setorização [Hamerly e Elkan 2002]. Dois métodos de inicialização foram utilizados inicialmente para implementar a solução proposta por este trabalho: *Forgy* e *Random partition*.

O método *Forgy* funciona escolhendo nós existentes do conjunto total como pontos de partida para os centroides. No caso das RSSF, se for preciso posicionar três *sinks* móveis, três sensores são escolhidos aleatoriamente como posições iniciais para os

centroides. Já o método *Random partition* define pontos aleatórios dentro da área a ser setorizada. Novamente, no caso das RSSF, para o posicionamento de três *sinks* móveis, o algoritmo deverá escolher três pontos dentro da área monitorada pela rede [Peña et al. 1999].

No processo de atribuição, cada nó é associado ao centroide mais próximo dele, formando os setores. A Equação 4.3 demonstra a obtenção do centroide mais próximo. Para cada iteração t , cada setor S_i será formado pelos nós denotados por s_p , de forma que a distância de cada nó ao centroide m_i desse setor seja menor que as distâncias do nó aos outros setores, sendo k o número total de setores.

$$S_i^{(t)} = \{s_p : \|s_p - m_i^{(t)}\|^2 \leq \|s_p - m_j^{(t)}\|^2 \forall j, 1 \leq j \leq k\} \quad (4.3)$$

Após a atribuição dos nós aos setores, é feito o processo de atualização. Nesta etapa, as posições dos centroides de cada setor são atualizadas de acordo com as posições dos nós. Como no processo de atribuição os nós podem mudar de setor, a etapa de atualização servirá para definir os novos centroides. As equações 4.4 e 4.5 descrevem esse procedimento. As coordenadas da nova posição do centroide na iteração $(t + 1)$ de cada centroide m_i será igual à soma dos valores das coordenadas de cada nó pertencente ao setor S_i , dividido pela quantidade de nós neste setor. Os dois procedimentos são então repetidos até o ponto de convergência (as posições dos centroides de cada setor não mudam mais após as iterações). Após a convergência, os setores estarão definidos.

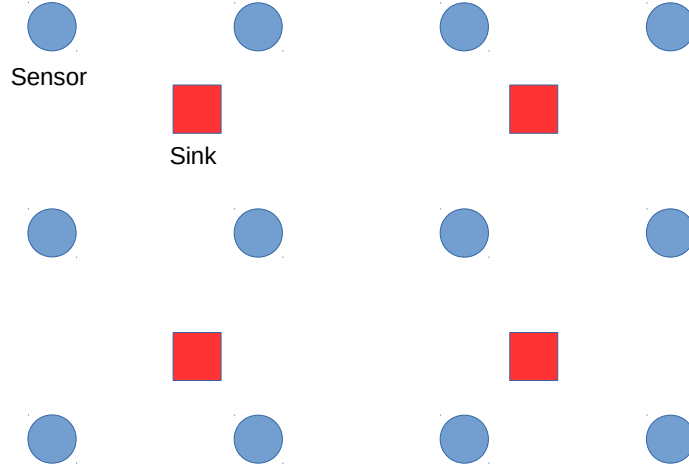
$$x(m_i)^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_i \in S_i^{(t)}} x_i \quad (4.4)$$

$$y(m_i)^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{y_i \in S_i^{(t)}} y_i \quad (4.5)$$

4.3.1 Método proposto

Neste trabalho, um terceiro método foi implementado. Chamado de *Relevance-based k-means clustering* [Peixoto e Costa 2015], essa abordagem considera os níveis de relevância dos sensores na RSSF para determinar as posições iniciais dos centroides na setorização *k-means*.

Inicialmente, os centroides (que em uma RSSF, serão os próprios *sinks*) são posicionados na área da rede formando uma grade, conforme a Figura 4.2. Em seguida, para cada centroide, o algoritmo procura o sensor mais próximo e usa sua posição como coordenadas iniciais para o centroide. Após ter sido escolhido uma vez, o sensor não poderá ser escolhido novamente para outro centroide, evitando que *sinks* diferentes ocupem a mesma posição na rede.

Figura 4.2: Quatro *sinks* em grade.

Após a inicialização dos centroides, cada sensor é atribuído a um setor, formando grupos, de forma semelhante à etapa de atribuição do algoritmo *k-means* convencional. Por último, as posições iniciais dos centroides são balanceadas de acordo com os níveis de relevância dos setores. Nesta etapa, os centroides são movidos em direção aos setores com maiores somas de níveis de relevância de seus sensores, de acordo com as Equações 4.6 e 4.7. Para cada centroide, sua nova posição será alterada em relação a cada outro grupo de sensores. Seja l_j um centroide distinto, a posição atual de k_i deverá ser balanceada de acordo com a distância de k_i a l_j , proporcional à soma dos níveis de relevância dos sensores associados a l_j em relação à soma dos níveis de relevância de todos os sensores da rede. Isso é feito para todo centroide k_i em relação a cada outro centroide l_j . O Algoritmo 1 descreve o método de inicialização proposto.

$$x(k_i) = x(k_i) + ((x(l_j) - x(k_i)) \cdot \frac{RL(l_j)}{RL(S)}), \forall i \in K, \forall l \in K - \{k_i\} \quad (4.6)$$

$$y(k_i) = y(k_i) + ((y(l_j) - y(k_i)) \cdot \frac{RL(l_j)}{RL(S)}), \forall i \in K, \forall l \in K - \{k_i\} \quad (4.7)$$

Após esta etapa de inicialização, os procedimentos de atribuição e atualização do algoritmo *k-means* são executados normalmente, até o ponto de convergência e os setores estarão definidos.

4.4 Posicionamento de múltiplos *sinks*

Para efetuar o posicionamento de múltiplos *sinks* considerando os níveis de relevância, foi proposto o uso do algoritmo descrito na Seção 4.1 para cada setor obtido

Algoritmo 1 Etapa de inicialização do *Relevance-based k-means clustering***Entrada:** conjunto de sinks K , conjunto de sensores S

```

1: posiciona sinks em forma de grade
2: para todo  $k \in K$  faça
3:   repita
4:      $n \leftarrow$  próximo nó mais perto de  $k$ 
5:   até  $usado[n] = \text{falso}$ 
6:      $k_x \leftarrow n_x$ 
7:      $k_y \leftarrow n_y$ 
8:    $usado[n] \leftarrow \text{verdadeiro}$ 
9: fim para
10: para todo  $s \in S$  faça
11:    $k \leftarrow$  sink mais perto de  $s$ 
12:   associa  $s$  ao setor de  $k$ 
13: fim para
14: para todo  $k \in K$  faça
15:   para todo  $l \in K$  faça
16:      $k_x \leftarrow k_x + (l_x - k_x) * RL(l)/RL(S)$ 
17:      $k_y \leftarrow k_y + (l_y - k_y) * RL(l)/RL(S)$ 
18:   fim para
19: fim para

```

através do método *Relevance-based k-means clustering*.

Como cada *sink* será responsável pela coleta de dados de um setor e o número de *sinks* já estará pré-determinado pela aplicação, o método de setorização com uso de *k-means* será capaz de definir os grupos de sensores de forma apropriada, considerando seus níveis de relevância. Dessa forma, o algoritmo de posicionamento único é executado em cada setor, definindo a posição do seu *sink*. Esta abordagem possibilita o posicionamento de múltiplos *sinks* em uma RSSF, considerando as relevâncias dos sensores.

Após posicionados os *sinks*, eles podem receber dados dos sensores em sua volta. Cada sensor envia pacotes para o próximo nó na rota até o *sink*, mantendo o tráfego dentro do mesmo setor, ou seja, pacotes de dados não trafegam entre setores diferentes.

Os *sinks* permanecem em suas posições estabelecidas até que o padrão ou a topologia da rede mude (por exemplo, um sensor para de transmitir dados, um sensor deixa de funcionar, o nível de relevância de um ou mais sensores é alterado, etc.). Caso haja algum tipo de mudança, a setorização é refeita e novas posições para os *sinks* são calculadas [Peixoto e Costa 2015].

4.5 Restrições de posicionamento

Após determinar as melhores posições para os *sinks*, algumas restrições de posicionamento são verificadas. Em se tratando de aplicações do mundo real, nem sempre um *sink* pode se mover para qualquer posição. No caso de aplicações para cidades inteligentes, por exemplo, temos as vias que são usadas pelos equipamentos para se moverem pela cidade (um *sink* poderia estar instalado em um carro), o que impede a entrada em alguns locais, como prédios e casas. Em se tratando de ambientes selvagens, como uma floresta, lagos podem ser impeditivos para os *sinks*, portanto, devem ser evitados. Nos dois exemplos citados, temos uma situação onde a movimentação do *sink* é limitada enquanto há uma outra situação na qual existem locais onde os *sinks* não podem ser posicionados. Em ambos os casos, temos regiões que são consideradas *zonas proibidas* (ZP).

A forma de determinar as zonas proibidas depende da aplicação. Em alguns casos, teremos os caminhos fixos pelos quais os *sinks* poderão se movimentar. Nessa situação, toda a área da rede que está fora dos caminhos determinados é considerada uma ZP. Porém, nos casos onde a movimentação do *sink* é livre, as ZPs podem ser pré-determinadas pela aplicação. No desenvolvimento deste trabalho, essas restrições são válidas apenas para os *sinks*, de forma que os sensores podem ocupar qualquer posição na rede.

4.5.1 Posicionamento com ZPs pré-determinadas

Voltando ao exemplo da floresta onde as lagoas são consideradas zonas proibidas para os *sinks*, o cálculo de sua posição final deve levar em consideração as localizações dessas zonas. Após determinar a posição final de um *sink*, caso ele esteja em uma ZP, seu posicionamento deve ser corrigido movimentando-o para a borda mais próxima da ZP, de forma a retirá-lo dessa zona, como mostra a Figura 4.3.

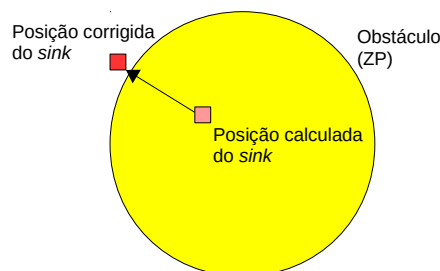


Figura 4.3: Correção da posição de um *sink* em uma ZP.

Para fins de simulação, foram consideradas zonas proibidas em formato de circunferência. Nesse caso, a verificação é determinada através do cálculo da distância do

sink ao centro da ZP. Se essa distância for menor ou igual ao raio da circunferência, significa que ele está dentro da ZP e deve ser movido. A nova localização do *sink* é então determinada pelas equações 4.8 e 4.9, onde x_s e y_s são as coordenadas do *sink* (ainda dentro da ZP), x_o e y_o são as coordenadas da ZP (ponto central da circunferência) e p é a proporção entre o raio da ZP e a distância entre seu ponto central e o *sink*, definida por $p = \text{raio}/\text{distância}$.

$$x = (x_s + x_o) \cdot p + x_o \quad (4.8)$$

$$y = (y_s + y_o) \cdot p + y_o \quad (4.9)$$

No caso específico onde o *sink* fica no ponto central da circunferência ($x_s = x_o$ e $y_s = y_o$), a distância entre a posição calculada do *sink* e qualquer ponto na borda da circunferência será a mesma. Quando isso ocorre, o *sink* é movido em direção ao sensor mais próximo que faça parte do mesmo setor, de forma a não perder conectividade.

4.5.2 Posicionamento com vias de movimentação

No caso de existirem vias definidas para a movimentação dos *sinks*, como ruas e avenidas, por exemplo, a abordagem muda. Nessa situação, os *sinks* devem ser posicionados apenas nos segmentos de retas definidos pelas vias, de forma que todo o resto da rede é uma zona proibida. Porém, ao invés de usar a mesma abordagem de posicionamento com ZPs pré-determinadas, após o cálculo da posição final do *sink*, é preciso procurar a via mais próxima a ele e corrigir sua posição.

Para cada *sink* da rede, deverá ser encontrada a via mais próxima a ele e calculada a projeção do *sink* nessa via. A projeção de um ponto P em uma reta r é o ponto de intersecção entre a reta r e uma outra reta s que passa pelo ponto P e é perpendicular a r . Como a via não é uma reta, mas um segmento de reta, existem casos onde a projeção ficará fora da área permitida, o que pode ser facilmente corrigido escolhendo a extremidade mais próxima da via. As Figuras 4.4(a) e 4.4(b) ilustram as duas situações.

O novo ponto encontrado será a posição final do *sink*. A Equação 4.10 demonstra a escolha da posição final (S_f).

$$S_f = \begin{cases} P & \text{se } P \in r \\ A & \text{se } d(S, A) < d(S, B) \\ B & \text{se } d(S, A) > d(S, B) \end{cases} \quad (4.10)$$

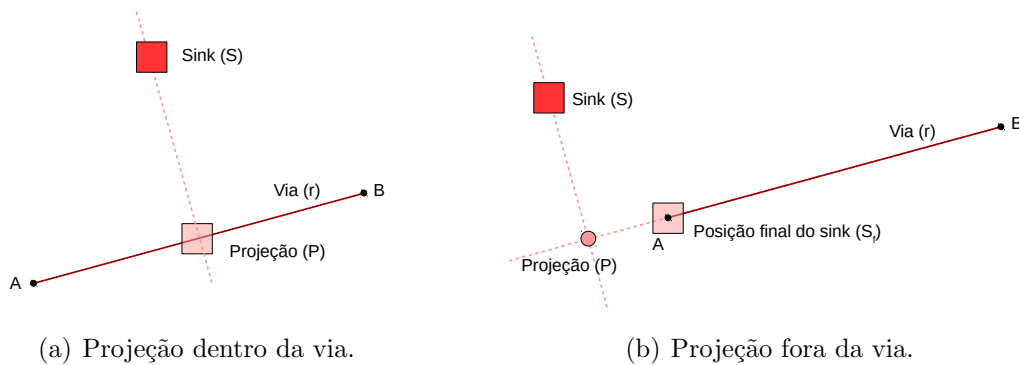


Figura 4.4: Projeção da nova posição de um *sink* em uma via.

4.5.3 Zonas Não Conectadas

Devido ao raio de alcance dos transmissores de rádio dos nós, é possível que um *sink* seja posicionado em um local onde não seja possível se conectar a nenhum sensor. Na abordagem desse trabalho, essas áreas onde não há conectividade são chamadas de *zonas não conectadas* (ZNC). Para evitar que algum *sink* permaneça em uma ZNC, impedindo que ele receba dados da rede, é verificado se existe pelo menos um sensor no seu raio de alcance, possibilitando a comunicação entre eles. Caso não exista, o *sink* é movido em direção ao sensor mais próximo até que seja possível estabelecer uma conexão. Porém, em casos onde há restrições de posicionamento por vias, o *sink* deve se movimentar pelos caminhos determinados por elas (a Figura 4.5(a) ilustra um exemplo de uma rede com vias onde toda a via está em uma ZNC). Para solucionar esse problema, o algoritmo precisa calcular as ZNCs de uma via, de forma que seja possível determinar em quais locais o *sink* poderá ser posicionado.

Inicialmente, todo o percurso das vias na rede são considerados ZNCs. Em seguida, para cada sensor próximo o suficiente de uma via (ou seja, onde a distância do sensor para a via seja menor ou igual ao seu raio de alcance), são calculados os pontos de intersecção entre o seguimento de reta que representa a via e a circunferência formada pelo raio de alcance do sensor. O segmento de reta formado por estes dois pontos é considerado uma *zona conectada* (ZC) na via, como ilustra a Figura 4.5(b). Após repetirmos esse cálculo para cada sensor, obtendo as ZCs das vias que estão próximas a eles, teremos um conjunto de ZCs onde podemos posicionar os *sinks*. Dessa forma, caso algum *sink* esteja em uma ZNC após ter sido determinada sua posição final, o algoritmo o movimenta para a ZC mais próxima. Vias que são ZNC ou com trechos não conectados não impedem que os *sinks* se movimentem por elas, as ZNCs apenas determinam trechos nas vias que não podem ser usados como posições finais para os *sinks*.

Uma ZC em uma via é calculada através da solução de um sistema de equações contendo a equação geral da reta, formada pelas extremidades do segmento de reta que representa a via e a equação reduzida da circunferência, formada pelo raio de

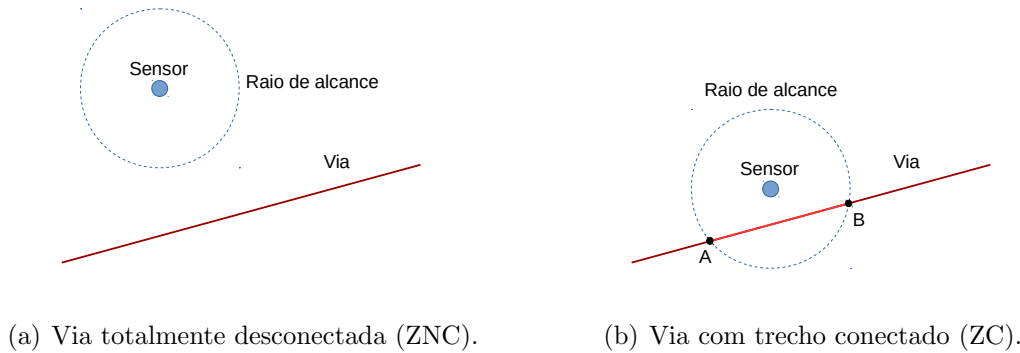


Figura 4.5: Exemplos de ZNC e ZC.

alcance do sensor, como mostra a Equação 4.11. A solução do sistema nos dá uma equação do 2º grau cujo o cálculo do seu discriminante (Δ) informa quantos pontos de intersecção existem. Caso $\Delta > 0$, a reta é secante à circunferência e possui dois pontos de intersecção. Como os pontos podem estar fora do segmento de reta (já que a equação geral formada considera uma reta e não um segmento), é preciso verificar se eles estão dentro da via e aplicar as devidas limitações: caso algum ponto esteja fora da via, deverá ser usada a extremidade mais próxima em seu lugar.

$$\begin{cases} ax + by + c = 0 \\ (x - a)^2 + (y - b)^2 = r^2 \end{cases} \quad (4.11)$$

4.5.4 Rotas para movimentação

Sempre que a topologia da rede muda, por exemplo, um sensor desliga, novos sensores são ativados ou o valor do nível de relevância de algum sensor muda, as posições dos *sinks* devem ser recalculadas. Devido a esta mudança, o algoritmo descrito neste capítulo deve ser executado novamente, desta vez, com a nova configuração da RSSF e as posições dos *sinks* poderão ser alteradas. Quando isso acontece em uma rede com restrições de posicionamento, a movimentação dos *sinks* deve obedecer tais restrições.

Para solucionar este problema, um caminho deve ser indicado, de forma que o *sink* tenha uma rota de movimentação até sua posição final. A última parte do algoritmo proposto cria um grafo onde os vértices são modelados a partir das posições iniciais dos *sinks*, suas posições finais e cada ponto de intersecção dos segmentos de reta que representam as vias. Ou seja, teremos tantos vértices no grafo quanto pontos de intersecção e posições iniciais e finais dos *sinks*. Já as arestas são modeladas a partir de cada segmento de reta na rede, como demonstra a Figura 4.6. Dessa forma, é possível construir um grafo e usar o algoritmo de menor caminho de Dijkstra [Dijkstra 1959] para determinar a rota mais curta para cada *sink*.

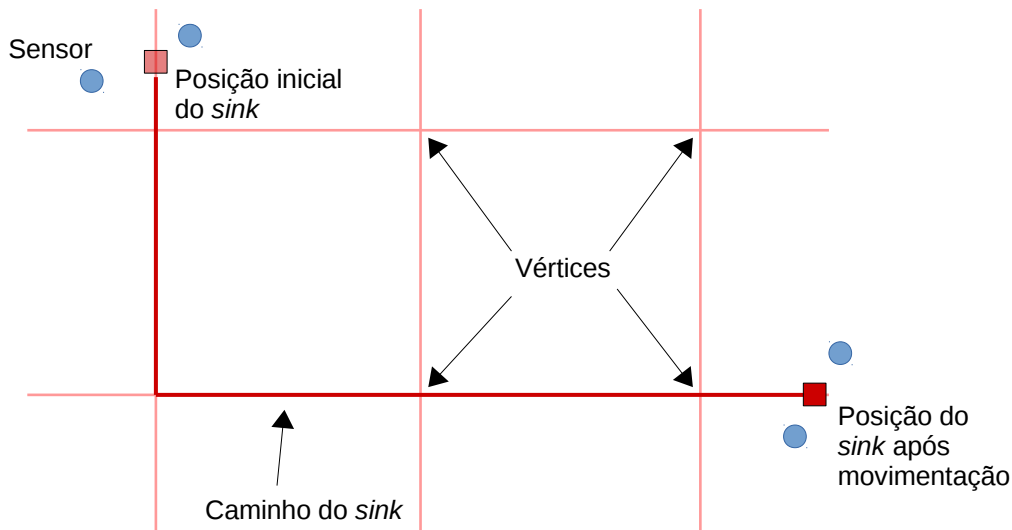


Figura 4.6: Exemplo de uma rota calculada com o algoritmo de Dijkstra.

Na Figura 4.6, um *sink* que estava em um canto da rede precisava se mover para outra posição. O modelo trata cada segmento de reta como uma aresta e cada ponto de intersecção como um vértice. Além disso, a posição atual do *sink* e sua posição final são tratadas como vértices de origem e de destino respectivamente. Após a construção do grafo, o algoritmo de Dijkstra pode dizer o caminho mais curto para que o *sink* possa chegar à sua posição final. O algoritmo considera que o grafo não é direcionado, ou seja, as vias que estão sendo modeladas são de mão-dupla. A conectividade da rede não é considerada no cálculo do menor caminho, ou seja, é possível que durante a movimentação do *sink*, ele perca a conexão com os sensores.

O algoritmo de Dijkstra ainda permite que as arestas do grafo possuam pesos diferentes. Na abordagem proposta, inicialmente cada aresta recebe como seu peso o valor do seu comprimento, o que possibilita o algoritmo de Dijkstra encontrar o menor caminho entre a posição anterior do *sink* e sua nova posição. É possível ainda usar multiplicadores para aumentar os pesos das arestas, modelando as condições de tráfego nas vias. Por exemplo, uma via curta, mas com tráfego muito pesado, teria seu peso inicial (comprimento da via) multiplicado por um valor que define a intensidade do tráfego, gerando a possibilidade de seu peso final ser maior que o peso de outra via mais longa, porém, com tráfego menor.

Os multiplicadores permitem simular as condições de tráfego em uma cidade e obter rotas melhores para o posicionamento dos *sinks*. Uma aplicação que deseje considerar as condições das vias para o cálculo de rotas pode facilmente obter esse resultado apenas definindo valores de intensidade de tráfego (leve, intenso, parado) ou tipos de vias (estradas de terra, calçamento, asfalto, esburacadas, etc.) e aplicando-os nos pesos das arestas.

Capítulo 5

Resultados Experimentais

Após as simulações do algoritmo de posicionamento de múltiplos *sinks* baseado em relevância, espera-se que seja possível diminuir o consumo de energia dos nós em uma RSSF em comparação com abordagens de *sinks* estáticos e também com movimentação pré-definida. Mais especificamente, espera-se conseguir preservar a energia nos sensores mais relevantes da rede e nos sensores próximos a eles, que fazem encaminhamento dos pacotes de tais sensores.

Com a publicação desse método de posicionamento de múltiplos *sinks* baseado em relevância, novos procedimentos de transporte de dados e roteamento, também baseados em relevância, podem ser desenvolvidos, ampliando ainda mais o tempo de vida e a vazão nesses tipos de redes de sensores. Aplicações que distinguem os sensores em prioridade poderão ser beneficiadas com essas novas abordagens.

5.1 Comparação dos métodos de inicialização *Forgy*, *Random partition* e *Relevance-based*

Durante a etapa de simulações da abordagem proposta neste trabalho, foi percebido que os métodos de inicialização para o algoritmo de setorização *k-means* do tipo *Forgy* e *Random partition* frequentemente geravam grupos de sensores visivelmente desfavoráveis à aplicação de posicionamento de múltiplos *sinks* em uma RSSF.

Por não utilizar os níveis de relevância dos nós da rede, esses dois métodos dividiam os sensores em grupos considerando apenas as posições dos nós. Ainda, por usar alguns procedimentos aleatórios, nem sempre a execução repetida do algoritmo na mesma rede gerava os mesmos grupos, o que torna a abordagem instável e não confiável.

A Figura 5.1(a) apresenta a disposição de dois *sinks* em uma rede de quatro sensores, usando o método de inicialização *Forgy* e em seguida, executando o algoritmo *k-means*. A abordagem conseguiu dividir a rede em dois grupos, de fato, dividindo a

quantidade de sensores por igual nos grupos. Porém, um sensor de nível $RL = 10$ é forçado a dividir um *sink* com outro sensor com $RL = 1$. Como o sensor do canto superior direito sozinho supera todo o resto da rede em níveis de relevância, seria mais apropriado que ele tivesse um *sink* dedicado.

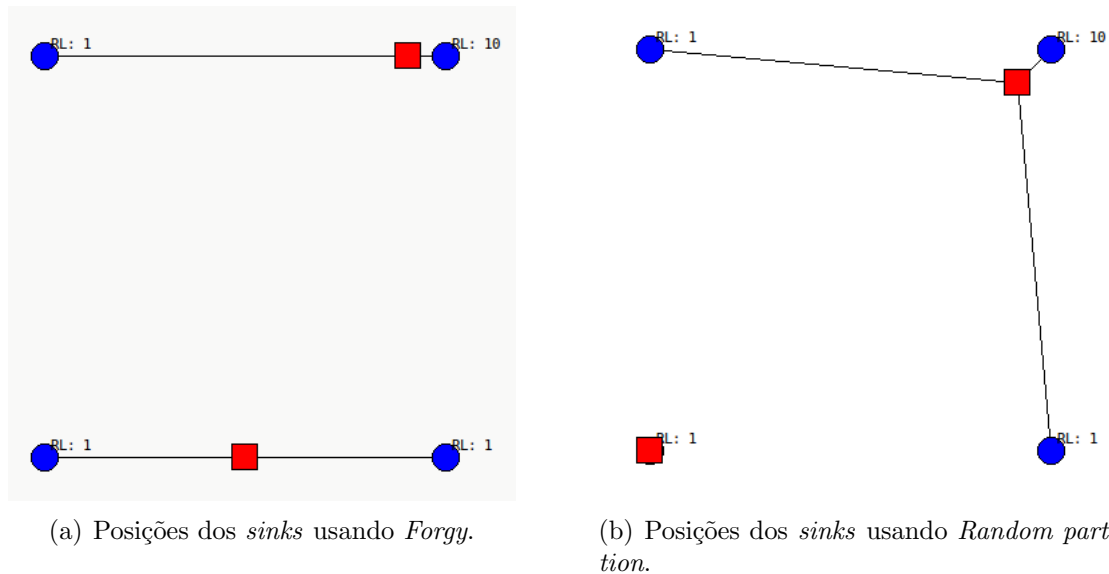


Figura 5.1: Posicionamento dos *sinks* usando métodos de inicialização convencionais.

Já a Figura 5.1(b) apresenta a disposição dos *sinks* na mesma configuração de rede, porém, usando o método de inicialização *Random partition*. Dessa vez, a disposição dos *sinks* fica ainda pior, pois enquanto um dos *sinks* está dedicado a um sensor de $RL = 1$ (muito pouco relevante), o outro fica responsável por dois sensores de $RL = 1$ e o sensor de $RL = 10$ (bem relevante). No método de inicialização *Random partition*, a cada nova execução, os grupos são formados de maneira diferente. Sendo assim, não há como garantir que em todas as tentativas os setores serão gerados em uma configuração benéfica à rede.

Por fim, a Figura 5.2 demonstra as posições dos dois *sinks* quando utilizado o método de inicialização *Relevance-based*, proposto nesse trabalho. Nesta abordagem, o sensor com $RL = 10$ recebe um *sink* dedicado, enquanto o outro se torna responsável por coletar os dados dos nós restantes. De fato, os outros três sensores juntos não são mais importantes para a aplicação que o sensor do canto superior direito. Esse posicionamento dos *sinks* supera os métodos de *Forgý* e *Random partition* em se tratando de uma RSSF que considera níveis de relevância dos nós.

5.2 Simulação

Para verificar a eficiência das abordagens propostas, foi desenvolvido um simulador em linguagem C++ para posicionamento de *sinks*. O *software* simula o consumo de

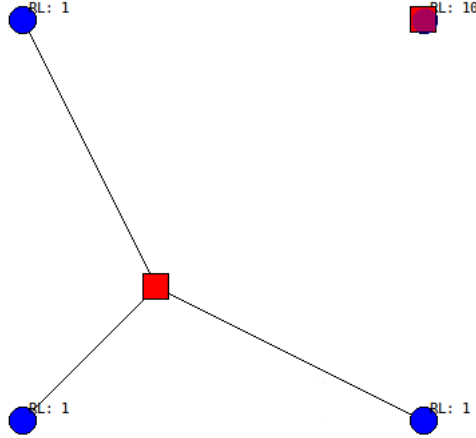


Figura 5.2: Posicionamento dos *sinks* usando o método de inicialização *Relevance-based*.

energia em uma RSSF com *sinks* estáticos, em movimentação fixa e com posicionamento baseado em relevâncias, sendo que neste último método, a setorização da rede é feita usando o algoritmo *k-means* com a possibilidade de selecionar os métodos de inicialização *Forgy*, *Random partition* e *Relevance-based* (proposto neste trabalho).

O modelo de consumo de energia utilizado nas simulações não considera perdas de pacotes nem congestionamento, ainda assim, é possível comparar o gasto de energia dos nós na abordagem proposta contra abordagens de *sinks* estáticos e de movimentos fixos [Costa et al. 2014]. Esse modelo define o consumo de energia para transmissão (E_t) e para recepção de dados (E_r). Para calcular estes valores, foram considerados também a quantidade de bits transmitidos de um nó para outro, além da energia consumida para efetuar cada transmissão e recepção em um caminho do sensor s ao *sink* do seu setor, formado por $H(s)$ nós intermediários.

Seja $D(s, h)$ a quantidade de *bits* que será transmitida da origem s pelo nó h para o nó $(h + 1)$. $Pw_t(s, h)$ e $Pw_r(s, h)$ são as potências de transmissão e recepção no nó h respectivamente, enquanto que $T(s, h)$ representa o tempo gasto para transmitir ou receber 1 *bit* oriundo de s no nó h , temos o modelo de consumo de energia representado nas equações 5.1 e 5.2.

$$E_t(s, h) = \begin{cases} 0 & \text{se } h = H(s) + 1 \\ D(s, h) \cdot Pw_t(s, h) \cdot T(s, h) & \text{se } h \leq H(s) \end{cases} \quad (5.1)$$

$$E_r(s, h) = \begin{cases} 0 & \text{se } h = 0 \\ D(s, h) \cdot Pw_r(s, h) \cdot T(s, h) & \text{se } h > 0 \end{cases} \quad (5.2)$$

De acordo com as equações, o consumo de energia para a transmissão de um pacote originado em s no sensor h é definido pela quantidade de bits do pacote, multiplicado

pela potência e tempo necessários para a transmissão de cada bit. Neste modelo foi considerada a potência máxima para transmissão e recepção dos nós, desconsiderando a distância entre os sensores (quando estão próximos, o consumo pode ser reduzido ainda mais diminuindo a potência dos rádios). A cada transmissão, o consumo é calculado e subtraído da energia residual do sensor. O mesmo ocorre no momento em que o nó recebe dados para encaminhar adiante.

Foram simuladas três redes diferentes com o objetivo de verificar o comportamento do algoritmo em situações diversas. A primeira rede consiste em um conjunto de 97 sensores sem restrições de posicionamento. É verificado o consumo de energia com dois e cinco *sinks*, estáticos e móveis. A segunda rede possui restrições de posicionamento, onde vias são modeladas e nove sensores atuam de forma a seguir uma programação pré-definida de atividades. Na terceira rede, também há restrições de posicionamento, com vias irregulares para movimentação dos *sinks* e com doze sensores, que também seguem uma programação de atividades. Além dos doze sensores, outros cem nós com níveis de relevância zero ($RL = 0$) são posicionados na rede em forma de uma grade, servindo apenas de nós intermediários para o encaminhamento de pacotes. Nas duas últimas redes, além do consumo de energia, o número médio de saltos necessários por cada sensor é analisado.

Para executar as simulações, foram usados alguns parâmetros para o consumo de energia baseados nos dados em [Costa et al. 2014]: 60 mW para transmissão e recepção dos dados, 4 μ s para transmissão de cada *bit* e pacotes de 1016 *bits* (127 *bytes*) cada.

5.2.1 Simulação sem restrições de posicionamento

Para esta simulação, foram utilizados 97 sensores. Destes, três com níveis de relevância $RL = 15$ foram posicionados no canto superior-esquerdo da rede, dois com $RL = 5$ no centro e mais dois com $RL = 1$ no lado oposto. Os outros 90 sensores foram posicionados uniformemente, formando uma grade, de forma a mantê-los espalhados pela área da rede, todos com níveis de relevância $RL = 0$, atuando apenas como intermediários na comunicação entre os sensores que geram dados e os *sinks*. A Figura 5.3 demonstra a localização de cada sensor na simulação.

Cada sensor possui um padrão de geração de pacotes de acordo com seu nível de relevância, apresentado na tabela 5.1. Sensores com maiores valores de RL geram mais dados.

Sob essa configuração, quatro simulações foram executadas: 2 *sinks* estáticos (sem cálculo de posição); 5 *sinks* estáticos (sem cálculo de posição); 2 *sinks* móveis com relevância; e 5 *sinks* móveis com relevância. As posições dos *sinks* estáticos nas simulações são apresentadas nas Figuras 5.4(a) e 5.4(b). Nas simulações onde são utilizados *sinks* móveis com relevância, suas posições são calculadas através do método proposto neste trabalho. Como os níveis de relevância dos sensores não são

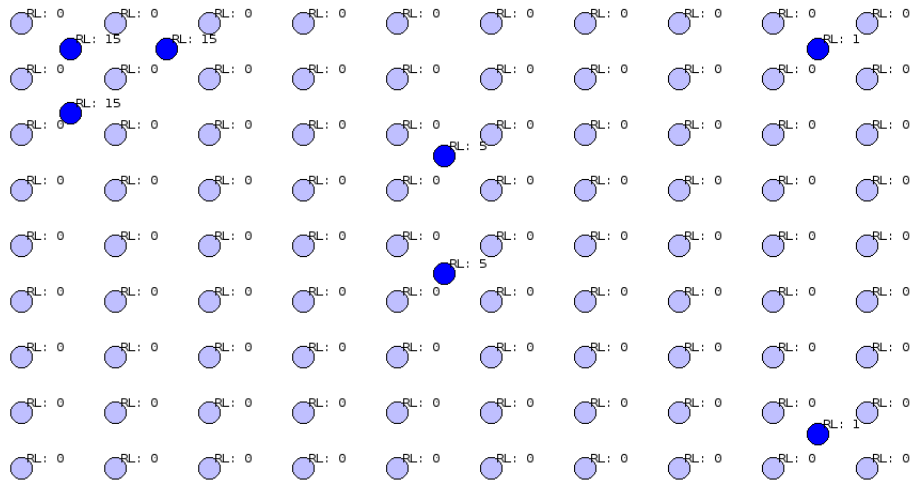


Figura 5.3: Posicionamento dos sensores na simulação, com destaque para sensores com $RL > 0$.

Tabela 5.1: Taxa de geração de pacotes de acordo com níveis de relevância.

RL	Pacotes/s
0	0
1 a 4	10
5 a 10	20
11 a 14	30
15	40

alterados durante a simulação, as posições dos *sinks* permanecerão inalteradas até o final do experimento.

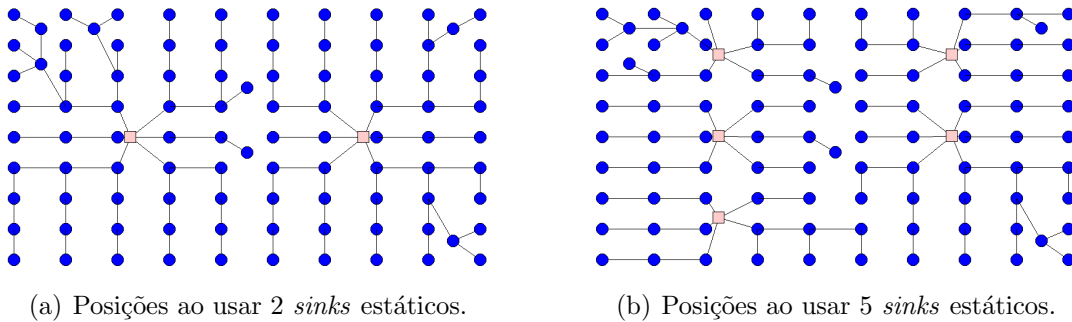


Figura 5.4: Posições iniciais dos *sinks* nas simulações.

Nas quatro simulações, foi analisado o consumo de energia de toda a rede. Era esperado que quanto maior o número de *sinks* disponíveis, menor fosse o consumo de energia, pois os sensores necessitarão de menos saltos para transmitir seus da-

dos. Além disso, com o uso de um posicionamento inteligente e móvel, é possível movimentar os *sinks* de forma a ficarem mais próximos dos sensores que transmitem mais pacotes.

Após cada simulação, os dados fornecidos pelo simulador foram coletados e comparados, gerando o gráfico exibido na Figura 5.5. É possível perceber que a abordagem de posicionamento baseado em relevância foi superior ao uso de *sinks* estáticos. Ainda, com apenas dois *sinks* móveis, foi possível consumir menos energia que ao usar cinco *sinks* estáticos.

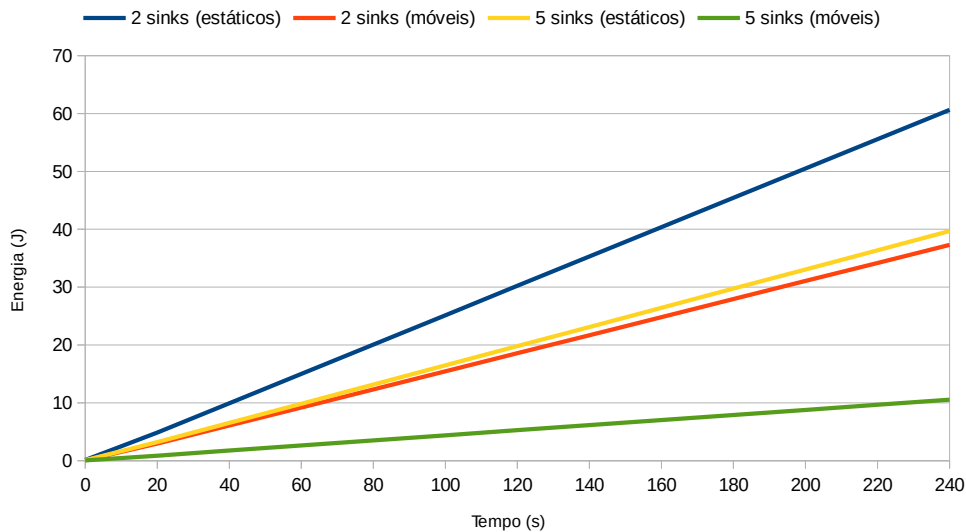


Figura 5.5: Consumo de energia nas simulações sem restrições de posicionamento.

Esse resultado se dá ao fato de que, ao posicionar os *sinks* próximos aos sensores que produzem mais pacotes, menor será o caminho que os dados deverão percorrer para chegar ao seu destino. Com menos saltos sendo utilizados, menos sensores precisam gastar energia para encaminhar os pacotes, preservando o tempo de vida da rede. Ainda, ao priorizar os sensores mais relevantes para a aplicação, estes serão os últimos a pararem por falta de energia. Isto ocorre porque prolongando a execução da aplicação.

5.2.2 Simulação com restrições de posicionamento

Para as simulações com restrições de posicionamentos, duas redes foram modeladas. A Rede 1 é composta de nove sensores posicionados uniformemente em uma área com oito vias, de maneira a formar ruas perpendiculares, como é mostrado na Figura 5.6.

Estes sensores seguiam uma agenda de atividades em uma simulação de 12 horas, como descrito na Tabela 5.2. A tabela indica o comportamento dos sensores em

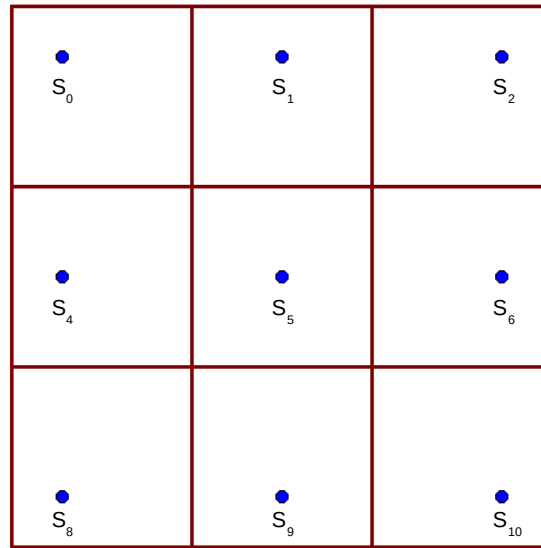


Figura 5.6: Configuração da Rede 1.

cada período de tempo. Para cada período, um valor de RL é mostrado na tabela, ou seja, naquele tempo específico, o sensor é ativado com o nível de relevância indicado, fazendo com que os *sinks* se movam. Um valor de -1 indica que o sensor está inativo, portanto, sem enviar nem encaminhar dados. O objetivo desta agenda de atividades é simular um comportamento dinâmico da rede em um local com ocorrência aleatória de eventos. Sensores podem ligar e desligar de acordo com o que acontece no ambiente e os *sinks* precisam reagir a isto, mudando suas posições.

Tabela 5.2: Programação de atividades dos sensores.

Tempo	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}
00:00:00	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
00:30:00	5	10	5							5	15	5
01:00:00	-1	-1	-1									
01:30:00				5		5	15	5				
02:30:00	5	10	5			-1	-1	-1				
03:00:00	-1	-1	-1							-1	-1	-1
04:00:00				5			15	5	5	5	15	
05:30:00				-1			-1	-1	-1	-1	-1	
07:00:00	5	10			5	5				5	15	5
08:30:00	-1	-1			-1	-1				-1	-1	-1
10:00:00					5	5	15		5	5	15	5
12:00:00					-1	-1	-1	-1	-1	-1	-1	-1

Para a Rede 2, 12 sensores foram distribuídos em uma região com vias irregulares, como mostra a Figura 5.7, seguindo a mesma agenda de atividades da Rede 1. Com

os 12 sensores, outros 100 nós com $RL = 0$ foram inseridos de maneira a formar uma grade. O papel desses 100 sensores é de servir como nós intermediários da rede, encaminhando pacotes oriundos dos 12 sensores com níveis de relevância maiores que 0.

Em cada uma das duas redes, três métodos de posicionamento de *sinks* foram simulados: estáticos, móveis sem consideração dos níveis de relevância, além do método proposto neste trabalho. Na simulação sem considerar os níveis de relevância, a importância de cada sensor na rede não era usada como parâmetro de priorização para o cálculo das posições dos *sinks*, porém, eles eram reposicionados sempre que um sensor era ativado ou desativado. Além disso, o mesmo modelo de consumo de energia e a taxa de geração de pacotes usados na simulação sem restrições de posicionamento foram adotados nessas duas redes.

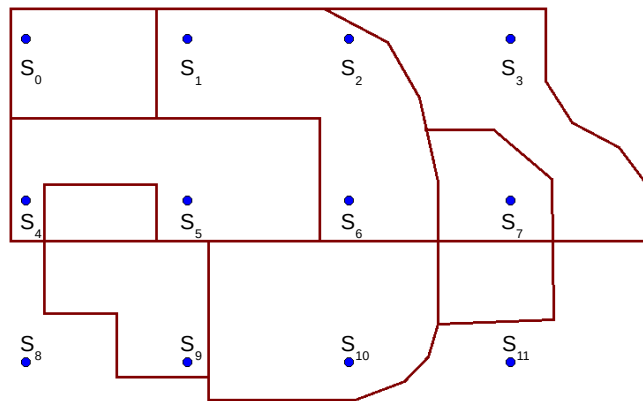


Figura 5.7: Configuração da Rede 2.

Para a Rede 1, os resultados são apresentados no gráfico da Figura 5.8. O uso de *sinks* estáticos para coletar dados dos sensores consumiu mais energia comparado ao uso de *sinks* móveis. Isto se dá ao fato de que como os *sinks* estão em posições fixas, os sensores mais distantes precisam encaminhar os pacotes através dos nós intermediários para que consigam alcançar algum *sink*. Os nós intermediários precisam gastar energia tanto para transmitir os próprios dados como para encaminhar os dados vindo dos outros sensores, consumindo mais energia ao todo. Ao usar *sinks* móveis, o consumo de energia pode ser reduzido porque eles se movimentam para próximo dos sensores. Por último, na abordagem de *sinks* móveis, mas sem considerar os níveis de relevância dos nós, os sensores gastam mais energia comparado à abordagem proposta neste trabalho, porém, ainda conseguem consumir menos energia do que uma rede com *sinks* estáticos.

Já na Rede 2, como é possível verificar no gráfico da Figura 5.9, ao usar a abordagem proposta neste trabalho, menos energia é necessária quando comparada ao uso de *sinks* estáticos e até móveis sem consideração de níveis de relevância.

Também foi comparado o número de saltos que os pacotes precisavam percorrer para alcançar algum *sink*. A latência das transmissões em RSSF depende de várias

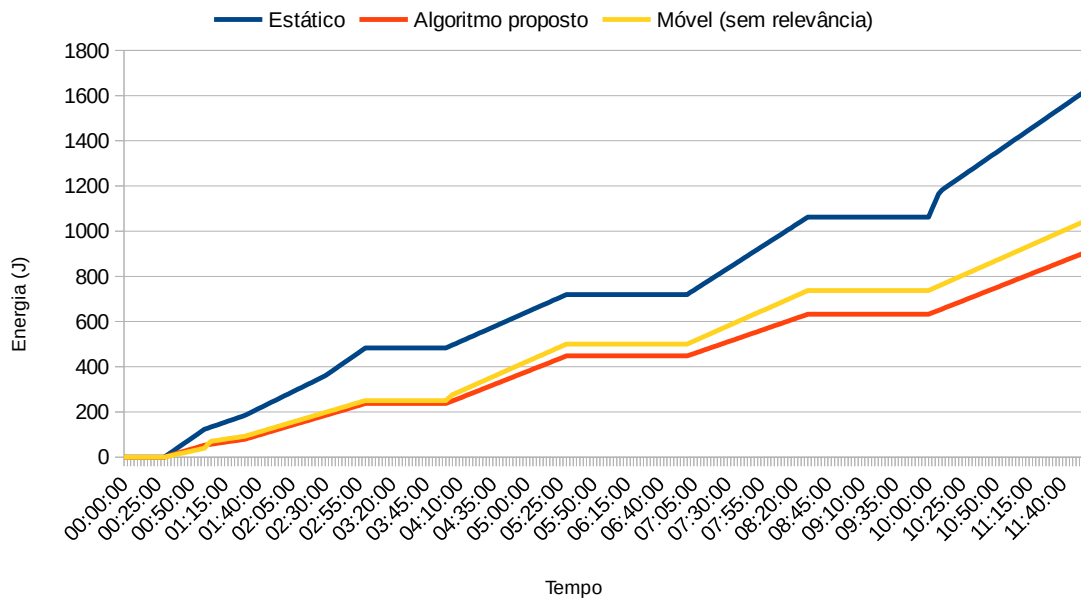


Figura 5.8: Consumo de energia na Rede 1.

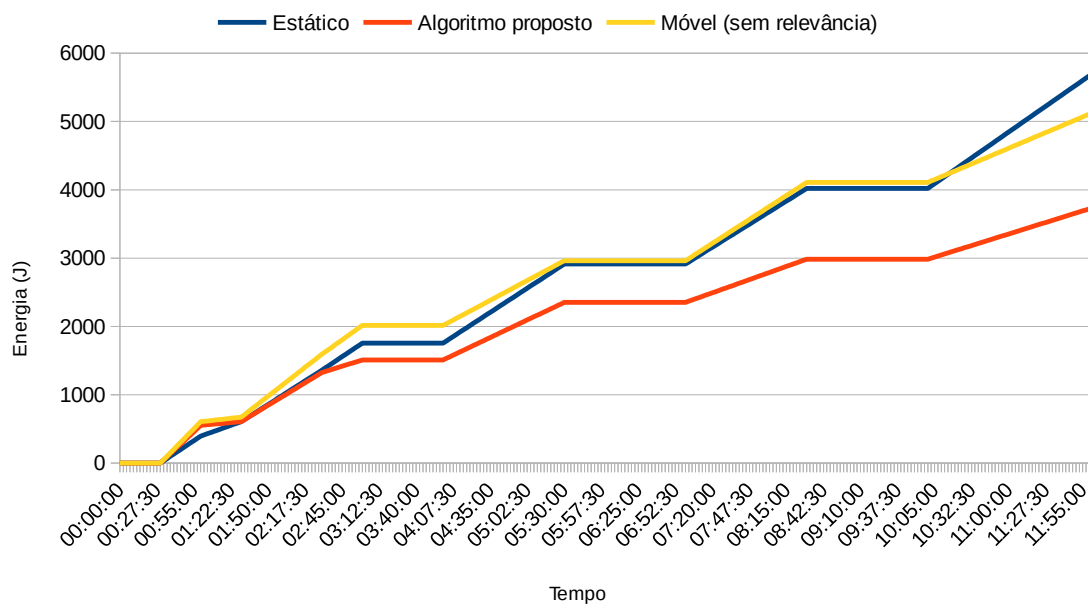


Figura 5.9: Consumo de energia na Rede 2.

características da rede, como protocolo de controle de acesso ao meio, tempo de processamento em nós intermediários e mecanismos de controle de congestionamento e recuperação de erros. Porém, para simplificar a análise, podemos estimar a latência

como uma proporção direta do número de saltos que os pacotes precisam percorrer, o que faz sentido para uma RSSF [Aoun e Boutaba 2006]. Tal métrica pode então ser considerada para avaliar a abordagem proposta.

Neste teste, foi considerado o número de saltos utilizado por cada sensor sempre que ele transmitia algum pacote. Nos momentos em que os sensores estavam inativos ($RL = -1$ na Tabela 5.2), o número de saltos para estes sensores não era contabilizado. Como cada sensor usava o mesmo valor de relevância nos momentos de atividade, foi possível comparar os números de saltos entre eles. Ao final da simulação, o número médio de saltos de cada sensor era a média de saltos entre ele e algum *sink*.

O gráfico na Figura 5.10 apresenta o número médio de saltos para cada sensor. Como esperado, o uso de *sinks* estáticos requereu mais saltos em geral, já que eles não se moviam para perto dos nós mais relevantes. Os sensores S_5 e S_7 foram exceções e precisaram de apenas um salto porque estavam localizados próximos aos *sinks* estáticos, permitindo uma conexão direta com eles.

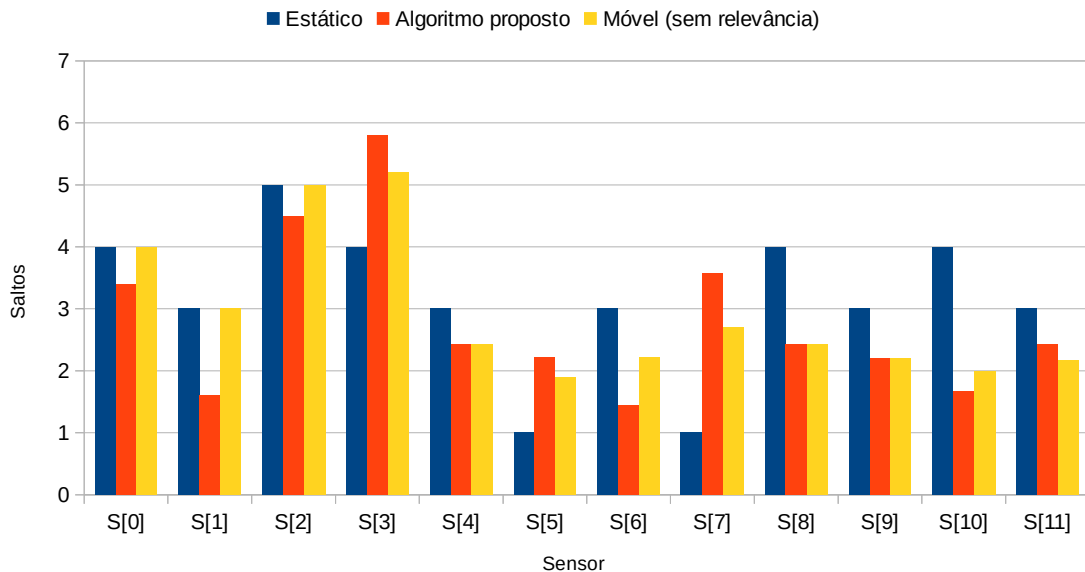


Figura 5.10: Número médio de saltos na Rede 2.

Ao simular a abordagem proposta neste trabalho, os nós mais relevantes (S_1 , S_6 e S_{10}) foram priorizados, portanto, necessitando de menos saltos para alcançar os *sinks*, enquanto os sensores menos relevantes precisaram se comunicar através de mais nós intermediários, já que os *sinks* estarão mais próximos dos sensores mais relevantes da rede. Nesta verificação, espera-se que os nós mais relevantes tenham uma menor latência. Isto não acontece ao usar *sinks* móveis sem distinção de relevância devido ao fato de que, como cada sensor tem a mesma importância para a

rede, não haverá diferenciação entre eles. É possível verificar no gráfico da Figura 5.10 que neste último caso, os sensores mais relevantes (S_1 , S_6 e S_{10}) não obtiveram vantagens em número de saltos comparados com os outros nós.

A ferramenta de simulação que foi desenvolvida para estas simulações também informa todas as rotas usadas pelos *sinks*. Cada vez que a configuração da rede muda (por exemplo, o nível de relevância de um sensor é alterado, um sensor é ativado ou desativado) as novas posições dos *sinks* são recalculadas e, caso precisem se mover, a ferramenta fornece o caminho completo que cada um deles precisou seguir.

5.3 Discussão dos resultados

Como esperado, ao usar posicionamento de *sinks* baseado em relevâncias, o consumo de energia é menor, comparado ao uso de *sinks* estáticos ou móveis sem distinção de relevâncias.

Na simulação sem restrições de posicionamento, o uso de dois *sinks* móveis foi capaz de consumir menos energia que o uso de cinco *sinks* estáticos. Isso mostra que movimentar os *sinks* para próximo dos sensores mais relevantes pode fazer com que a rede necessite de menos energia ao todo, favorecendo a abordagem proposta. Os resultados dessa simulação mostram ainda que ao usar dois *sinks*, a abordagem proposta neste trabalho consumiu o equivalente a 61,5% da energia consumida ao usar a abordagem estática, enquanto que ao inserir cinco *sinks*, o consumo de energia com o método proposto foi de 26,5% na mesma comparação.

Já nas simulações com restrições de posicionamento, ao simular a Rede 1, o uso da abordagem proposta neste trabalho consumiu 86,7% do total da energia utilizada ao usar *sinks* móveis sem considerar os níveis de relevância, enquanto que na Rede 2, o consumo foi de 72,9% na mesma comparação. Além disso, ao reduzir o número de saltos necessários pelos sensores mais relevantes, a abordagem proposta pode reduzir a latência dos pacotes transmitidos por estes nós.

Essas simulações mostraram que quando os sensores possuem níveis de importância distintos em uma RSSF, soluções otimizadas para o posicionamento de *sinks* móveis podem ser desenvolvidas. Obviamente, a quantidade de energia economizada dependerá diretamente dos níveis de relevância dos sensores e, portanto, diferentes resultados podem ser obtidos. Ainda assim, é esperado que a rede possa operar com maior performance quando os sensores são diferenciados pela sua importância para as funções de monitoramento da rede.

Capítulo 6

Considerações Finais

Redes de Sensores Sem Fio podem ser aplicadas a diversas áreas de pesquisa e indústria como ferramenta para monitoramento de ambientes, rastreamento de objetos, detecção de intrusão, etc. Suas características exigem um consumo inteligente de energia, de forma a prolongar o máximo possível seu tempo de vida.

O posicionamento de *sinks*, se feito de forma adequada, pode reduzir o gasto de energia de uma RSSF, assim como com a aplicação de múltiplos *sinks* em uma única rede.

Este trabalho propôs uma nova abordagem para o posicionamento de múltiplos *sinks* em uma RSSF, o que também levou ao desenvolvimento de um novo método de inicialização para o algoritmo de setorização *k-means*.

Os algoritmos desenvolvidos neste trabalho fazem uso dos níveis de relevância de cada sensor, onde os nós mais importantes devem ser priorizados, mantendo-os em funcionamento pelo maior tempo possível. Simulações demonstraram que a abordagem proposta trouxe uma maior economia de energia em toda a RSSF, além da redução da latência média no tráfego da rede em comparação com o uso de *sinks* estáticos e móveis sem distinção de relevância.

Diversas outras abordagens de posicionamento de *sinks* já existem na literatura de RSSF, porém, nenhuma delas levou em consideração os níveis de relevância dos sensores. Em casos de aplicações onde alguns nós tem importância ou prioridade acima de outros, o método de posicionamento proposto nesse trabalho demonstrou ser vantajoso.

Propõe-se que, futuramente, esse trabalho seja estendido de forma a ser aplicado em cidades inteligentes com veículos autônomos, onde sensores de tráfego podem alterar as características das vias, alterando as possibilidades de movimentação dos *sinks*. Neste tipo de aplicação, a modelagem de vias utilizada neste trabalho através de grafos pode considerar multiplicadores para os pesos das arestas, fazendo com que o algoritmo de menor caminho (Dijkstra) seja capaz de obter a melhor rota de

movimentação para um *sink* considerando a situação do tráfego e a qualidade das vias.

Pretende-se também comparar esta proposta com algoritmos de movimentação fixa, onde os *sinks* se movimentam através de um percurso pré-definido. Nesta situação, diferentes velocidades de movimentação pode trazer resultados distintos, o que nos leva a ter que encontrar um meio de calcular a velocidade que possa trazer um desempenho mínimo exigido pela aplicação. Ao considerar os níveis de relevância dos sensores da RSSF, podemos definir velocidades variáveis para cada área da rede, ou seja, ao passar por uma área com sensores de alta relevância, a velocidade do *sink* é diferente da utilizada ao passar por uma área com sensores de baixa relevância.

Referências Bibliográficas

- [Abbasi e Younis 2007] Abbasi, A. A. e Younis, M. (2007). A survey on clustering algorithms for wireless sensor networks. *Comput. Commun.*, 30(14-15):2826–2841.
- [Abdul Latiff et al. 2007] Abdul Latiff, N., Tsimenidis, C., e Sharif, B. (2007). Performance comparison of optimization algorithms for clustering in wireless sensor networks. In *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE International Conference on*, pp. 1–4.
- [Akyildiz et al. 2002] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., e Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422.
- [Almalkawi et al. 2010] Almalkawi, I. T., Guerrero Zapata, M., Al-Karaki, J. N., e Morillo-Pozo, J. (2010). Wireless multimedia sensor networks: Current trends and future directions. *Sensors*, 10(7):6662–6717.
- [Aoun e Boutaba 2006] Aoun, B. e Boutaba, R. (2006). Clustering in wsn with latency and energy consumption constraints. *Journal of Network and Systems Management*, 14(3):415–439.
- [Baronti et al. 2007] Baronti, P., Pillai, P., Chook, V. W. C., Chessa, S., Gotta, A., e Hu, Y.-F. (2007). Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards. *Computer Communications*, 30(7):1655–1695.
- [Basagni et al. 2007] Basagni, S., Carosi, A., e Petrioli, C. (2007). Controlled vs. uncontrolled mobility in wireless sensor networks: Some performance insights. In *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pp. 269–273.
- [Chatzigiannakis et al. 2006] Chatzigiannakis, I., Kinalis, A., e Nikolettseas, S. (2006). Sink mobility protocols for data collection in wireless sensor networks. In *Proceedings of the 4th ACM International Workshop on Mobility Management and Wireless Access, MobiWac '06*, pp. 52–59, New York, NY, USA. ACM.
- [Costa 2013] Costa, D. G. (2013). *Otimizações da Transmissão de Imagens em Redes de Sensores Visuais Sem Fio Explorando a Relevância de Monitoramento dos Nós Fontes e Codificação DWT*. PhD thesis, Universidade Federal do Rio Grande do Norte.

- [Costa e Guedes 2012] Costa, D. G. e Guedes, L. A. (2012). A survey on transport protocols for wireless multimedia sensor networks. *TIIS*, 6:241–269.
- [Costa et al. 2013] Costa, D. G., Guedes, L. A., Vasques, F., e Portugal, P. (2013). Adaptive monitoring relevance in camera networks for critical surveillance applications. *IJDSN*, 2013:15.
- [Costa et al. 2014] Costa, D. G., Guedes, L. A., Vasques, F., e Portugal, P. (2014). Relevance-based balanced sink mobility in wireless visual sensor networks. In *INDIN*.
- [Costa et al. 2015] Costa, D. G., Guedes, L. A., Vasques, F., e Portugal, P. (2015). Research trends in wireless visual sensor networks when exploiting prioritization. *Sensors*, 15(1):1760–1784.
- [Dai e Wu 2005] Dai, F. e Wu, J. (2005). On constructing k-connected k-dominating set in wireless networks. In *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*, pp. 81a–81a.
- [Das e Dutta 2005] Das, A. e Dutta, D. (2005). Data acquisition in multiple-sink sensor networks. *Mobile Computing and Communications Review*, 9(3):82–85.
- [Dijkstra 1959] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.
- [Enz et al. 2004] Enz, C., El-Hoiydi, A., Decotignie, J.-D., e Peiris, V. (2004). Wisenet: an ultralow-power wireless sensor network solution. *Computer*, 37(8):62–70.
- [Friedmann e Boukhatem 2007] Friedmann, L. e Boukhatem, L. (2007). Efficient multi-sink relocation in wireless sensor network. In *ICNS*, pp. 90. IEEE Computer Society.
- [Gao et al. 2009] Gao, S., Zhang, H., e Das, S. (2009). Efficient data collection in wireless sensor networks with path-constrained mobile sinks. In *World of Wireless, Mobile and Multimedia Networks Workshops, 2009. WoWMoM 2009. IEEE International Symposium on a*, pp. 1–9.
- [Hamerly e Elkan 2002] Hamerly, G. e Elkan, C. (2002). Alternatives to the k-means algorithm that find better clusterings. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management, CIKM '02*, pp. 600–607, New York, NY, USA. ACM.
- [Hartenstein e Laberteaux 2008] Hartenstein, H. e Laberteaux, K. (2008). A tutorial survey on vehicular ad hoc networks. *IEEE Communications Magazine*, 46(6):164–171.
- [Hartigan 1975] Hartigan, J. A. (1975). *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 99th edition.
- [Heinzelman et al. 2002] Heinzelman, W., Chandrakasan, A., e Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *Wireless Communications, IEEE Transactions on*, 1(4):660–670.

- [Iyer et al. 2005] Iyer, Y. G., Gandham, S., e Venkatesan, S. (2005). Stcp: a generic transport layer protocol for wireless sensor networks. In *ICCCN*, pp. 449–454. IEEE.
- [Khan et al. 2013] Khan, M. I., Gansterer, W. N., e Haring, G. (2013). Static vs. mobile sink: The influence of basic parameters on energy efficiency in wireless sensor networks. *Computer Communications*, 36(9):965 – 978. Reactive wireless sensor networks.
- [Kim et al. 2005] Kim, Y.-J., Govindan, R., Karp, B., e Shenker, S. (2005). Geographic routing made practical. In Vahdat, A. e Wetherall, D., editors, *Proceedings of the 2nd Conference on Symposium on Networked Systems Design and Implementation - Volume 2*, NSDI'05, pp. 217–230, Berkeley, CA, USA. USENIX Association.
- [Koç e Korpeoglu 2014] Koç, M. e Korpeoglu, I. (2014). Controlled sink mobility algorithms for wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2014:12.
- [Krishnamurthy et al. 2005] Krishnamurthy, L., Adler, R., Buonadonna, P., Chhabra, J., Flanigan, M., Kushalnagar, N., Nachman, L., e Yarvis, M. D. (2005). Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea. In [Redi et al. 2005], pp. 64–75.
- [Kulkarni et al. 2005] Kulkarni, P., Ganesan, D., Shenoy, P. J., e Lu, Q. (2005). *SensEye*: a multi-tier camera sensor network. In Zhang, H., Chua, T., Steinmetz, R., Kankanhalli, M. S., e Wilcox, L., editors, *Proceedings of the 13th ACM International Conference on Multimedia, Singapore, November 6-11, 2005*, pp. 229–238. ACM.
- [Lee e Jung 2010a] Lee, J.-H. e Jung, I.-B. (2010a). Adaptive-compression based congestion control technique for wireless sensor networks. *Sensors*, 10(4):2919–2945.
- [Lee e Jung 2010b] Lee, J.-H. e Jung, I.-B. (2010b). Reliable asynchronous image transfer protocol in wireless multimedia sensor networks. *Sensors*, 10(3):1486–1510.
- [Lin et al. 2004] Lin, P., Qiao, C., e Wang, X. (2004). Medium access control with a dynamic duty cycle for sensor networks. In *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, volume 3, pp. 1534–1539 Vol.3. IEEE.
- [Marta e Cardei 2009] Marta, M. e Cardei, M. (2009). Improved sensor network lifetime with multiple mobile sinks. *Pervasive and Mobile Computing*, 5(5):542 – 555.
- [Mateu et al. 2006] Mateu, L., Codrea, C., Lucas, N., Pollak, M., e Spies, P. (2006). Energy harvesting for wireless communication systems using thermogenerators. In *XXI Conference on Design of Circuits and Integrated Systems (DCIS)*.

- [Meng et al. 2007] Meng, M., Wu, X., Jeong, B.-S., Lee, S., e Lee, Y.-K. (2007). Energy efficient routing in multiple sink sensor networks. In *Computational Science and its Applications, 2007. ICCSA 2007. International Conference on*, pp. 561–566.
- [Peixoto e Costa 2015] Peixoto, J. P. J. e Costa, D. G. (2015). Qoe-aware multiple sinks mobility in wireless sensor networks. In *New Technologies, Mobility and Security (NTMS), 2015 7th International Conference on*, pp. 1–4.
- [Peña et al. 1999] Peña, J., Lozano, J., e Larrañaga, P. (1999). An empirical comparison of four initialization methods for the k-means algorithm. *Pattern Recognition Letters*, 20(10):1027 – 1040.
- [Raghunathan et al. 2005] Raghunathan, V., Kansal, A., Hsu, J., Friedman, J., e Srivastava, M. B. (2005). Design considerations for solar energy harvesting wireless embedded systems. In *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks, IPSN 2005, April 25-27, 2005, UCLA, Los Angeles, California, USA*, pp. 457–462. IEEE.
- [Rajagopalan e Varshney 2006] Rajagopalan, R. e Varshney, P. K. (2006). Data-aggregation techniques in sensor networks: A survey. *IEEE Communications Surveys and Tutorials*, 8(1-4):48–63.
- [Redi et al. 2005] Redi, J., Balakrishnan, H., e Zhao, F., editors (2005). *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems, SenSys 2005, San Diego, California, USA, November 2-4, 2005*. ACM.
- [Saad et al. 2009] Saad, E. M., Awadalla, M. H., e Darwish, R. R. (2009). Adaptive energy-aware gathering strategy for wireless sensor networks. *IJDSN*, 5(6):834–849.
- [Seah et al. 2009] Seah, W., Eu, Z. A., e Tan, H. (2009). Wireless sensor networks powered by ambient energy harvesting (wsn-heap) - survey and challenges. In *Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology, 2009. Wireless VITAE 2009. 1st International Conference on*, pp. 1–5.
- [Simon et al. 2004] Simon, G., Maróti, M., Lédeczi, A., Balogh, G., Kussy, B., Nádas, A., Pap, G., Sallai, J., e Frampton, K. (2004). Sensor network-based countersniper system. In Stankovic, J. A., Arora, A., e Govindan, R., editors, *SenSys*, pp. 1–12. ACM.
- [Tolle et al. 2005] Tolle, G., Polastre, J., Szewczyk, R., Culler, D. E., Turner, N., Tu, K., Burgess, S., Dawson, T., Buonadonna, P., Gay, D., e Hong, W. (2005). A macroscope in the redwoods. In [Redi et al. 2005], pp. 51–63.
- [Ye et al. 2004] Ye, W., Heidemann, J. S., e Estrin, D. (2004). Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 12(3):493–506.

- [Yick et al. 2008] Yick, J., Mukherjee, B., e Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330.
- [Younis et al. 2003] Younis, M., Akkaya, K., e Kunjithapatham, A. (2003). Optimization of task allocation in a cluster-based sensor network. In *Computers and Communication, 2003. (ISCC 2003). Proceedings. Eighth IEEE International Symposium on*, pp. 329–334 vol.1.
- [Zhang et al. 2008] Zhang, L., Hauswirth, M., Shu, L., Zhou, Z., Reynolds, V., e Han, G. (2008). Multi-priority multi-path selection for video streaming in wireless multimedia sensor networks. In Sandnes, F., Zhang, Y., Rong, C., Yang, L., e Ma, J., editors, *Ubiquitous Intelligence and Computing*, volume 5061 of *Lecture Notes in Computer Science*, pp. 439–452. Springer Berlin Heidelberg.
- [Zhang et al. 2006] Zhang, R., Zhao, H., e Labrador, M. A. (2006). The anchor location service (als) protocol for large-scale wireless sensor networks. In Chlamtac, I., editor, *InterSense*, volume 138 of *ACM International Conference Proceeding Series*, pp. 18. ACM.