

Camila Helena Souza Oliveira

**Gerenciamento Autônomo de Energia em Redes de Sensores
Sem Fio Através do Escalonamento de Atividade dos Nós**

Dissertação de Mestrado submetida à Coordenação do MDCC - Mestrado e Doutorado em Ciência da Computação da Universidade Federal do Ceará como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Miguel Franklin de Castro, PhD.

Fortaleza - CE

Novembro - 2011

Dedico esta dissertação
ao meu esposo, meu Pai (in memoriam),
minha Mãe e minhas irmãs.

Agradecimentos

A realização deste trabalho não teria sido possível sem a contribuição de muitas pessoas. Por isso, deixo aqui registrado o reconhecimento sincero pela contribuição recebida de todos que me ajudaram:

Primeiramente agradeço ao meu bom Deus, que com toda sua graça e misericórdia me escolheu para ser uma pessoa abençoada me dotando de perfeita saúde, física e mental, embora algumas pessoas duvidem desta última =), e principalmente da capacidade de me cercar de pessoas especiais que fazem com que tudo na minha vida pareça mais fácil, mesmo que ela seja recheada de dificuldade e desafios. Obrigada Deus por permitir a realização de mais um sonho!

Ao meu amado esposo, Nazareno Mateus, que além de todo incentivo, compreensão e amor, a mim dedicados durante todo período do mestrado, foi quem me deu condições para que eu pudesse me dedicar exclusivamente à realização dos meus sonhos, sacrificando, por todo esse tempo, a conquista de seus próprios objetivos. Por toda dedicação e empenho em querer me fazer cada vez mais feliz, sem saber que, na verdade, tê-lo ao meu lado já é o bastante para minha felicidade. Ao meu amado, todo meu amor e eterna gratidão.

À minha família que sempre me apoiou e incentivou a seguir o caminho que escolhi. Em especial, à minha irmã Rafaela, que se faz sempre presente na minha vida me ajudando em tudo que preciso.

Ao meu orientador, Miguel Franklin, por acreditar no meu potencial e contribuir com sua experiência e conhecimento nos momentos decisivos. Ele, com toda sua paciência e leveza diante das dificuldades, conseguiu me trazer equilíbrio e confiança, e, não levando a sério as bobagens ditas por mim, permitiu que fosse estabelecido um canal aberto de comunicação entre nós, sem o qual, teria sido muita mais difícil concluir este trabalho.

Ao Stênio Fernandes, professor que me recebeu e auxiliou com atenção durante o estágio realizado no Canadá, por ter dedicado seu tempo e colaborado com seu conhecimento durante a fase inicial do meu trabalho. Sua conduta e dedicação à pesquisa tornaram-se exemplo a ser seguido por mim.

Ao querido amigo e professor Carlos Brito que me acompanha desde a graduação se fa-

zendo presente nos momentos mais importantes desta caminhada. Seu apoio, conselhos e incentivo marcaram minha vida de uma forma tão intensa e transformadora que talvez nem ele saiba. Ele me mostrou que a vida é muito mais do que essa roda viva na qual ficamos presos, me ajudou a sair da zona de conforto e explorar novas possibilidades. Ele é mais que um exemplo, é uma inspiração, é o tipo de professor que eu quero ser.

Às minhas amadíssimas amigas Karol e Cibele que estão sempre ao meu lado e nunca me deixam desanimar. Elas não me oferecem apenas um ombro amigo, ou enxugam minhas lágrimas, elas me ajudam a carregar o fardo que muitas vezes a vida nos impõe, por isso, tudo fica mais fácil com elas ao meu lado.

Aos meus amigos Bruno Góis, Michel Sales e Carlos Giovanni por todo apoio que me deram. Em especial, ao Bruno Góis, que esteve sempre disponível para me ajudar em tudo que precisei durante o ultimo ano do mestrado, desde problemas de implementação a discussões da solução proposta. Sua ajuda facilitou e possibilitou a resolução de muitos problemas com maior rapidez.

Ao Levi Ribeiro, autor do protocolo que serviu de inspiração para o trabalho desta dissertação, por ter disponibilizado os códigos fontes do seu protocolo. Além disso, por ter se disponibilizado a tirar minhas dúvidas sempre que foi preciso, facilitando o entendimento da sua implementação.

Aos demais colegas do GREat que contribuíram de alguma forma para conclusão deste trabalho. Ao suporte físico oferecido pelo GREat e por sua equipe de profissionais que fazem sempre o melhor para facilitar a vida dos estudantes de mestrado e doutorado.

Finalmente à CAPES pelo financiamento da bolsa de estudos para realização deste trabalho.

A satisfaão est no esforo e no apenas na realizao final.

Mahatma Gandhi.

Resumo

A evolução no desenvolvimento de novos dispositivos, cada vez mais baratos e eficientes, expandiu o uso das Redes Sensores Sem Fio (RSSF) e incentivou a criação de novas aplicações, no cenário contemporâneo da Computação Ubíqua e Pervasiva. A limitação de energia, no entanto, continua sendo um desafio na área de RSSF. Essa situação é agravada ainda mais pela inviabilidade de recarga de energia, já que, em muitos casos, as RSSF são utilizadas em ambientes inacessíveis. Com o barateamento dos dispositivos empregados nas RSSF, ficou mais fácil empregar redes muito densas e de larga escala nos ambientes a serem monitorados. O emprego de redes densas, que apresentam elevado grau de redundância dos nós, permite que a rede continue funcional mesmo com a exaustão de alguns nós. Além de oferecer tolerância a falhas, o uso de redes muito densas oferece a oportunidade da implementação de mecanismos de escalonamento dos nós redundantes, de forma que o tempo de vida da rede seja ainda melhor otimizado. Assumindo um cenário com rede muito densas, este trabalho descreve a implementação de um mecanismo de escalonamento autônomo, simples, robusto e escalável, com o objetivo de melhorar ainda mais os resultados já apresentados pelo BiO4SeL, um protocolo de roteamento baseado em Colônia de Formigas e desenvolvido para maximizar o tempo de vida da rede. Os resultados mostram que o novo esquema de escalonamento, efetivamente, melhora o tempo de vida de uma RSSF baseada no BiO4SeL em cenários densos.

Palavras-chave: Redes de Sensores Sem Fio. Otimização do tempo de vida da rede. Mecanismo de escalonamento. Autonomia. Protocolo BiO4SeL.

Abstract

The evolution and development of new devices, increasingly cheaper and more efficient, expanded the use of Wireless Sensor Networks (WSN) and encouraged the creation of new applications in the contemporary scenery of Ubiquitous and Pervasive Computing. However, energy limitation remains a challenge in the field of WSN. This situation is aggravated even more by the infeasibility of energy recharge since, in many cases, WSN are used in inaccessible environments. With cheapness devices used in WSN, became easier to employ dense and large-scale networks in environments that will be monitored. The use of dense networks, which have a high degree redundancy of nodes, allows the network remains functional even with the exhaustion of some nodes. In addition to provide fault tolerance, the use of very dense networks offer the opportunity of implementing scheduling mechanisms for redundant nodes, in a way that the network lifetime is even better optimized. Assuming a scenery with very dense networks, this dissertation describes the implementation of an autonomic scheduling mechanism, simple, robust and scalable, in order to further improve the results already presented by BiO4SeL, which is a routing protocol based on Ant Colony and designed to maximize the network lifetime. The results show that the new scheduling scheme effectively improves the WSN lifetime based on BiO4SeL in dense scenarios.

Keywords: Wireless Sensor Network , Optimization of the Network lifetime, Scheduling mechanisms, Autonomy, BiO4SeL protocol.

Lista de Figuras

2.1	Rede com a formação de grupos.	p. 27
2.2	Roteamento Multi-hop.	p. 32
3.1	<i>Feedback</i> positivo.	p. 44
5.1	Diagrama de Estados.	p. 68
5.2	Escolha do nó substituto.	p. 70
5.3	Mensagem <i>UpdateStatus</i>	p. 72
6.1	Comparação entre os tempos de vida dos protocolos.	p. 80
6.2	Média de bateria na rede	p. 82
6.3	Distribuição energética na rede, no tempo de vida	p. 84
6.4	Distribuição energética na rede, no tempo de vida, continuação	p. 85
6.5	Coefficiente de variação da média energética pelo tempo	p. 86
6.6	Histograma de envio de pacotes de sinalização	p. 87
6.7	A taxa de entrega de pacotes.	p. 88
6.8	Quantidade de nós ativos pelo tempo.	p. 89
6.9	Energia pela distância ao destino.	p. 90

Lista de Tabelas

3.1	Cobertura garantida com diferentes números de vizinhos. Adaptada de (MENG et al., 2010).	p. 49
6.1	<i>Cenários utilizados nos testes</i>	p. 77
6.2	<i>Configurações do NS.</i> (RIBEIRO, 2009).	p. 79
6.3	<i>Parâmetros do protocolo.</i> Adaptada de (RIBEIRO, 2009).	p. 80

Lista de Algoritmos

- 1 Fase de troca de dados do BiO4SeL com mecanismo de escalonamento BiO4seL v2 e v3. (RIBEIRO; CASTRO, 2010a) p.75

Lista de Abreviaturas e Siglas

AP	<i>Access Point</i>
ARN	<i>Active Routing Nodes</i>
ANR	<i>Active Node Request</i>
AS	<i>Adaptive Sleeping</i>
ADV	<i>Advertisement</i>
AODV	<i>Ad-hoc On Demand Distance Vector Routing</i>
AOER	<i>Ant-based On-demand Energy Routing</i>
ARAMA	<i>Ant Routing Algorithm for Mobile Ad-hoc networks</i>
AIS	<i>Artificial Immune Systems</i>
BT	<i>Beacon Time</i>
BiO4SeL	<i>Biologically-inspired Optimization for Sensor network Lifetime</i>
broadId	<i>broadcast Identifier</i>
CT	<i>Checking Time</i>
CH	<i>Cluster Head</i>
CI	<i>Computational Intelligence</i>
CBR	<i>Constant Bit Rate</i>
DD	<i>Directed Diffusion</i>
NSGA-II	<i>Elitism Non-dominated Sorting Genetic Algorithm</i>
GA	<i>Genetic Algorithm</i>
GEAR	<i>Geographical and Energy Aware Routing</i>
GPS	<i>Global Positioning System</i>
IEA	<i>Interference and Energy Aware topology control</i>
iaTp	<i>initial ant' Type of packet</i>
InitEn	<i>Initial Energy</i>
iant	<i>initialization ant</i>
iHello	<i>initialization Hello</i>
LDAS	<i>Lightweight Deployment-Aware Scheduling</i>
LEACH	<i>Low-Energy Adaptive Clustering Hierarchy</i>
MAC	<i>Media Access Control</i>
NBR	<i>neighbour</i>

negAnt	<i>negative Ant</i>
NS-2	<i>Network Simulator</i>
NSGA	<i>Non-dominated Sorting Genetic Algorithm</i>
pbAnt	<i>piggyback Ant</i>
PEAS	<i>Probing Environment and Adaptive Sleeping</i>
PE	<i>Probing Environment</i>
PEGASIS	<i>Power-Efficient GAthering in Sensor Information Systems</i>
RNA	<i>Redes Neurais Artificiais</i>
RSSF	<i>Redes Sensores Sem Fio</i>
REQ	<i>Request</i>
RequestHello	<i>Request Initialization Hello</i>
respReqHello	<i>response Request Initialization Hello</i>
SMAC	<i>Sensor Media Access Control</i>
SPIN	<i>Sensor Protocols for Information via Negotiation</i>
TT	<i>Ticket Threshold</i>
ST	<i>Sleeping Time</i>
ToCMA	<i>Topology Control using Memetic Algorithm</i>

Sumário

1	Introdução	p. 16
1.1	Contextualização e Motivação	p. 16
1.2	Objetivos	p. 18
1.2.1	Objetivo Geral	p. 18
1.2.2	Objetivos Específicos	p. 18
1.3	Estrutura da Dissertação	p. 18
2	Redes Sensores Sem Fio	p. 20
2.1	Aplicações de RSSF	p. 22
2.1.1	Aplicações Militares	p. 22
2.1.2	Aplicações Ambientais	p. 23
2.1.3	Aplicações na Saúde	p. 23
2.1.4	Outras Aplicações	p. 24
2.2	Características	p. 24
2.3	Desafios	p. 27
2.3.1	Controle de Topologia	p. 28
2.3.2	Redução da Quantidade de Informação Transmitida	p. 30
2.3.3	Roteamento Eficiente em Energia	p. 31
2.3.4	Escalonamento dos Nós	p. 35
3	Trabalhos Relacionados	p. 41
3.1	Roteamento Eficiente em Energia	p. 41

3.1.1	Protocolos Baseados em Colônia de Formigas	p. 43
3.2	Escalonamento dos Nós	p. 47
3.2.1	Soluções de Escalonamento Baseadas no Número Máximo de Con- juntos de Cobertura	p. 47
3.2.2	Soluções de Escalonamento Baseadas na Densidade de Vizinhos de Cada Nó	p. 49
3.2.3	Soluções de Escalonamento Preocupadas Com a Atividade de Rote- amento	p. 53
4	BiO4SeL	p. 57
5	Mecanismo de Escalonamento no BiO4SeL	p. 64
5.1	Descrição do Problema	p. 65
5.2	BiO4SeL v2 e BiO4SeL v3	p. 67
6	Resultados e Discussões	p. 76
6.1	Especificação dos experimentos	p. 76
6.2	Parâmetros da Simulação	p. 77
6.2.1	Parâmetros do NS	p. 77
6.2.2	Parâmetros dos Protocolos	p. 79
6.3	Resultados	p. 79
6.3.1	Experimento 1: Tempo de Vida pela Quantidade de Nós	p. 80
6.3.2	Experimento 2: Média de Bateria na Rede	p. 81
6.3.3	Experimento 3: Distribuição Energética no Tempo de Vida da Rede	p. 82
6.3.4	Experimento 4: Coeficiente de Variação da Média Energética	p. 83
6.3.5	Experimento 5: Pacotes de Sinalização pelo Tempo	p. 85
6.3.6	Experimento 6: Taxa de Entrega pela Quantidade de Nós	p. 86
6.3.7	Experimento 7: Quantidade de Nós Ativos pelo Tempo de Simulação	p. 88
6.3.8	Experimento 8: Energia dos Nós pelos Saltos à Estação Base	p. 89

7 Conclusão e Trabalhos Futuros	p.91
7.1 Conclusões	p.91
7.2 Contribuições	p.92
7.3 Trabalhos Futuros	p.93
Referências Bibliográficas	p.94

1 *Introdução*

Esta dissertação apresenta a implementação de um mecanismo de gerenciamento de energia autônomo, realizado através do escalonamento de atividades dos nós, no protocolo BiO4SeL. O *Biologically-inspired Optimization for Sensor network Lifetime* (BiO4SeL) é um protocolo de roteamento baseado em Formigas, utilizado para maximizar o tempo de vida da rede.

Na Seção 1.1, são mostrados a contextualização e a motivação do trabalho desenvolvido nesta dissertação. Os objetivos do trabalho são apresentados na Seção 1.2 e, para finalizar, na Seção 1.3, é expressa a organização do restante do texto.

1.1 Contextualização e Motivação

A evolução tecnológica que ocorreu nos últimos anos fez com que o mundo moderno entrasse de vez na era digital dominada pelos *smartphones*, *tablets*, computadores superpotentes e dispositivos cada vez mais inteligentes. Esses dispositivos são utilizados para interagir de forma colaborativa e discreta e assim auxiliar as pessoas na execução de suas tarefas, facilitando suas vidas, de forma que elas não tenham que fazer nenhum esforço para isso.

Com dispositivos mais modernos sendo desenvolvidos, tornou-se popular o uso de redes sem fio, tanto em ambientes de trabalho como em ambiências pessoais. Empresas, escolas, universidades, hotéis e casas aderiram ao uso das redes sem fio, pela facilidade de instalação e comodidade no uso dos dispositivos, pois, com o uso das placas de redes sem fio, que empregam rádio frequência como meio de comunicação, os problemas e inconveniências gerados pelo uso de cabeamento foram resolvidos.

Dentre as redes sem fio que surgiram com o advento da tecnologia *wireless*, as Redes Sensores Sem Fio (RSSF) são um tipo de rede utilizada para realizar o monitoramento de ambientes e fazer com que as informações monitoradas cheguem até uma estação-base, responsável por coletar essas informações. Essas redes são formadas por pequenos dispositivos que apresentam características peculiares, como limitação no poder de processamento e na energia disponível.

O protocolo Bio4sel foi proposto em (RIBEIRO; CASTRO, 2010b) como uma solução de roteamento autônomo em RSSF. O algoritmo aplica o conceito de colônia de formigas para maximizar o tempo de vida da rede quando comparado a protocolos como *Ad-hoc On Demand Distance Vector Routing* (AODV) e *Ant Routing Algorithm for Mobile Ad-hoc networks* (ARAMA). Com o uso das formigas e da definição de feromônio, o BiO4SeL consegue aumentar o tempo de vida da rede e se adaptar às constantes mudanças de topologia características das RSSF. O algoritmo, no entanto, ainda se limita ao uso dos nós que circundam o caminho mais curto entre a fonte e a estação base. Apesar de incentivar a exploração de novos caminhos, o acúmulo de feromônio depositado pelas formigas ao longo dos melhores caminhos faz com que os nós situados no melhor caminho morram quando ainda existem muitos nós, com muita energia, em sua vizinhança.

Mecanismos de escalonamento de atividade dos nós estão sendo utilizados com o objetivo de maximizar o tempo de vida de redes densas, mediante o revezamento dos nós que ficam ativos e dos nós que dormem. Na literatura, muitas técnicas estão sendo propostas para realizar a escolha desses nós de forma otimizada e levando em consideração questões como conectividade e cobertura da rede. Algumas propostas garantem cobertura total da rede e outras apenas parcial, dependendo do objetivo da rede e da aplicação para a qual está sendo projetada. A maioria das técnicas utilizadas para garantir cobertura exige conhecimento global da rede, inclusive informações de posicionamento geográfico, como foi proposto em (TIAN; GEORGANAS, 2003).

Como exposto anteriormente, o BiO4SeL é uma proposta autônoma, em que as rotas são estabelecidas dinamicamente com o encaminhamento de formigas, o que permite fácil adaptação às mudanças de topologia e elimina a sobrecarga de manutenção de informações de caminhos e de todos os nós da rede. No Bio4sel não são utilizadas informações de localização, os nós guardam informações apenas de seus vizinhos e ele foi simulado apenas em ambientes de redes esparsas, onde cada nó possui um número pequeno de vizinhos. Como os mecanismos de escalonamento de atividades dos nós são desenvolvidos para serem implementados em redes muito densas, onde cada nó apresenta um número elevado de vizinhos, foram realizados testes do funcionamento do BiO4SeL usando cenários muito densos. Como o BiO4SeL continua desempenhando bem suas funções nesse novo cenário, a proposta deste trabalho é estender o BiO4SeL, de forma que o ganho na economia de energia obtido por sua utilização nessas redes seja ainda maior do que os resultados apresentados quando ele foi utilizado em redes esparsas, consoante mostrado em (RIBEIRO, 2009). Para isso, este trabalho apresenta um mecanismo que realiza o gerenciamento de energia da rede de forma autônoma integrado ao protocolo BiO4SeL. O mecanismo proposto faz o escalonamento de atividade dos nós, com o objetivo de maximizar o tempo de vida das redes densas.

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é estender o protocolo BiO4SeL, de forma que sua utilização em redes densas consiga prover um ganho ainda maior no tempo de vida da rede. Um mecanismo que faz o gerenciamento autonômico da energia da rede, através do escalonamento de atividades dos nós, foi integrado ao BiO4SeL para alcançar esse objetivo.

1.2.2 Objetivos Específicos

Para alcançar o objetivo proposto, foram definidos dois objetivos específicos, que serão descritos a seguir.

- Propor um mecanismo de escalonamento de atividades dos nós que funcionasse de forma distribuída e autonômica. Além disso, o mecanismo deve apresentar três características de alto nível, conforme é definido em (WANG; XIAO, 2006), que são: robustez, simplicidade e escalabilidade.
- Implementar e integrar o mecanismo de escalonamento dos nós proposto, no protocolo BiO4SeL, de forma que o gerenciamento autonômico da energia obtenha o ganho de vida desejado.

1.3 Estrutura da Dissertação

Esta dissertação é dividida em sete capítulos. Neste, foram descritos o contexto, a motivação e os objetivos do trabalho. O restante serão definidos na sequência.

Capítulo 2 São apresentadas as definições, principais características e algumas das áreas mais comuns para o emprego das RSSF. Além disso, serão exibidos os pontos que ainda se mostram como desafios em RSSF e as estratégias mais utilizadas para superar esses desafios, segundo (MAHFOUDH; MINET, 2008). Entre essas estratégias, estão duas das áreas de atuação deste texto: roteamento eficiente em energia e escalonamento de atividades dos nós.

Capítulo 3 São indicados os trabalhos relacionados com a proposta desta dissertação. A descrição detalhada de algumas soluções e uma análise crítica dos mecanismos utilizados também fazem parte deste capítulo.

Capítulo 4 Traz uma análise detalhada do protocolo fonte de interesse deste trabalho, o Bio4SeL ((RIBEIRO; CASTRO, 2010a)). O mecanismo utilizado e toda a estrutura implementada pelo autor serão exteriorizados com riqueza de detalhes, para facilitar o entendimento da contribuição proposta neste trabalho.

Capítulo 5 É exposto o mecanismo de escalonamento de atividade dos nós e sua integração no BiO4seL. A proposta foi dividida em duas versões, o BiO4SeL v2 e v3, e suas diferenças também serão descritas neste capítulo.

Capítulo 6 Denotam-se os parâmetros do simulador, o NS-2, e do protocolo BiO4SeL. Além disso, são apontados também os parâmetros acrescentados ao BiO4SeL para implementação do escalonamento BiO4SeL v2 e v3. Para finalizar o capítulo, são exibidos gráficos, que representam os resultados obtidos, e uma análise crítica do desempenho dos mecanismos implementados com base nos objetivos propostos.

Capítulo 7 Delineiam-se as conclusões obtidas após a análise dos resultados apresentados pela proposta deste trabalho. São amostrados, também, as contribuições alcançadas e alguns trabalhos a serem concluídos no futuro próximo.

2 *Redes Sensores Sem Fio*

Atualmente as redes sem fio são bastante utilizadas em ambientes de trabalho e pessoais. O ganho de espaço das redes sem fio em ambientes antes dominados por redes cabeadas trouxe para o mercado a necessidade de desenvolvimento de novas aplicações e dispositivos. Esta expansão e comercialização das rede sem fio resultaram também no surgimentos de importantes e, atuais, tópicos de pesquisa, pois, propiciaram o surgimento de vários desafios, um deles, é o desenvolvimento de novos protocolos que permitam o bom funcionamento das novas aplicações desenvolvidas para as redes sem fios, com relação ao gasto energético.

Para que as redes sem fio continuem crescendo e oferecendo oportunidades de desenvolvimento de novas aplicações, questões como segurança, gerenciamento de mobilidade, energia e autonomia são cruciais e vêm sendo estudadas árduamente no meio acadêmico. As redes Sensores sem Fio (RSSF) são compostas por uma grande quantidade de pequenas unidades de hardware empregadas com o objetivo de monitorar um ambiente. Em razão das dimensões limitadas dos dispositivos usados em RSSF, essas redes apresentam restrições de hardware e bateria. Nos últimos anos, as RSSF vêm sendo empregadas em várias aplicações, dentre as mais conhecidas estão: o monitoramento de reservas florestais, para impedir o alastramento de incêndios, a segurança comercial e residencial, o monitoramento da vida de animais silvestre, a predição de terremotos e erupções vulcânicas e a vigilância em campos de batalha. Embora seja uma tecnologia já bastante difundida, há ainda muitos desafios a serem resolvidos, e a durabilidade da energia do nós é um deles.

As redes móveis sem fio diferem das redes convencionais por utilizarem uma interface de comunicação que utiliza ondas de rádio no lugar dos cabos de redes comuns usados nas redes convencionais. O desenvolvimento das interfaces de redes sem fios e de dispositivos menores e mais modernos fez com que rapidamente essas redes se expandissem. As redes móveis sem fio podem ser classificadas como redes infraestruturada ou redes Ad-Hoc. As redes infraestruturadas possuem um dispositivo chamado de *Access Point* (AP), responsável por toda a comunicação da rede e pelo gerenciamento da mobilidade dos nós móveis. Nas redes Ad-Hoc, os nós não possuem uma infraestrutura que controle a mobilidade e a comunicação entre

eles. Assim, cada nó pode se comunicar diretamente com qualquer nó que esteja no seu raio de alcance e desempenhar a função de roteador.

A RSSF é um tipo de rede Ad-Hoc composta por centenas ou milhares de pequenos dispositivos com severa restrição de energia e poder computacional bastante limitado. Os dispositivos das RSSF são denominados nós e são capazes de monitorar um ambiente, coletar informações requisitadas pela aplicação e encaminhar dados para um nó chamado de estação-base. A estação-base possui uma configuração mais robusta, com maior quantidade de energia disponível, poder de processamento e alcance de transmissão (YICK; MUKHERJEE; GHOSAL, 2008). Além disso, é o responsável por transmitir as informações recebidas para o usuário da rede, que pode estar conectado a ele por uma rede cabeada ou através da internet.

Os elementos básicos que definem a configuração de um nó comum em uma RSSF são: memória, processador, interface de rede sem fio, bateria e sensores. Os sensores podem medir condições físicas, como temperatura, pressão, umidade, velocidade etc. As informações coletadas pelos sensores são utilizadas para caracterizar os objetos e ambientes a serem monitorados. De acordo com os valores monitorados, a aplicação da rede pode identificar condições que disparam a execução de uma ação ou coletar informações compatíveis com comportamentos que precisam ser estudados e analisados. Em razão da grande variedade de sensores disponíveis no mercado, uma RSSF pode ser utilizada na construção de uma grande variedade de aplicações, como, por exemplo, no monitoramento de ambientes físicos, interno e externo, na detecção de intrusão em campos de batalha, em alerta de incêndios nas florestas, na detecção de desastres naturais, como terremotos e tsunamis, na observação de animais selvagens em seu habitat natural, entre outras.

Apesar das RSSF serem um tipo de rede Ad-Hoc, os protocolos para redes Ad-Hoc não podem ser empregados nas RSSF, pois as duas diferem em questões fundamentais. O endereçamento global utilizado nas redes Ad-Hoc não pode ser usado nas RSSF, em decorrência da escalabilidade e carência computacional dos nós sensores. As RSSF utilizam um endereçamento baseado em atributo. Esse tipo de endereçamento preocupa-se apenas com os valores atribuídos aos parâmetros monitorados. Como nas RSSF o interesse maior da rede é obter informações sobre os fenômenos que estão sendo monitorados, ou seja, os atributos que caracterizam esses fenômenos, e a identificação do nó que proverá essa informação é irrelevante, o roteamento baseado em atributo é a escolha mais adequada.

Com relação ao fluxo de comunicação, nas RSSF, ele é estabelecido a partir de muitas fontes em direção a um único nó sorvedouro, enquanto nas redes Ad-Hoc a comunicação é fim a fim. Outra diferença pode ser encontrada também na quantidade de nós que fazem parte da

rede: nas RSSF, é utilizada uma quantidade de nós que chega a ordens de magnitude maiores do que nas redes Ad-Hoc. Outro aspecto importante é que os nós sensores são propensos a falhas e por isso as RSSF são, normalmente, densamente povoadas. Além disso, os protocolos das RSSF têm que lidar com diversos problemas, como grande quantidade de dados redundantes, restrição energética e uso de mensagens *broadcast*, tendo como principal objetivo maximizar o tempo de vida da rede que normalmente não é um problema nas redes Ad-Hoc (JIANG et al., 2006).

A seguir serão discutidas as aplicações de RSSF, a RSSF e suas principais características, o roteamento nas RSSF e os desafios que ainda são uma realidade nas RSSF; e a apresentação foi organizada da seguinte maneira. Na seção 2.1, são descritas as principais aplicações de RSSF. A seção 2.2 traz as características mais importantes das RSSF. Finalmente, para concluir o capítulo, na seção 2.3 os principais desafios das RSSF e o que vem sendo feito para tentar resolvê-los são discutidos.

2.1 Aplicações de RSSF

Em uma RSSF os nós sensores trabalham de forma distribuída e cooperativa para realizar as tarefas de sensoriamento e coleta de informações requisitadas pela aplicação (SALEEM; CARO; FAROOQ, 2011a). Como mencionado na introdução, existe grande variedade de sensores disponíveis no mercado com a capacidade de realizar o sensoriamento das mais diversas condições físicas e ambientais. O rápido avanço no desenvolvimento dos dispositivos eletrônicos e das interfaces de comunicação sem fio possibilitou a redução dos preços dos sensores, a diminuição do tamanho do dispositivo e o aumento da inteligência, estimulando a aplicação de RSSF em setores como segurança, saúde e agricultura.

Akyildiz et al. (2002) classifica as aplicações em grandes áreas, como militares, ambientais, saúde, casa e áreas comerciais. Já em (YICK; MUKHERJEE; GHOSAL, 2008), as aplicações são divididas primeiro quanto a função desempenhada pelos sensores, que pode ser de dois tipos: monitoramento e rastreamento. Dentro dessas duas categorias, as aplicações são divididas em áreas que são basicamente as mesmas definidas em (AKYILDIZ et al., 2002b).

2.1.1 Aplicações Militares

Os interesses militares de alguns países poderosos, como os EUA, motivaram o desenvolvimento de muitas tecnologias hoje bastante difundidas na comunidade civil. A Internet e as

RSSF são exemplos disso. Dentre as aplicações de RSSF na área militar, estão o monitoramento das condições do terreno e dos veículos militares, detecção de tropas inimigas no campo de batalha, rastreamento dos inimigos na terra ou no mar e detecção de ataques químico e biológico (AKYILDIZ *et al.*, 2002b). Em (KUMAR, 2010), é apresentada uma solução aplicada nas RSSF utilizadas para realizar a coleta de informação sobre tropas inimigas no campo de batalha e transmissão dos dados com alta qualidade para a estação-base.

2.1.2 Aplicações Ambientais

Questões de preservação do meio ambiente, como a luta contra o desmatamento florestal e a economia no consumo de água, auferem cada vez mais adeptos e estimulam o desenvolvimento de tecnologias que possam auxiliar na luta em defesa da natureza. Nos últimos anos, as RSSF tem sido cada vez mais utilizadas no monitoramento de florestas, com o objetivo de detecção de incêndio, rastreamento de animais em seu habitat natural, detecção de desastres naturais, como enchentes e terremotos, previsão meteorológica e melhor aproveitamento de recursos utilizados na agricultura. Em (REHMAN *et al.*, 2011), é mostrado como as RSSF oferecem suporte para o desenvolvimento da agricultura, em termos do uso inteligente de recursos como, água e fertilizantes, detecção e controle de pragas e monitoramento das condições do terreno, como por exemplo, o pH do solo. Kwong *et al.* (2011) mostram que as RSSF também podem ser aplicadas na pecuária, fazendo o monitoramento da movimentação dos animais nos pastos.

2.1.3 Aplicações na Saúde

Com o desenvolvimentos de dispositivos cada vez menores e modernos, o número de aplicações na área da saúde cresceu bastante nos últimos anos. As RSSF podem ser usadas no auxílio a pessoas inválidas, na administração de remédios, no monitoramento de pacientes e informações, como pressão sanguínea, taxa respiratória e fluxo sanguíneo. Além disso, as RSSFS auxiliam nos diagnósticos e na localização de médico e pacientes (AKYILDIZ *et al.*, 2002b; GARCÍA-HERNÁNDEZ PABLO H. IBARGÜENGOYTIA-GONZÁLEZ,). Em (RODRIGUES; PEREIRA; NEVES, 2011), é externada uma solução para conseguir que dados como temperatura interna e externa do corpo, batimentos cardíacos e pressão arterial cheguem de forma compreensível aos médicos e pacientes. Nessa aplicação, os parâmetros que medem as condições de saúde do paciente são coletados através de uma RSSF, utilizada para realizar o monitoramento do paciente 24 horas por dia, sete dias por semana. Outra aplicação de monitoramento de pacientes é descrita em (DILMAGHANI *et al.*, 2011), onde os pacientes são monitorado em suas casas e os parâmetros observados são enviados para o hospital via Internet.

2.1.4 Outras Aplicações

As redes de sensores sem fio também pode ser empregadas no projeto de ambientes inteligentes, como pode ser visto em (BAEG et al., 2007). Nos campos de extração de petróleo, o vazamento de água e gás pode ser evitado com o monitoramento e detecção de sinais feitos por uma RSSF, como mostrado em (YOON et al., 2011). Na indústria do petróleo, o problema de produção de areia, que pode resultar numa erosão significativa nos fluidos produzidos, pode ser solucionado com o uso de RSSF. Em (ABDELGAWAD; BAYOUMI, 2011), uma aplicação de RSSF é definida para detectar a produção de areia e a taxa com que é produzida dentro dos oleodutos. Na indústria de aeronaves, as redes de sensores também são utilizadas com o objetivo de identificar possíveis problemas, com base nos dados coletados em tempo real. Mediante as informações coletadas, acidentes podem ser evitados de forma preventiva e precisa. Além disso, sensores podem ser empregados, por exemplo, nas asas de uma aeronave para ajudar a reduzir a resistência aerodinâmica (BUR; OMIYI; YANG, 2010).

2.2 Características

Como visto, as RSSF podem ser utilizadas nos mais variados ambientes com diversos objetivos diferentes. Por esse motivo, uma característica marcante das RSSF é ser orientada à aplicação. A quantidade de nós que compõem a rede, a forma como os dados serão encaminhados e a periodicidade da coleta de novas informações são definidas de acordo com os objetivos traçados pela aplicação. Além disso, questões como escalabilidade, tolerância a falha, custos de produção, ambiente de operação, topologia da rede, restrições de *hardware*, meio de transmissão e consumo de energia também devem ser levados em consideração no projeto de RSSF (AKYILDIZ et al., 2002a).

O funcionamento de uma RSSF é simples, descentralizado e autônomo. Uma vez que os nós se encontram no ambiente a ser monitorado, os sensores começam a realizar o sensoriamento da área, coletar os dados e encaminhá-los à estação-base. Normalmente, existe apenas uma estação-base numa RSSF, fazendo com que a maioria dos nós esteja localizada a uma distância fora do seu alcance. Como o alcance de cada nó é limitado, para encaminhar os dados até o destino final, estes são transferidos utilizando os nós intermediários em um caminho entre a fonte e a estação-base.

O dinamismo da rede, o posicionamento dos nós, o consumo de energia, os modelos de entrega dos dados, a capacidade dos nós e a agregação de dados são descritos em (AKKAYA; YOUNIS, 2005) como algumas das principais características que devem ser levadas em consi-

deração no desenvolvimento de protocolos e soluções para RSSF. A seguir, discuti-se brevemente cada um.

Dinamismo da Rede A mobilidade pode ser implementada de três formas diferentes: nos nós sensores, na estação-base ou no evento que está sendo monitorado. Na maioria dos trabalhos encontrados na literatura, assume-se o uso de redes estacionárias, por isso são poucos os trabalhos que consideram a mobilidade dos nós sensores (alguns exemplos podem ser visto em (PAPADOPOULOS et al., 2011) e (C.; NASSER, 2006)). Com relação ao evento, a mobilidade depende da aplicação. Por exemplo, no caso do monitoramento de temperatura, as informações que estão sendo coletadas são estacionárias. Já na observação de animais selvagens, o evento a ser monitorado estará quase sempre em constante movimento.

Em certos casos, a mobilidade do nó *sink* é utilizada com o objetivo de aumentar o tempo de vida da rede por meio da distribuição do gasto energético entre os nós, (ALSALIH; HASSANEIN; AKL, 2010), (AZAD; CHOCKALINGAM, 2006). Isso ocorre porque dessa forma o nó *sink* pode se mover para regiões da rede onde os nós possuem mais energia, evitando que os nós com pouca energia residual sejam usados no roteamento dos dados. Além disso, o nó *sink* também pode se mover para perto dos nós fontes, diminuindo o número de nós intermediários entre a fonte e o destino e, como consequência, reduzindo o consumo de energia (YANG; FONOAGE; CARDEI, 2010).

Implantação dos nós A implantação dos nós no ambiente de monitoramento pode ser feita manualmente, sendo a posição de cada nó predeterminada e conhecida, ou os nós podem ser lançados randomicamente, criando uma infraestrutura ad-hoc (AKKAYA; YOUNIS, 2005). O lançamento randômico é o mais utilizado, uma vez que, na maioria dos casos, o ambiente de aplicação das redes sensores é de acesso difícil, como no caso do monitoramento de desastres naturais. No caso de monitoramento industrial, o posicionamento dos nós pode ser feito manualmente, o que permite a escolha de uma boa localização para cada nó. Nesse caso, o roteamento tem a possibilidade de ser realizado da maneira mais econômica possível, em termos de gasto energético.

Modelo de entrega dos dados O modelo de entrega dos dados define a forma que os nós são coletados e encaminhados ao destino. O modelo escolhido, que tem forte influência no algoritmo de roteamento, pode ser orientado a eventos ou consulta, contínuo ou híbrido. No modelo orientado a eventos ou consulta, o dado é transmitido quando algum evento é detectado ou quando o nó *sink* consulta a rede em busca de alguma informação. No modo contínuo, os dados são enviados constantemente pelos nós da rede. No modelo híbrido,

encontra-se uma composição dos outros dois modelos. O modelo híbrido é utilizado para otimizar o uso dos modelos de entrega dos dados, de acordo com as tarefas da aplicação. Dessa forma, o modelo é escolhido consoante a necessidade da rede de transmitir a informação, e tem como consequência a economia de energia.

Capacidade dos nós A capacidade dos nós em uma rede sensor sem fio está relacionada com a função do nó na rede. Algumas RSSF definem uma hierarquia entre os nós para atribuir tarefas diferentes a eles (IWANICKI; STEEN, 2009). Um exemplo onde os nós apresentam capacidade diferentes são as redes que dividem os nós em grupos. Os grupos de nós são liderados por um nó, chamado líder do grupo, que pode desempenhar funções de agregação e/ou fusão dos dados, e é responsável pelo envio dos dados sensorizados para a estação-base.

Nessas redes é possível definir níveis de hierarquia, onde os nós que pertencem a diferentes níveis desempenham diferentes funções. Por exemplo, os nós membros de um grupo desempenham o papel de sensoriar o ambiente, coletar as informações e enviá-las ao líder do seu grupo. Já o nó líder tem a função de receber as informações e transmiti-las para a estação-base ou para um nó líder pertencente a outro nível na hierarquia, como mostrado na Figura 2.1. Além disso, em razão da complexidade das tarefas realizadas pelos nós líderes, em alguns casos, esses nós apresentam capacidade de energia e processamento diferenciada.

Agregação de dados A agregação de dados é um mecanismo utilizado por alguns protocolos para evitar o desperdício de energia com a transmissão de dados redundantes. Os dados redundantes resultam do emprego de redes muito densas, onde uma mesma região é monitorada por vários nós. O uso de grandes quantidades de nós tem a vantagem de ajudar na tolerância a falhas, mas, por outro lado, gera muita informação desnecessária. Para diminuir a quantidade de informação que trafega pela rede, os nós podem se organizar em grupos. Os líderes dos grupos são responsáveis pelo recebimento dos dados de seus membros e por realizarem a agregação dos dados, eliminando as redundâncias e salvando as informações que realmente são importantes. Com o processo de agregação dos dados redundantes, o tamanho e a quantidade de mensagens roteadas são reduzidos, diminuindo também o desperdício de energia, que é um recurso bastante limitado nas RSSF.

Energia A energia é uma questão crucial na utilização das RSSF. Como, dificilmente, as baterias de um nó podem ser recarregadas ou substituídas, a questão energética é sempre levada em consideração em qualquer projeto de RSSF. As soluções e protocolos implementados em RSSF devem ser de baixa complexidade, para que a rede dure por muito

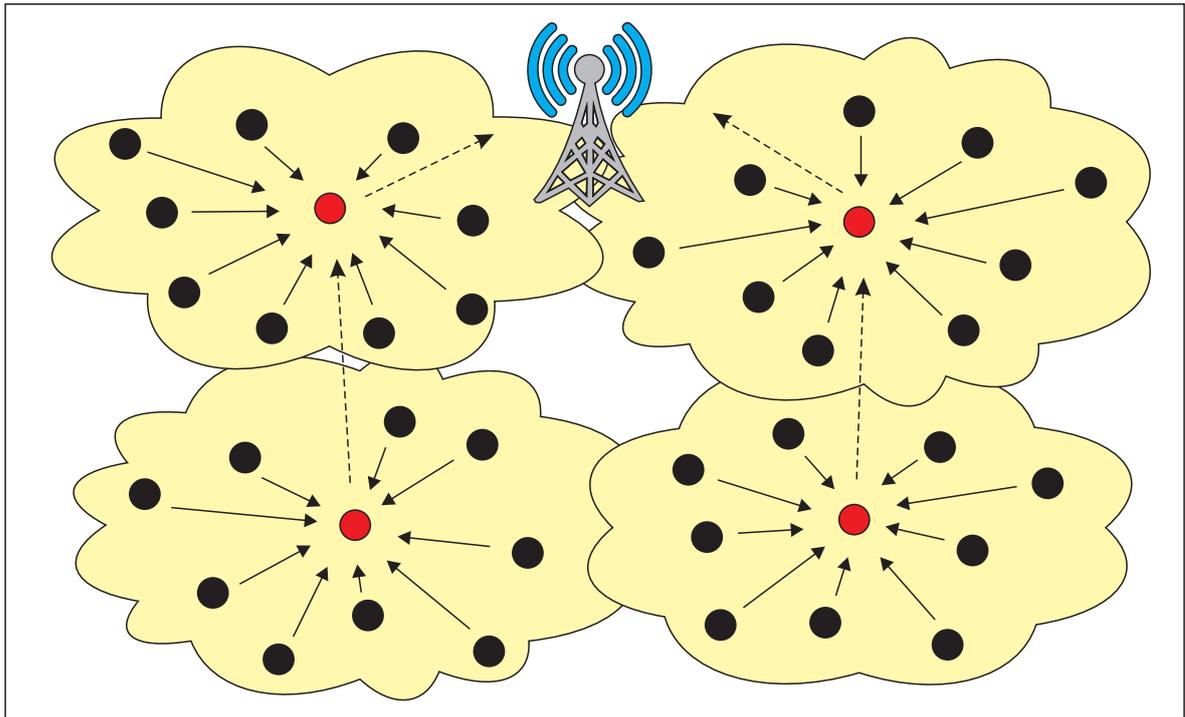


Figura 2.1: Rede com a formação de grupos.

tempo, fazendo bem as tarefas definidas pela aplicação. As questões que mais influenciam no tempo de vida da rede e o que vem sendo desenvolvido para superar esses desafios serão detalhados na Seção 2.3.

2.3 Desafios

A energia e seu consumo de forma eficiente e econômica são os principais desafios quando se trata de RSSF (ZHANG; MAO; ANDERSON, 2010). As características e limitações intrínsecas dessas redes fazem com que a energia seja a maior preocupação no desenvolvimento de projetos e novas soluções. Apesar do avanço no desenvolvimento tecnológico de modernos e eficientes dispositivos, a bateria continua sendo uma fonte de energia limitada que, em razão de os lugares onde geralmente as RSSF são empregadas, não podem ser recarregada ou substituída. Por isso, diferentes mecanismos são utilizados na resolução dos mais diversos problemas enfrentados por uma RSSF com o objetivo de maximizar o tempo de vida da rede com o uso eficiente da energia de cada nó.

Existem várias definições para o tempo de vida de uma RSSF. O tempo de vida da rede pode, por exemplo, ser representado pelo tempo em que pelo menos uma funcionalidade da aplicação deixa ser executada, ou ainda, o tempo da morte de um nó que causa a partição da

rede, impossibilitando a transferência de informação. Outras definições podem ser encontradas em (DIETRICH; DRESSLER, 2009), no entanto, a definição mais utilizada na literatura para o tempo de vida da rede é descrita como o tempo em que o primeiro nó da rede morre. Neste trabalho será utilizada essa definição.

Segundo (MAHFOUDH; MINET, 2008), as estratégias usadas para alcançar o objetivo de aumentar o tempo de vida da rede podem ser classificadas em quatro categorias.

- Controle de topologia.
- Redução da quantidade de informação transmitida.
- Roteamento eficiente em energia.
- Escalonamento dos nós.

Detalhes sobre esses problemas e as soluções que estão sendo estudadas em cada área são discutidas a seguir, no entanto, uma ênfase maior será imprimida nas categorias de Roteamento eficiente em energia e Escalonamento dos nós, pois são os tópicos diretamente relacionados com este trabalho.

2.3.1 Controle de Topologia

As RSSF têm que lidar com severa limitação de energia e ainda cumprir os requisitos das suas aplicações. Para garantir que as funcionalidades das aplicações sejam realizadas com sucesso, os projetos de aplicações e protocolos para RSSF precisam lidar com problemas como cobertura de sensoriamento e conectividade da rede. Existem muitos trabalhos na literatura preocupados em definir um número ótimo de nós que uma rede deve ter para que não tenha um espaço na região a ser monitorada que não esteja coberto pelos nós sensores. Nem todos, porém, estão preocupados em garantir que a quantidade de nós da rede seja suficiente e que possa se organizar de tal forma que todo dado coletado possa chegar até a estação-base.

O uso de redes densas é bastantes comum e apresenta vantagens, como facilidade no controle de sensoriamento e conectividade da rede, além de contribuir para que mecanismos de tolerância à falha sejam implementados. Uma rede com uma grande quantidade de nós, todavia, pode enfrentar problemas como interferência entre os nós, formação de muitas rotas e, dependendo do alcance de transmissão utilizado pelos sensores, pode haver também gasto excessivo de energia com transmissão de informação para nós que estejam muito distantes (MANOLOPOULOS; KATSAROS; PAPADIMITRIOU, 2010). Segundo (ZHANG; MAO; ANDERSON,

2010), quanto maior o poder de transmissão de um nó, maior o gasto energético na transmissão de pacotes. Portanto, para economizar energia, é importante o desenvolvimento de protocolos que realizem os ajustes do poder de transmissão dos nós, mantendo a funcionalidade da rede, ou seja, garantindo sua cobertura e conectividade.

O controle do poder de transmissão é um fator relacionado com o balanceamento do gasto energético e, por consequência, com o aumento do tempo de vida da rede. Ishmanov *et al.* (2010) definem “Controle de Topologia” como uma técnica que realiza o gerenciamento do poder de transmissão de cada nó dinamicamente e de forma que a rede permaneça conectada. Jaewook; Noel; Tang (2010) definem que o grafo de comunicação gerado por um protocolo de controle de topologia deve apresentar três características principais: o grau de cada nó deve ser pequeno, o diâmetro da rede também deve ser pequeno e a topologia dos nós tem que representar um grafo conexo. O grau de um nó é definido como sua quantidade de vizinhos. Com uma topologia definida por um grafo que apresente essas características, uma RSSF apresentará menos interferência e poderá fazer melhor aproveitamento da largura de banda disponível, além de reduzir o consumo de energia por meio de roteamento multi-hop por caminhos curtos. Buscando obter essas vantagens, Jaewook; Noel; Tang (2010) propõem um protocolo, baseado em teoria dos grafos, que gera uma topologia de comunicação para redes de sensores muito densas, onde cada nó tem no máximo grau 4, ou seja, cada nó tem no máximo 4 vizinhos.

Zhang; Zhu (2011) propõem um algoritmo com o objetivo de melhorar os problemas de interferência e diminuir o consumo de energia chamado *Interference and Energy Aware topology control* (IEA). O algoritmo é executado em três fases. Na primeira ocorre a coleta de informação dos vizinhos, como poder de transmissão e distância aproximada entre eles. Na segunda fase, com base nas informações coletadas na primeira fase, o grau de interferência de cada *link* é calculado. Além disso, é utilizado o PRIM, um algoritmo que calcula a árvore geradora mínima de um grafo conexo, para achar uma árvore geradora mínima para cada nó, gerada com diferentes pesos, tamanho e interferência dos *links*. Na terceira fase, a topologia final é gerada baseada nas árvores geradas na fase dois.

Em (KONSTANTINIDIS *et al.*, 2007), uma meta-heurística baseada em Genetic Algorithm (GA) e Busca Local (BL) é usada para manter a conectividade da rede. O algoritmo ToCMA (*Topology Control using Memetic Algorithm*) verifica se a rede está fortemente conectada. Se não estiver, ele executa um processo de reparação, onde uma nova topologia é gerada. Como RSSF apresentam restrições de processamento e energia, o ToCMA é executado *offline*, normalmente, na estação-base, e só depois que a solução é encontrada, ela é disseminada para o restante da rede. Muitas técnicas e heurísticas diferentes são empregadas no controle de to-

pologia. Em (SANTI, 2005), são apresentados os problemas relacionados com o controle de topologia e o estado da arte das soluções propostas para solucioná-los.

2.3.2 Redução da Quantidade de Informação Transmitida

Outra estratégia utilizada para economizar energia é diminuir a quantidade de informação transmitida. O controle dos dados transmitidos pode ser feito de modos diferentes. A agregação e/ou fusão de dados são normalmente as técnicas mais utilizadas para evitar desperdício de energia com transmissões de dados redundantes. Técnicas mais simples, como a disseminação da informação apenas para os nós que tenham interesse, como apresentado em (HEINZELMAN; KULIK; BALAKRISHNAN, 1999), e atualização das informações da rede em intervalos de tempo proporcionais à distância dos nós também podem ser utilizadas, com a finalidade de redução da quantidade de dados transmitida pela rede (MAHFOUDH; MINET, 2008).

Nas redes planas, onde todos os sensores apresentam a mesma configuração e desempenham a mesma função, a difusão do interesse é um dos mecanismos utilizado para realizar agregação. Apesar de haver alguns casos de agregação em redes planas, como mostrado em (RAJAGOPALAN; VARSHNEY, 2006), na maioria dos casos, as técnicas de agregação são utilizadas em redes hierárquicas. Isso acontece porque nas redes planas a disseminação de interesse resulta em comunicação excessiva com a estação-base. Apesar de conseguir diminuir as transmissões nas redes hierárquicas, o ganho de energia obtido pelo mecanismo de agregação nessas redes é influenciado pela performance do processo de agrupamento. Este, por sua vez, é influenciado pelo processo de eleição do líder do grupo e pelos critérios utilizados na formação dos grupos (GHAFFARIYAN, 2010).

Nas redes organizadas em grupos, as técnicas de agregação são executadas nos nós líderes, responsáveis por fazer a fusão dos dados antes de enviá-los à estação-base. Um exemplo de agregação usando o agrupamento dos nós é dado em (HEINZELMAN; CHANDRAKASAN; BALAKRISHNAN, 2000). A fusão dos dados consiste na representação dos dados usando um número menor de *bits* de tal forma que seja possível a reconstrução do dado original na estação-base sem perda de informações críticas. Como um número menor de *bits* é transmitido nas redes com a agregação de dados, uma quantidade significativa de energia é economizada (KASIRAJAN; LARSEN; JAGANNATHAN, 2010).

Apesar de ser a mais utilizada, a organização dos nós em grupos não é a única maneira de realizar agregação de dados em redes hierárquicas, pois outras estruturas, como cadeia e árvore, também são utilizadas. Em (LINDSEY; RAGHAVENDRA, 2002), PEGASIS, um sistema de coleta de dados eficiente em termos de energia, realiza agregação de dados mediante a formação

de uma cadeia de nós onde os nós transmitem(recebem) informação apenas para(de) o vizinho mais próximo. A cada etapa de coleta de dados, os nós recebem a informação, realizam a fusão dos dados recebidos com seus próprios dados e enviam o pacote de dados resultantes para o próximo nó da cadeia. Quando a informação chegar no líder do grupo, que é eleito a cada etapa de coleta de dados, os dados são enviados para a estação-base, assim como acontece no agrupamento.

Quando a estrutura utilizada é uma árvore, a estação-base é representada pela raiz da árvore e o processo de agregação ocorre em todo nó que possui filhos das folhas até a raiz. Em (AVOKH; MIRJALILY, 2010), uma árvore geradora balanceada e dinâmica é utilizada para realizar agregação de dados, levando em consideração o balanceamento do gasto energético pela rede. O balanceamento é obtido por meio da criação, dinamicamente, de uma árvore lógica de roteamento a cada *round*.

Mecanismos biologicamente inspirados também são utilizados na agregação de dados. Em (SUN; CAI; HUANG, 2010), é proposto um esquema de agregação de dados usando redes neurais *back-propagation*. O esquema BPND (Back-Propagation Networks Data Aggregation) é utilizado no protocolo de roteamento que realiza formação de grupos, o LEACH (Low-Energy Adaptive Clustering Hierarchy) (HEINZELMAN; CHANDRAKASAN; BALAKRISHNAN, 2000). Na estrutura formada pelo LEACH, o processamento dos dados originais coletados pelos nós membros do grupo são pré-processados e enviados ao nó líder, onde o processamento através da rede neural *back-propagation* é realizado. Depois da agregação dos dados, o nó líder envia à estação-base as informações úteis extraídas pela rede neural. Em (LIAO; KAO; FAN, 2008) e (WANG; LUO, 2010), algoritmos baseados em colônias de formigas são usados para construir árvores de agregação com base na quantidade feromônica acumulada em um caminho. Um estudo sobre a classificação e as características dos protocolos de agregação mais conhecidos é apresentado em (RAJAGOPALAN; VARSHNEY, 2006).

2.3.3 Roteamento Eficiente em Energia

A pilha de protocolos utilizada pelos nós sensores de uma RSSF é composta pela camada de aplicação, camada de transporte, camada de rede, camada de enlace, camada física, plano de gerenciamento de mobilidade, plano de gerenciamento de energia e o plano de gerenciamento de tarefas (AKYILDIZ et al., 2002a). Todos os protocolos projetados para RSSF devem levar sempre em consideração a restrição energética da rede e procurar desenvolver soluções que evitem o desperdício de energia. Os protocolos da camada física e da camada de enlace são comuns a todas as redes sensores, por isso, o estudo nessa área é concentrado no desenvolvimento de no-

vas interfaces sem fio e protocolos da camada de enlace que levem em consideração o gasto de energia (JIANG et al., 2006). Como a proposta deste trabalho será aplicado no nível de camada de rede, esta seção preocupa-se em discutir apenas questões relacionadas com a funcionalidade dessa camada.

Nos cenários de RSSF, a maioria dos nós encontra-se distante da estação-base, o que torna impossível que todos os dados cheguem até ela por transmissão direta a partir das fontes. Por esse motivo, na maioria das RSSF, é realizado o roteamento *multi-hop* (ZHAO et al., 2010), como exemplificado na Figura 2.2. No encaminhamento *multi-hop*, quando um nó deseja transmitir seus dados, ele deve escolher entre os seus vizinhos aquele que está mais próximo do destino. A vizinhança de um nó é composta pelos nós situados no raio de alcance da interface de comunicação sem fio que, durante o encaminhamento *multi-hop*, são representados como os nós que estão a um salto de distância do nó que está enviando a mensagem.

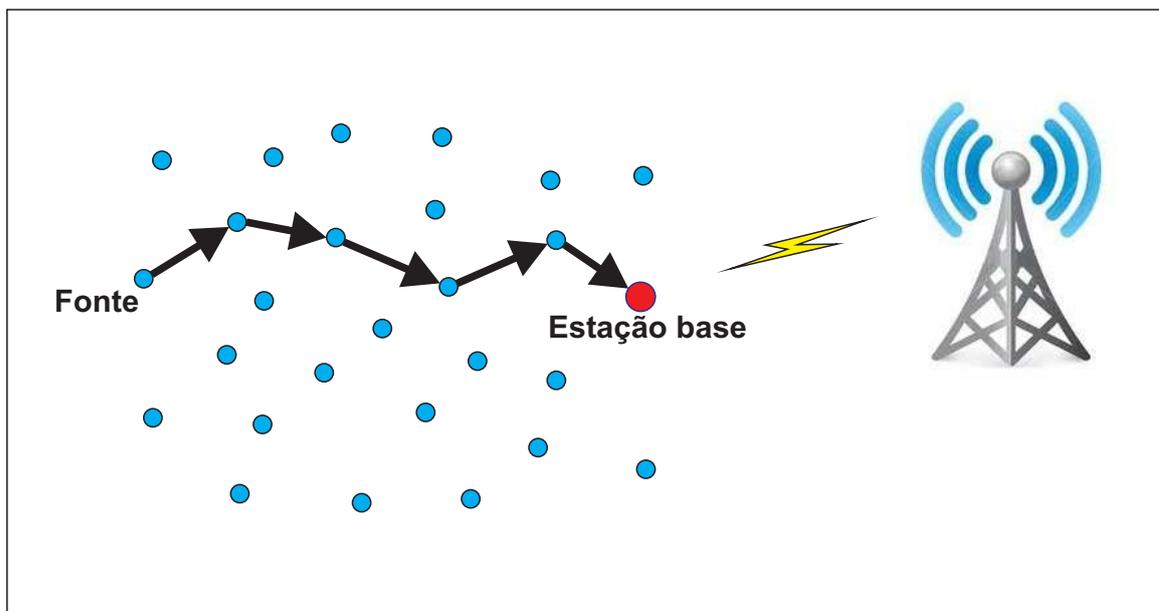


Figura 2.2: Roteamento Multi-hop.

Dentre as tarefas realizadas por um nó, a comunicação, realizada por meio da transmissão e recepção de mensagens, é a que mais consome energia (LOTF; GHAZANI, 2010). Por isso, o principal objetivo dos protocolos desenvolvidos para a camada de rede é realizar a configuração das rotas e entrega dos dados de forma segura e com o menor gasto energético possível. Além do problema do gasto energético, os protocolos de roteamento também devem ser capazes de se adaptar às mudanças que ocorrem na rede, sejam elas causadas pela morte de um nó (no contexto de RSSF, um nó morre quando toda sua energia é consumida) ou por defeitos apresentados nos dispositivos, que são comuns, dada a natureza inóspita dos ambientes onde os nós são distribuídos. Para lidar com tais problemas, os protocolos precisam ser dinâmicos, autônomos,

escaláveis e tolerantes a falhas.

Os primeiros protocolos de roteamento que foram propostos, no entanto, partiram de ideias bastante simples e se preocupavam apenas em fazer os dados chegarem à estação-base. Algumas abordagens usadas para encontrar rotas usavam como critério o caminho mais curto, ou seja, o caminho que tivesse o número mínimo de saltos entre a fonte e a estação-base. Outra opção seria escolher o caminho que gastasse menor quantidade de energia para transferir os dados, ou ainda o caminho que tivesse maior quantidade de energia disponível quando somadas as energias dos nós pertencentes ao caminho. É importante frisar que, se o alcance de transmissão é igual para todos os nós, o gasto de energia na transmissão de um pacote será igual em todo nó. Portanto, nesse caso, o caminho mais curto é também o caminho que gasta menor quantidade de energia para transferir os dados (AKYILDIZ et al., 2002a).

Algumas técnicas utilizadas nos primeiros protocolos de roteamento nas redes sensores foram descritas em (AKYILDIZ et al., 2002a). A técnica mais simples, e também a mais ineficiente, é o *flooding*. No *flooding*, a informação coletada por um nó é enviada por inundação para todos os vizinhos e todo vizinho que recebe a mensagem a retransmite, também por inundação, para todos os seus vizinhos. Um protocolo que faz uso dessa técnica não apresenta custo de descoberta e manutenção de rotas, entretanto, produz intenso gasto energético com as recepções e as transmissões repetidas de informações.

Com o tempo, a preocupação com o gasto energético começou a fazer parte das soluções apresentadas no roteamento das RSSF. Protocolos de negociação e mecanismos de agregação de dados começaram a ser utilizados com o objetivo de economizar energia, reduzindo o número de mensagens redundantes transmitidas. Os protocolos de negociação estabelecem um sistema de troca de mensagens para que as informações coletadas sejam enviadas apenas para os nós que tenham interesse em recebê-las. Já os mecanismos de agregação se utilizam para eliminar a redundância das informações transmitidas.

O *Sensor Protocols for Information via Negotiation* (SPIN) (HEINZELMAN; KULIK; BALAKRISHNAN, 1999) é um exemplo em que um sistema de negociação é usado para economizar energia. Neste protocolo, é definido o conceito de metadados, e são utilizados três tipos de mensagens - ADV, REQ e DATA. Um metadado é uma informação sobre o dado que foi coletado por um nó, que é enviado na mensagem ADV para que os vizinhos possam verificar se a informação coletada é de seu interesse. Com o emprego do metadados, o protocolo consegue economizar energia, uma vez que, a informação é enviada apenas para aqueles que têm interesse recebê-la. O processo de negociação é feito da seguinte maneira. A mensagem ADV é enviada por inundação pelo nó que tem um dado a ser encaminhado contendo a descrição dos

dados coletados (o metadado). O nó vizinho que tiver interesse naquela informação envia uma mensagem de volta, a REQ, apenas para o nó que enviou o metadado. E, quando o nó-fonte recebe a REQ, ele envia a mensagem DATA apenas para os nós que enviaram uma REQ.

A posição geográfica dos nós foi outro fator que passou a ser levado em consideração em algumas soluções com o objetivo de economizar energia. O GEAR (*Geographical and Energy Aware Routing: a recursive data dissemination protocol for wireless sensor networks*), apresentado em (YU; GOVINDAN; ESTRIN, 2001), usa a localização geográfica em duas situações diferentes. A primeira situação é na escolha do próximo nó, onde a localização do nó é incluída no cálculo da heurística utilizada para escolher o nó mais adequado para encaminhamento dos pacotes de dados. E, na segunda, a localização geográfica é utilizada para direcionar a mensagem de interesse enviada da estação-base para a região onde se encontra o evento sobre o qual ela quer receber informações.

Hoje, na literatura, existem muitos protocolos de roteamento, que apresentam diferentes características de acordo com as especificações da aplicação. A maioria deles, no entanto, apresenta uma característica em comum: a preocupação com a energia. Tentando definir uma classificação para os protocolos, vários estudos foram realizados por diferentes autores (AK-KAYA; YOUNIS, 2005; JIANG et al., 2006; AL-KARAKI; KAMAL, 2004). Em alguns casos, como em (LOTF; HOSSEINZADEH; ALGULIEV, 2010), os autores fazem uma classificação simples, levando em consideração apenas a forma como os nós da rede se organizam e no uso de informações de posição geográfica. Segundo essa classificação, os protocolos podem ser:

- planos, onde todos os nós da rede desempenham a mesma função e apresentam igual capacidade;
- hierárquicos, onde ocorrem a formação de grupos e a eleição dos líderes dos grupos; e
- baseados em localização, onde a posição obtida através de GPS é utilizada para estabelecimento e manutenção das rotas.

Nos últimos anos, novos mecanismos utilizando técnicas de inteligência artificial (BARBANCHO et al., 2007) e sistemas biologicamente inspirados (PAONE et al., 2010) são utilizados no roteamento de redes sensores. Os protocolos biologicamente inspirados procuram aplicar comportamentos e estratégias observadas em sistemas biológicos para resolver problemas computacionais. Em (PAVAI; SIVAGAMI; SRIDHARAN, 2009), esses novos protocolos entram na classificação dos protocolos de roteamento que passam a ser divididos em três principais categorias: protocolos biologicamente inspirados, baseados na localização e os hierárquicos.

Dentre os muitos sistemas biológicos existentes, o cérebro humano, o sistema imunológico humano e os insetos que possuem organização social têm inspirado muitos trabalhos na área de RSSF. Esses sistemas são empregados por meio de paradigmas desenvolvidos com suporte em suas principais características. No caso das Redes Neurais Artificiais (RNA), o poder de processamento e a capacidade de aprendizado baseado na experiência faz com que o uso dessas redes seja indicado para solucionar problemas de reconhecimento de padrões, classificação, otimização, entre outros (BASHEER; HAJMEER, 2000). O sistema imunológico artificial ou Artificial Immune Systems (AIS), por sua vez, apresenta-se como um sistema dinâmico, distribuído, autônomo e tolerante ao erro (HOFMEYR; FORREST, 2000). Além disso, é bastante utilizado no domínio de detecção de intrusão de redes (HAAG et al., 2007; KOTOV; VASILYEV, 2009; KOTOV; VASILYEV, 2010).

Colônia de formigas é uma das abordagens baseada nos insetos com organização social e possui basicamente as mesmas características descritas para o AIS. Segundo (SALEEM; CARO; FAROOQ, 2011b), colônia de formigas podem ser utilizadas na solução de problemas que cobrem desde problemas de otimização combinatorial até robótica distribuída. Além disso, é bastante utilizada nos problemas de RSSF, como coleta de dados, detecção de intrusão (AROLKAR; SHETH; TAMHANE, 2011a) e roteamento (GUO; ZHANG; LU, 2010; GUAN et al., 2009; RU; XU, 2010). Soluções biologicamente inspiradas vêm sendo empregadas em muitas áreas ultimamente, como em sistemas computacionais e otimização; no entanto, segundo (DRESSLER; AKAN, 2010), essas soluções aplicadas em problemas de rede, como roteamento e gerenciamento de serviços, ainda têm muitos desafios a serem superados. Apesar desse paradigma já ser um tema bem conhecido e estudado, sua aplicação nas redes sensores é recente.

Como visto, diversas soluções diferentes são aplicadas com o objetivo de realizar o roteamento eficiente em energia e cumprindo requerimentos exigidos pela aplicação. Os trabalhos da literatura partem de suposições diferentes e são aplicados na resolução de problemas específicos. Dessa forma, ainda há muito trabalho a ser feito no desenvolvimento de soluções para o roteamento das RSSF, que maximize o tempo de vida da rede, sem deixar de levar em consideração características como mobilidade, heterogeneidade, tolerância à falhas, e que ainda sejam autônomicas e adaptativas.

2.3.4 Escalonamento dos Nós

Como visto nas subseções anteriores, existem vários mecanismos sendo utilizados para maximizar o tempo de vida da rede. Em redes muito densas, o mecanismo utilizado, normalmente, é o escalonamento dos nós, onde um conjunto de nós é escolhido para permanecer ativo

enquanto os outros ficam no modo dormindo (BULUT; KORPEOGLU, 2011). O escalonamento dos nós busca economizar energia, colocando nós redundantes para dormir; no entanto, o mecanismo deve ser executado de forma que a conectividade da rede e as funcionalidades da aplicação sejam mantidas (MAHFOUDH; MINET, 2008). Os mecanismos de escalonamento são desenvolvidos partindo de premissas diferentes que dependem dos variados contextos das aplicações empregadas na rede. No projeto de tais mecanismos, questões como modelo de detecção, alcance de transmissão, modelo de falha, informação de distância e localização, estrutura e organização da rede, entre outros, devem ser levadas em consideração.

Cada mecanismo de escalonamento possui objetivos específicos determinados pela aplicação, como garantir alta taxa de entrega de dados, escalabilidade, consumo de energia balanceada, robustez, cobertura de sensoriamento e conectividade. Todos apresentam, porém, um objetivo em comum, que é maximizar o tempo de vida da rede. Segundo (WANG; XIAO, 2006), o projeto de mecanismos de escalonamento deve ser baseado em suposições e objetivos. Os objetivos são determinados pela aplicação, com exceção do aumento do tempo de vida, que é um objetivo comum aos mecanismos de escalonamento.

Limitação no suprimento energético e maior tempo de vida possível são suposições básicas que devem ser feitas quando se projeta qualquer solução a ser implantada em RSSF. As outras suposições podem ser classificadas de acordo com a estrutura da rede, estratégia de emprego dos nós e capacidade dos sensores.

- *Estrutura da rede*: quanto à estrutura, a rede pode ser plana ou hierárquica. Nas redes planas, os nós desempenham a mesma função e, geralmente, são homogêneos, onde todos os nós possuem a mesma configuração. Nas redes hierárquicas, os nós desempenham papéis diferentes em virtude da formação dos grupos. Em algumas redes, os nós líderes possuem configuração diferente dos outros nós, pois seu papel exige que tenham maior poder de processamento, maior quantidade de energia disponível e, às vezes, antena com maior alcance.
- *Estratégia de emprego dos nós*: os nós podem ser empregados de forma distribuída e uniforme, por exemplo, por via de um lançamento dos sensores de um avião ou colocados manualmente no ambiente a ser monitorado. No segundo caso, os sensores são colocados em posição selecionada, previamente pensada, de forma que a cobertura de sensoriamento esteja garantida.
- *Capacidade do sensor*: a capacidade do sensor pode estar relacionada com fatores diferentes, como raio de transmissão e sensoriamento e mobilidade. A seguir, os fatores

definidos por (WANG; XIAO, 2006).

- *Modelo de Detecção*: de forma geral, a capacidade do nó em sensoriar um evento está relacionada com o raio de alcance de sensoriamento do nó, o que determina um modelo de detecção determinístico; no entanto, em (XING et al., 2004), essa capacidade é definida em função da distância do objeto ao sensor, o que determina um modelo de detecção probabilístico.
- *Área de sensoriamento*: com relação à área de sensoriamento, devem ser definidos o formato da área e o alcance do sensor. Quanto à área, esta pode ser definida de forma arbitrária, em formato circular ou quadrangular, em 2D ou 3D. O alcance, por sua vez, pode ser ajustável de acordo com a necessidade da situação ou fixo.
- *Sincronização de tempo*: podem ser sincronizados ou não. Alguns mecanismos assumem que os nós são sincronizados para que todos acordem ao mesmo tempo a fim de iniciar um novo ciclo no processo de escalonamento.
- *Modelo de falha*: os mecanismos podem considerar que um nó falha apenas quando sua energia acaba ou quando acontece algum acidente que cause um dano ao nó sensor antes que sua energia tenha acabado.
- *Mobilidade do Sensor*: os nós de uma rede podem ser móveis ou estacionários, no entanto, a maioria dos mecanismos assume que os nós são estacionários. A mobilidade é mais encontrada na estação-base como forma de distribuir o gasto energético dos nós (ALSALIH; HASSANEIN; AKL, 2010); (AZAD; CHOCKA-LINGAM, 2006).
- *Informação de Localização*: a informação de localização que, nas redes de sensores, são obtidas por meio do uso de dispositivos GPS, são consideradas na maioria dos mecanismos de escalonamento.
- *Informação de Distância*: informações de distância também podem ser utilizadas pelos mecanismos e são obtidas por meio da informação de localização.

Os objetivos podem ser divididos entre comum, de alto nível e de QoS. O objetivo comum é o aumento no tempo de vida da rede. O tempo de vida da rede é parâmetro principal de avaliação de sua performance. Além disso, muitos outros parâmetros, como cobertura, conectividade e disponibilidade dos nós, podem ser reduzidas à análise do tempo de vida da rede. Em (DIETRICH; DRESSLER, 2009), são discutidas essas questões e as várias formas diferentes utilizadas na literatura para definir esse parâmetro. Nesta dissertação, como definido anteriormente, será utilizada a definição mais comum, em que o tempo de vida da rede é definido como

o tempo em que o primeiro nó da rede morre. Outras variantes dessa definição são apresentadas em (DIETRICH; DRESSLER, 2009). Uma delas é definida pelo tempo em que uma fração dos nós vivos ultrapassa um certo *threshold*; outra, divide os nós em dois grupos, os nós críticos e os não críticos. No grupo dos nós não crítico, é permitido que até k nós possam morrer e no grupo dos críticos não existe tolerância. Assim, o tempo em que o primeiro nó do grupo crítico morre, ou o número de mortos do grupo não crítico ultrapassar o valor k , é definido como o tempo de vida da rede.

Os objetivos de alto nível são:

- *Escalabilidade*: está relacionada com a capacidade do mecanismo utilizado na rede não sobrecarregar os nós, de forma que o mecanismo possa ser utilizado em redes com maior número de nós.
- *Robustez*: é um objetivo comum aos protocolos e mecanismos propostos para RSSF. Os mecanismos robustos são capazes de lidar com problemas inesperados, como a perda de nós ou mesmo a inserção de novos nós na rede.
- *Simplicidade*: objetivo imprescindível para os mecanismos aplicados em redes sensores, dada sua limitação em termos de poder computacional.

Os objetivos de qualidade de serviço dependem das tarefas a serem executadas na RSSF. Os mais almejados são garantia de conectividade e cobertura, boa taxa de entrega dos dados, qualidade na vigilância dos alvos móveis a serem monitorados e balanceamento do consumo de energia. Dentre esses, a conectividade pode ser considerada um objetivo primário, já que ela é essencial para que a rede desempenhe sua função. Isto porque, pois para que as informações coletadas em qualquer lugar da rede possam chegar à estação-base, é necessário que todos os nós estejam conectados. Uma rede é considerada conectada se todos os nós conseguem alcançar a estação-base por meio de um ou múltiplos saltos (BULUT; KORPEOGLU, 2011).

Outro objetivo que influencia diretamente no desempenho da rede é a cobertura. A cobertura de sensoriamento é um problema muito estudado, como pode ser visto em (NGUYEN et al., 2010; WANG et al., 2010; YARDIBI; KARASAN, 2010). Segundo (HUANG; TSENG, 2003), o problema de cobertura é umas das questões fundamentais em RSSF, uma vez que ela determina se um ambiente está sendo bem monitorado ou não. A forma como os nós são empregados e a quantidade de dispositivos utilizados são fatores determinantes para se obter uma boa cobertura no ambiente monitorado. Nos mecanismos de escalonamento a cobertura e a conectividade devem ser levadas em consideração durante o processo de escolha dos nós que vão

dormir. A quantidade de nó que permanece ativa deve ser suficiente para que a rede continue conectada e toda região monitorada coberta. Nem todos os mecanismos de escalonamento de atividade se preocupam em garantir cobertura e conectividade. Isso acontece porque foi provado que, se o raio de transmissão de um nó for pelo menos duas vezes o raio de sensoriamento e área for convexa, é garantida a conectividade dos dispositivos, (LIN; HU; ZHANG, 2010).

Os mecanismos de escalonamento podem ser implementados tanto na camada de enlace como na camada de rede (MAHFOUDH; MINET, 2008). Os trabalhos citados nos parágrafos anteriores são exemplos de escalonamento usados na camada de rede. Na camada de enlace, os esquemas geralmente implementam ciclos de trabalhos onde os nós revezam entre um tempo acordado e outro dormindo para evitar o desperdício de energia em permanecer apenas escutando o meio de transmissão. Soluções para camada de enlace podem ser encontradas em (YANG; HEINZELMAN, 2009); (TANG et al., 2011) e (CHOI; CHOI; YOO, 2011).

Em (YANG; HEINZELMAN, 2009), é apresentada uma comparação entre a utilização de esquemas de escalonamento dos nós sensores implementados na camada de rede, na camada de enlace e nas duas camadas simultaneamente. Os resultados obtidos mostram que o desempenho desses mecanismos são influenciados pelas condições da rede e requerimentos exigidos pela aplicação. Na camada de rede, o escalonamento dos nós pode ser realizado em cima de protocolos de controle de topologia que garantem a conectividade da rede, impedindo que os nós indispensáveis para manutenção da conectividade da rede sejam escolhidos para dormir ou em cima de protocolos de roteamento. Assim, os nós que não estão momentaneamente diretamente ligados com o roteamento dos pacotes de dados podem dormir. Já na camada de enlace, os dispositivos têm sua interface de rede desligada para evitar o grande desperdício de energia durante o tempo em que o nó fica apenas escutando o canal.

As simulações do mecanismo apresentado em (YANG; HEINZELMAN, 2009) foram feitas usando um protocolo de roteamento já bastante conhecido na literatura, o *Directed Diffusion* (DD). O protocolo escolhido, em sua implementação original, não permite que os nós da rede se revezem em tempos acordados e dormindo. Dessa forma é implementado um mecanismo baseado em ciclos de dormir e acordar para os nós que não estão no caminho de roteamento dos pacotes em um determinado momento.

Na camada de enlace, o protocolo escolhido para realizar a simulação foi o SMAC, especificamente utilizado para redes sensores sem fio. Ele implementa ciclos de dorme/acorda para diminuir o tempo em que o nó fica apenas escutando. Os resultados mostram que, em um cenário denso com muitos nós redundantes, o escalonamento na camada de rede apresenta melhor resultado. Já o mecanismo implementado na camada de enlace é mais indicado para cenários de

aplicação com pouco tráfego e em redes de pequena escala. Para a utilização do escalonamento dos nós nas duas camadas, seria mais indicado um gerenciamento por meio das camadas que possa escolher entre usar um dos mecanismos ou os dois de acordo com o cenário apresentado.

Os mecanismos de escalonamento são utilizados com o objetivo de salvar energia e aumentar o tempo de vida da rede, por isso eles devem ser o mais simples possível; no entanto, a maioria das técnicas utilizadas exige que os mecanismos tenham informações que, na maioria das vezes, não estão disponíveis nas RSSF. O conhecimento da rede como um todo, por exemplo, é necessário para encontrar o conjunto mínimo e ótimo de nós que devem permanecer ativos. Já informações de localização são utilizadas por alguns trabalhos para verificar se questões como cobertura e conectividade estão sendo garantidas. Logo, desenvolver um mecanismo simples, robusto, escalável e autônomo é, ainda, uma questão em aberto.

A constatação de que há uma carência de soluções com as características citadas há pouco será obtida com a discussão dos trabalhos relacionados, apresentada no próximo capítulo. Os trabalhos que serão expostos se concentram nas soluções de roteamento eficiente em energia e em mecanismos de escalonamento de atividades dos nós porque envolvem os aspectos que serão levados em consideração na solução proposta nesta dissertação.

3 *Trabalhos Relacionados*

A limitação de energia presente nas RSSF é a principal preocupação das soluções desenvolvidas para essas redes. Neste capítulo, serão discutidos os trabalhos relacionados que usam diferentes técnicas na implementação de mecanismos de escalonamento e de roteamento com o objetivo de maximizar o tempo de vida da rede. Na Seção 3.1, são discutidos alguns trabalhos de roteamento encontrados na literatura que fazem uso de inteligência de enxame. Estes trabalhos foram escolhidos porque são baseados na mesma abordagem, colônia de formigas, em que o BiO4SeL se baseia. O BiO4SeL é o protocolo utilizado para implementação da proposta deste trabalho e será mostrado, em detalhes, no próximo capítulo. Na Seção 3.2, são exibidos os trabalhos que utilizam o escalonamento dos nós para economizar energia e aumentar o tempo de vida da rede. Os trabalhos da Seção 3.2 foram agrupados de acordo com os critérios por eles utilizados para realizar a escolha dos nós que podem dormir. Na Subseção 3.2.1, são discutidos os mecanismos que acham o conjunto máximo de conjuntos disjuntos, ou não, para que possam revezar entre períodos dormindo e acordado. Na Subseção 3.2.2, os mecanismos utilizam a informação de quantidade de vizinhos redundantes na hora de escalonar os nós, e, na Subseção 3.2.3, os mecanismos que escalonam os nós de acordo com o papel deles no protocolo de roteamento.

3.1 **Roteamento Eficiente em Energia**

Como mostrado anteriormente, o roteamento é um tópico que há alguns anos é muito estudado e onde muitas soluções já foram apresentadas. Em razão, porém, dos diferentes tipos de aplicação e arquiteturas de rede, existe uma carência de propostas eficientes em termos energéticos que podem ser utilizadas em vários cenários, adaptando-se à arquitetura da rede definida e funcionando de acordo com os requisitos exigidos pela aplicação. Além disso, questões como mobilidade, heterogeneidade dos nós e controle de congestionamento são aspectos ainda pouco considerados nos trabalhos da literatura.

Para cumprir o principal objetivo do roteamento eficiente em energia, que é minimizar a

energia consumida na transmissão de um pacote da fonte ao destino, levando em consideração a energia residual de cada nó, técnicas como *clusterização*, agregação de dados, algoritmos de localização, teoria dos jogos, entre outros, foram utilizadas. Com o surgimento de novas aplicações para RSSF, porém, cresceu também a preocupação com questões como garantia de qualidade de serviço, mobilidade, heterogeneidade dos nós e controle de congestionamento. Portanto, há ainda muito trabalho sendo feito nessa área como pode ser visto em (SONG; CHOI; KIM, 2010; ABE; HONIDEN, 2010; WU; CHEN; CHEN, 2010; BHUIYAN; GONDAL; KAMRUZZAMAN, 2010; LEE; RA, 2010; LUO; HUBAUX, 2010).

O roteamento é responsável por fazer as informações monitoradas chegarem à estação-base. Assim, os primeiros protocolos de roteamento preocupavam-se apenas em realizar essa tarefa. Com o avanço dos estudos feitos nessa área, descobriu-se que a transmissão e a recepção de dados são as tarefas que mais consome a energia dos nós. A partir daí, novos mecanismos começaram a ser implementados com o objetivo de salvar energia durante o roteamento das informações.

Os protocolos de roteamento empregados nas RSSF devem lidar com frequentes mudanças de topologia, que podem ser causadas pela mobilidade dos nós, por defeitos físicos, ou morte ou inclusão de um novo dispositivo. Em razão das características presentes nas RSSF, como dinamismo e tolerância a falhas, as soluções a serem empregadas nesses ambientes precisam ser autonômicas. Segundo (NAMI; BERTELS, 2007), o termo autonômico é definido pela capacidade de um sistema complexo se gerenciar e se adaptar em situações inesperadas sem interferência humana. Com o objetivo de favorecer o comportamento autonômico, adaptativo e descentralizado dos protocolos de roteamento, mecanismos baseados em sistemas biologicamente inspirados estão sendo utilizados em recentes trabalhos (KANDRIS et al., 2011; YANG et al., 2009; GUO; ZHANG; LU, 2010).

A inspiração na natureza despertou o interesse de muitos estudiosos e é aplicada em vários campos de pesquisa. Computational Intelligence (CI) é a expressão definida em (KULKARNI; ANDRSTER; VENAYAGAMOORTHY, 2011) para designar o conjunto de técnicas, biologicamente inspiradas, que fazem uso de mecanismos adaptativos capazes de gerar um comportamento inteligente diante de situações complexas. O uso dos mecanismos de inteligência computacional permite que o sistema se comporte de maneira autonômica, tolerante a falhas e com capacidade de adaptação às mudanças de topologia do cenário.

Ainda segundo (KULKARNI; ANDRSTER; VENAYAGAMOORTHY, 2011), a CI envolve mecanismos como redes neurais, lógica *fuzzy*, aprendizagem de máquina, inteligência de enxame e sistemas imunológicos artificiais. A semelhança entre alguns dos sistemas bi-

ológicos encontrados na natureza e a arquitetura das RSSF incentivou o desenvolvimento de soluções para vários problemas das RSSF. Por exemplo, na detecção de intrusão nos *clusters head* de uma RSSF (AROLKAR; SHETH; TAMHANE, 2011b), em questões de conectividade dos nós (LI; LIU; CUI, 2010), em protocolos de roteamento preocupados com o gasto energético (WANG; SUN; MA, 2010), na atribuição de serviços aos nós (HEIMFARTH; JANACIK, 2008), em mecanismos de escalonamento dos nós sensores (LIN; HU; ZHANG, 2010), entre outros.

O comportamento apresentado nas colônias de formigas, como a forma dinâmica como elas se organizam e se adaptam aos obstáculos que surgem no caminho, caracteriza um sistema autônomo que trabalha de forma auto-organizada e independente de liderança ou controle centralizado. Essas características despertaram o interesse dos pesquisadores, que passaram a criar modelos para soluções de problemas, baseando-se no comportamento das formigas. As Redes Sensores sem fio, em razão da sua natureza restrita com relação à capacidade computacional e energética, apresentam semelhanças com as colônias de formigas, ou seja, um sistema composto por dispositivos limitados que trabalham juntos para solucionar um problema complexo. A seguir serão discutidos alguns trabalhos que utilizam os mecanismos usados pelas formigas para realizar o roteamento das informações.

3.1.1 Protocolos Baseados em Colônia de Formigas

Dorigo; Maniezzo; Colorni (1996) foram os primeiros a empregar soluções baseadas em colônias de formigas no trabalho que definiu um novo paradigma computacional, nomeado *Ant Systems*. A nova abordagem proposta em (DORIGO; MANIEZZO; COLORNI, 1996) foi inicialmente utilizada para solucionar problemas de otimização combinatorial, como, por exemplo, o problema do caixeiro-viajante. Dentre as vantagens apresentadas por essa abordagem, como robustez, versatilidade e autonomia, a principal é o mecanismo de busca baseado em *feedback* positivo, característico de abordagens baseadas em populações. O *feedback* positivo é representado pelo feromônio acumulado pelas formiga no caminho percorrido entre a colônia e fonte de comida.

No caso das colônias de formigas, o uso do *feedback* positivo pode ser representado como na Figura 3.1. Inicialmente, assumindo que as formigas procuram por fontes de comida seguindo caminhos ainda não conhecidos, quando elas se deparam com um obstáculo, no caso da figura, uma bifurcação, elas escolhem entre uma das duas opções de caminho, aleatoriamente. De acordo com a figura, existe um caminho menor do que o outro. As formigas que escolheram o caminho mais curto irão percorrer o caminho de volta à colônia mais rápido. Assim, através

da deposição de feromônio realizada pelas formigas que chegam à colônia mais rápido, as outras formigas terão mais chance de escolher o menor caminho por conta da quantidade de feromônio acumulada, que será maior no caminho mais curto, dado que um número maior de formigas já passaram por ele. Outra vantagem das formigas é a evaporação. A evaporação é a dissipação do feromônio acumulado em um caminho, fazendo com que esse caminho passe a ser evitado. Como, seguindo feromônio acumulado, a maioria das formigas passa a escolher o mesmo caminho, depois de algum tempo, a fonte de comida encontrada através do caminho ótimo se esgota e as formigas deixam de seguir por ele. Como consequência, o feromônio acumulado no caminho é dissipado e ele, gradualmente, deixará de ser escolhido.

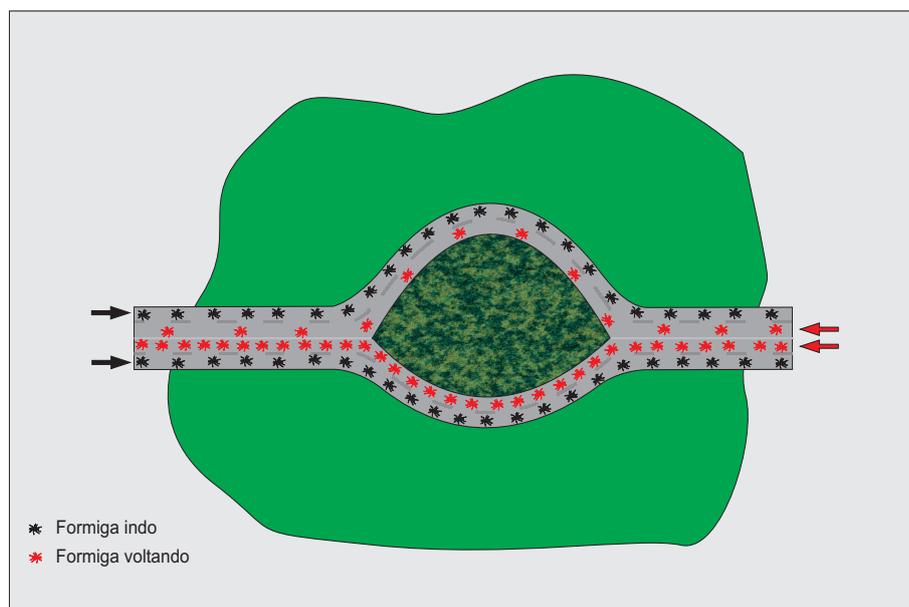


Figura 3.1: *Feedback* positivo.

As tarefas realizadas pelas formigas, como exploração e descoberta de caminhos, estabelecimento de um caminho ótimo e a adaptação às mudanças provocadas por fatores ambientais no percurso definido, correspondem basicamente aos mecanismos necessários para o desempenho satisfatório dos protocolos de roteamento em RSSF. Shuang *et al.* (2007) apresentam um protocolo de roteamento baseado na idéia de colônia de formigas com o objetivo de otimizar o gasto de energia em redes Ad-hoc. São definidas três métricas usadas na descoberta de rotas do nó-fonte para o nó-destino: energia residual dos nós, custo energético do caminho e quantidade de saltos. As métricas escolhidas promovem o gasto de energia de forma distribuída, tendo como consequência o aumento do tempo de vida da rede.

O *Ant-based On-demand Energy Routing* (AOER), como é conhecido o protocolo proposto no artigo, usa basicamente dois tipos de formigas para estabelecer um caminho entre um nó-origem e um destino. A *ForwardAnt* é encaminhada para descobrir uma rota de uma origem s

a um destino d . Isso acontece quando não existe uma entrada na tabela de roteamento referente ao caminho entre s e d . Durante a descoberta da rota, a *ForwardAnt* carrega o custo energético do caminho entre o nó origem e o nó onde ela se encontra no momento. Quando a *ForwardAnt* é recebida em um nó intermediário, ela calcula uma métrica M , que leva em consideração os três parâmetros definidos há pouco. O feromônio do caminho de origem s que chega até o nó n_j através do n_i é baseado na métrica M_{sji} calculada no nó n_j .

Quando a *ForwardAnt* chega ao destino, o pacote é avaliado, o feromônio atualizado e uma formiga do tipo *BackwardAnt* é enviada para o nó origem. A *BackwardAnt* tem a função de escolher o melhor caminho dentre os caminhos descobertos pela *ForwardAnt* entre fonte e destino. A escolha é realizada em cada nó usando uma probabilidade calculada com base no feromônio e, em seguida, armazenada na tabela de roteamento. Para evitar uma utilização excessiva do caminho escolhido, quando a *BackwardAnt* escolhe o próximo nó do caminho, ela diminui o feromônio deixado entre o nó anterior e o nó escolhido. A manutenção de rotas só é realizada quando um caminho está sendo muito utilizado e, como consequência, a energia residual dos nós no caminho atinge um limiar mínimo definido inicialmente. Apesar do ganho no tempo de vida da rede, o algoritmo proposto não se mostra dinâmico e adaptável às modificações que podem ocorrer no ambiente. Como a manutenção das rotas é realizada apenas quando muita energia de um mesmo caminho é gasta, o algoritmo não funciona bem quando existe mobilidade dos nós e a inserção de novos nós na rede.

Em (PAONE et al., 2010), uma solução dinâmica e autonômica para o roteamento de redes sensores sem fio baseado em inteligência de enxame é apresentada. As principais características do protocolo são capacidade de auto-organização e tolerância a falhas. Os autores introduzem um conceito chamado atitude de envio, que é um valor atribuído ao nó e utilizado na construção de um gradiente de feromônio na direção dos *sinks* presentes na rede. Esse valor é calculado em função do estado interno do nó, onde o nível de energia residual é comumente utilizado, e do valor da atitude de envio dos nós vizinhos.

Os pacotes de dados são roteados seguindo o gradiente de feromônio para o *sink* mais próximo. O primeiro passo do algoritmo é a troca de mensagens, sinalizando o valor da atitude de envio e a criação das tabelas de roteamento. No segundo passo, é realizado o roteamento dos dados, onde a escolha do próximo nó é feita baseada no valor da atitude de envio armazenado nas tabelas de roteamento. Periodicamente, um processo de evaporação é realizado pelos nós e as tabelas de roteamento são atualizadas.

Apesar de apresentar uma solução autonômica e distribuída, o gasto energético dos nós no algoritmo não é bem distribuído, uma vez que a escolha do próximo nó no caminho é feita

de forma determinística, baseando-se apenas no alto valor do feromônio. Dessa forma, alguns nós morrerão rapidamente, o que não favorece um dos principais objetivos dos algoritmos de roteamento, que é maximizar o tempo de vida da rede. Outro fator negativo é a escolha de sempre enviar o pacote para o *sink* mais próximo. Se o nó não considera, no momento da escolha, se o caminho para um *sink* mais distante seria o melhor em termos de gasto energético para alguns nós com baixa energia, o algoritmo também deixa de maximizar o tempo de vida da rede, fazendo uma distribuição do gasto energético pouco eficiente.

O algoritmo baseado em formigas proposto por Guo; Zhang; Lu (2010), o *Comprehensive Routing Protocol* (CRP), usa uma tabela de roteamento probabilística, onde são armazenadas as probabilidades de escolha do próximo nó no caminho entre a fonte e o destino. O cálculo da probabilidade de escolha de um nó leva em consideração quatro parâmetros: energia residual do nó candidato, o quadrado da distância entre o nó corrente e o nó candidato, o feromônio acumulado no caminho entre o nó corrente e o nó candidato e a reputação do nó a ser escolhido, onde a reputação é definida como a frequência com que um nó atua como roteador.

A execução do algoritmo é dividida em três fases: a configuração da tabela de roteamento, a comunicação dos dados e a manutenção das rotas. Na primeira fase, a configuração da tabela de roteamento é realizada por meio de uma formiga K, enviada por inundação do destino até a fonte, para que todos os caminhos sejam encontrados. Durante a descoberta das rotas, é realizado também o cálculo da probabilidade de cada caminho, levando em consideração os quatro parâmetros citados há pouco. Na fase de comunicação dos dados, os pacotes são encaminhados para os nós de acordo com as probabilidades armazenadas na tabela de roteamento. Depois que um nó é escolhido, o caminho entre ele e o nó anterior tem seu feromônio atualizado. Na fase de manutenção das rotas, uma inundação é realizada do destino para a fonte para atualizar as rotas e manter os caminhos que estão funcionando de acordo com a situação atual.

O diferencial apresentado no algoritmo CRP é a utilização do parâmetro *reputação* no cálculo da probabilidade de um caminho. O uso dessa métrica busca diminuir a taxa de perda de pacotes, o que é muito importante para algumas aplicações de redes de sensores. O uso de uma formiga enviada por inundação, no entanto, toda vez que os caminhos precisam ser atualizados, gera uma grande sobrecarga na rede, causando desperdício de energia. Além disso, a necessidade de uma formiga de inundação para manter as rotas mostra que o protocolo não é capaz de se adaptar dinamicamente às mudanças que podem ocorrer na rede.

3.2 Escalonamento dos Nós

O projeto de toda RSSF é feito para que ela dure o máximo de tempo possível, em virtude das restrições descritas no Capítulo 2. Uma maneira de conseguir maximizar o tempo de vida da rede é colocando alguns nós para dormir, enquanto os outros permanecem acordados, responsáveis pela realização do sensoriamento e transmissão das informações monitoradas. O escalonamento entre os nós da rede, de forma que haja um revezamento entre os nós que são colocados para dormir e os que permanecem acordados, permite que o gasto de energia seja distribuído uniformemente, produzindo como principal consequência, o aumento no tempo de vida da rede. O escalonamento dos nós pode ser feito de maneiras diferentes. A técnica utilizada depende, normalmente, dos objetivos secundários que se deseja alcançar, dado que todo mecanismo tem, em comum, como principal objetivo, maximizar o tempo de vida. A seguir serão apresentados alguns trabalhos separados pelas diferentes técnicas utilizadas para realizar o escalonamento dos nós.

3.2.1 Soluções de Escalonamento Baseadas no Número Máximo de Conjuntos de Cobertura

Nas RSSF, um conjunto de cobertura é definido como um subconjunto do conjunto de nós empregados em uma rede tal que os nós que compõem o subconjunto sejam suficientes para cobrir a zona-alvo da área monitorada. Como o objetivo dos mecanismos de escalonamento é maximizar o tempo de vida da rede, muitos mecanismos modelam esse problema como o problema de *disjoint set cover*, de forma que o número máximo de conjuntos encontrados possa revezar no estado ativo, distribuindo o gasto energético entre todos os nós da rede. Como nesta subseção os aspectos discutidos são comuns aos mecanismos, a discussão dos trabalhos será apresentada no final da subseção.

Em (CHEN et al., 2009), um mecanismo de escalonamento baseado em algoritmos genético é proposto. Neste trabalho, os autores modelam o problema de maximizar o tempo de vida da rede como o problema de determinar o número máximo de conjuntos de cobertura, relaxando a condição de que os conjuntos sejam disjuntos. O mecanismo proposto utiliza *elitism non-dominated sorting genetic algorithm* (NSGA-II) para selecionar o conjunto de cobertura ativo em um determinado tempo t . O NSGA-II é um abordagem que adiciona o conceito de elitismo ao conceito de dominância já empregado pelo NSGA para classificar as soluções das populações por níveis de dominância (BARRICO, 2007). Outro exemplo de uso dessa abordagem na solução do problema de garantia de cobertura e conectividade pode ser visto em (LI et

al., 2010).

No trabalho apresentado, o autor propõe sua solução, supondo que é empregada uma arquitetura de rede baseada em *clusters* e que cada nó tem a capacidade de determinar sua localização e enviá-la para a estação-base ou seu *cluster head*. Além disso, assume que o algoritmo é executado no *cluster head* e que este tem a responsabilidade de avisar aos nós, membros do *cluster*, a hora em que cada nó fica ativo.

Em (LIN; HU; ZHANG, 2010), é empregada colônia de formigas para determinar um grupo de conjuntos de sensores disjuntos que possam alternar seus estados de funcionamento durante o tempo de vida da rede, garantindo que cada conjunto de sensores formados obedeça as propriedades de cobertura e conectividade. Para que os conjuntos disjuntos possam ser encontrados pela formigas, é necessário construir um grafo que represente os nós sensores e os *sinks*. Na abordagem de formiga utilizada, a chance de dois nós pertencerem ao mesmo conjunto está associada ao valor de feromônio atribuído à aresta que liga esses dois nós .

O mecanismo de escalonamento apresentado em (PAZAND; DATTA, 2009) utiliza uma técnica de conjunto dominante mínimo encontrada em teoria dos grafos para eliminar a redundância dos nós encontrada nos conjuntos de coberturas. A ideia dos autores é encontrar o número máximo de conjuntos de cobertura, de forma que cada conjunto tenha o número mínimo de nós necessários para cobrir as áreas-alvo da região monitorada. A abordagem proposta é dividida em três fases. Na primeira fase, os nós realizam a descoberta da sua vizinhança e a construção do grafo que corresponde a rede formada após a distribuição dos nós. A descoberta dos vizinhos é feita de forma distribuída em cada nó da rede; já a construção do grafo é de responsabilidade da estação-base e é feita de forma centralizada.

A descoberta dos conjuntos mínimos dominantes é feita na segunda fase e, assim como a construção do grafo que representa a rede, é centralizada na estação-base. Na terceira fase, os conjuntos encontrados na segunda fase são escalonados de forma que cada conjunto permaneça ativo por um tempo t , calculado pela estação-base. Embora a proposta não utilize informações de localização, ela é baseada no conhecimento global da rede e realiza muitas tarefas importantes para o escalonamento, de forma centralizada. Como mostrado no Capítulo 2, soluções centralizadas não são indicadas para redes dinâmicas e autônomas como as RSSF.

As soluções apresentadas preocupam-se em aumentar o tempo de vida de rede e garantir qualidade de cobertura dos eventos que ocorrem nas regiões que estão sendo monitoradas. Para oferecer garantia de cobertura, no entanto, as soluções são, na maioria das vezes, implementadas de forma centralizada. Mesmo os mecanismos baseados em uma solução distribuída, como a colônia de formigas em (LIN; HU; ZHANG, 2010), precisam fazer algumas suposições

Números de vizinhos	Porcentagem de área coberta
5	91.62%
7	96.89%
9	98.85%
11	99.57%

Tabela 3.1: Cobertura garantida com diferentes números de vizinhos. Adaptada de (MENG et al., 2010).

para garantir propriedade de cobertura e conectividade, exigindo, para isso, conhecimento da rede como um todo e informação de localização. Portanto, os requisitos das soluções propostas inviabilizam sua aplicação na maioria das RSSF, onde soluções autônomicas e distribuídas obtêm melhor desempenho e mais se aproximam da realidade das aplicações desenvolvidas atualmente.

3.2.2 Soluções de Escalonamento Baseadas na Densidade de Vizinhos de Cada Nó

Wu *et al.* (2005) propõem um método para estimar o grau de redundância dos sensores sem o uso de informação de localização e um algoritmo, chamado *Lightweight Deployment-Aware Scheduling* (LDAS), para escalonar os nós de uma rede. O modelo matemático proposto é usado para estimar a redundância em redes onde os sensores são empregados randomicamente e cada nó sabe seu raio de sensoriamento. Os dados resultantes do modelo matemático são apresentados na Tabela 3.2.2. Outro trabalho que também faz o cálculo do grau de redundância dos nós para garantir sensoriamento é encontrado em (MENG et al., 2010). Os valores mostrados na Tabela 3.2.2 são heurísticas que podem ser utilizadas para estimar a quantidade de vizinhos necessários para obter o grau de cobertura desejado pela aplicação. Utilizando a análise teórica realizada por intermédio do modelo matemático, o algoritmo LDAS escolhe os nós redundantes que podem dormir. Para que o LDAS possa utilizar as heurísticas encontradas pelo modelo matemático, os autores partem do princípio de que não são utilizadas informações de localização, os sensores são empregados randomicamente seguindo uma distribuição uniforme e, além disso, os nós possuem mecanismos para descobrir quantos vizinhos estão localizados na sua área de sensoriamento.

O controle de densidade da rede é feito por meio de um esquema de votação randômica que possibilita os nós revezarem entre três estados: ativo, dormindo e pronto para dormir. Como o LDAS assume que cada nó conhece os vizinhos que estão no seu raio de sensoriamento, quando

um nó está no seu estado ativo, além das tarefas de transmissão e sensoriamento, ele verifica a cada intervalo de tempo, chamado *Checking Time* (CT), a quantidade de vizinhos que estão acordados e ativos. Vizinhos, para o LDAS, são os nós que estão na área de sensoriamento do nó. A quantidade de vizinhos é comparada com um valor-limite pré-configurado e escolhido de acordo com os requisitos de qualidade de serviço exigidos pela aplicação. O valor limite é escolhido com base nos dados apresentados na Tabela 3.2.2. Caso a quantidade de vizinhos seja maior do que o valor pré-configurado, o nó envia um *ticket* para algum vizinho escolhido aleatoriamente. Além disso, no estado ativo, o nó também é responsável por verificar se sua quantidade de *ticket*, recebida dos seus vizinhos, já não é suficiente para que possa dormir.

Um nó passa do estado ativo para o estado pronto para dormir quando ele verifica que já recebeu um número de *tickets* maior do que um *Ticket Threshold* (TT) previamente definido. Com o objetivo de tentar minimizar a quantidade de pontos cegos na rede, quando um nó vai para o estado pronto para dormir, ele espera por um tempo aleatório entre 0 e W_{max} antes de mudar de estado, verifica novamente a quantidade de vizinhos ativos e, somente se o número de vizinhos ainda cumprir os requerimentos exigidos, o nó vai para o estado dormindo, onde ficará por um período *Sleeping Time* (ST). Caso contrário, o nó permanece no estado pronto para dormir, realizando sensoriamento e transmissões.

O LDAS usa basicamente três estruturas de dados: uma tabela de vizinhos, uma mensagem de *beacon* e uma mensagem de *ticket*. Na tabela de vizinhos de cada nó, são guardados o id de cada vizinho, seu estado, o número de vizinhos ativos do nó vizinho e o tempo de atualização de cada entrada na tabela. A mensagem de *beacon* é utilizada para avisar que o nó está vivo e é enviada a cada intervalo de tempo, denominado *Beacon Time* (BT). A mensagem possui dois campos, o id do nó e a quantidade de vizinhos dele que está ativa. Quando um nó recebe uma mensagem de *beacon*, ele atualiza sua tabela de vizinhos e, quando fica muito tempo sem receber uma mensagem de um de seus vizinhos, ele muda o estado do vizinho para dormindo.

A mensagem de *ticket* é enviada por um nó para avisar aos seus vizinhos que um deles pode dormir. A mensagem de *ticket* possui dois campos, o id do nó que a está enviando e o id do nó vizinho que foi escolhido aleatoriamente para dormir. O número de mensagens-*tickets* enviadas depende do número de nós redundantes necessários para cumprir os requisitos de QoS, assumamos esse valor ser r . Quando um nó verifica se seu número de vizinhos ativos é maior do que o valor definido pela aplicação, ele decide se envia ou não um *ticket*. Se o número de nós ativos é n , o nó enviará $n-r$ mensagens-*tickets*, significando que todos os vizinhos redundantes podem dormir.

Apesar de ser definido um tempo de *back-off* para evitar o surgimento de pontos cegos,

essa medida pode não ser suficiente em redes muito densas. Nas redes densas, a quantidade de nós redundantes é muito grande, e como no LDAS não é enviada nenhuma mensagem para avisar que o nó mudará seu estado para dormindo, o roteamento de pacotes de dados pode ser fortemente afetado. Isso acontece porque, como o LDAS deixa apenas um número mínimo de nós acordados, o número de nós dormindo será muito grande, aumentando a possibilidade de um nó que está dormindo ser escolhido para encaminhar um pacote de dados.

Fan Ye *et al.* (2002) apresentam o *Probing Environment and Adaptive Sleeping* (PEAS), que é um protocolo distribuído com o objetivo de aumentar o tempo de vida de rede, deixando apenas um conjunto de nós ativos. Os nós que ficam dormindo acordam de vez em quando para verificar se algum nó na sua vizinhança precisa ser substituído, de forma que seja mantida a densidade de nós ativos e a minimização do consumo de energia da rede. O PEAS é um mecanismo para ser aplicado em redes sensores muito densas, o que explica sua capacidade de adaptação a falhas inesperadas. Além disso, não faz uso de informações sobre os vizinhos de um nó e o conjunto de sensores que permanecem acordados é suficiente para garantir cobertura e conectividade.

O PEAS assume que o nó é capaz de variar seu poder de transmissão, adaptando o alcance da transmissão de acordo com a área circular de raio R que deve ser coberta. Durante o funcionamento da rede, o sensor pode permanecer em três estados diferentes: dormindo, sondando e ativo. Além disso, o protocolo é dividido em dois componentes, o *Probing Environment* (PE) e o *Adaptive Sleeping* (AS). No PEAS, cada nó dorme por um período que segue uma distribuição exponencial calculada por uma função de densidade probabilística (PDF) que varia de acordo com uma taxa de sondagem λ . A taxa de sondagem determina a rapidez com que a rede contará com uma quantidade de nós ativos suficiente para garantir cobertura e conectividade na fase inicial do protocolo.

O *Probing Environment* permite que o nó acorde e sonde sua vizinhança local definida por um raio de sondagem R_s . O valor do R_s é definido de acordo com a redundância de nós ativos exigida para cumprir os requisitos da aplicação com relação a cobertura e conectividade. Inicialmente todos os nós da rede estão no estado dormindo. Como os nós dormem por períodos calculados randomicamente, eles vão acordando em tempos diferentes. Quando um nó acorda, ele muda para o estado sondando e procura no seu R_s por nós que estejam ativos. A procura é realizada por meio de uma mensagem *PROBE* enviada por inundação para todos os vizinhos que estejam no seu alcance de transmissão, onde o alcance de transmissão de cada nó é definido de acordo com o raio de sondagem. Os nós ativos que receberem a mensagem *PROBE* devem enviar como resposta uma mensagem *REPLY*. Se o nó receber pelo menos uma mensagem

REPLY, ele volta a dormir por um novo tempo calculado pela mesma PDF definida inicialmente, mudando apenas o λ , que é alterado de acordo com informações de *feedback* recebidas pela mensagem *REPLY*. Caso contrário, o nó muda para o estado ativo e permanece nele até acabar sua bateria ou ocorrer algum defeito.

O componente *Adaptive Sleeping* é responsável por ajustar a taxa de sondagem utilizada por parte de cada nó que volta para o estado dormindo depois de receber uma mensagem *REPLY*. A taxa de sondagem (λ) de um nó no estado sondando é ajustada de acordo com uma taxa desejada (λ_d) definida pela aplicação e com uma taxa de agregação de sondagem $\bar{\lambda}$, calculada com base em todos os vizinhos de um nó ativo que estão dormindo. Todo nó ativo quando recebe uma mensagem *PROBE* incrementa um contador N . Quando um nó ativo recebe a primeira mensagem *PROBE*, o contador N recebe o valor 0 e a variável t_0 recebe o tempo de recebimento da mensagem. Após receber k mensagens, onde k é um limite estabelecido previamente, o nó ativo calcula a taxa de agregação corrente $\hat{\lambda}$ de acordo com a seguinte função:

$$\hat{\lambda} = \frac{k}{t - t_0}, \quad (3.1)$$

onde t é o tempo corrente. Depois de efetuado o cálculo de $\hat{\lambda}$, o nó ativo seta t_0 como t e N como 0 e realiza todo o processo novamente. A taxa $\hat{\lambda}$ calculada e a taxa de desejo (λ_d) são as informações enviadas na mensagem *REPLAY* e utilizada pelo nós no estado sondando para ajustar sua taxa de sondagem λ e calcular o período em que vai dormir novamente. A nova taxa de sondagem de um nó no estado sondando é calculada segundo a equação:

$$\lambda^{new} = \lambda \frac{\lambda_d}{\hat{\lambda}}. \quad (3.2)$$

Apesar de apresentar a vantagem de não precisar de informações sobre os vizinhos, como localização, o autor assume que o nó pode variar seu poder de transmissão de acordo com a área a ser sensoriada, deixando o mecanismo dependente do uso de técnicas que realizem o controle do poder de transmissão, de forma que a cobertura da rede não seja prejudicada. Além disso, o fato de não ser levada em consideração a energia remanescente do nó na escolha dos nós que ficam acordados, nem no cálculo do tempo em que eles ficam dormindo, não ajuda a aumentar o tempo de vida da rede. Outra desvantagem que vai de encontro ao objetivo de maximizar o tempo de vida da rede dos mecanismos de escalonamento é o fato de um nó, que por um momento esteve sem vizinhos acordado, não poder voltar a dormir, futuramente, mesmo que a densidade de nós ativos esteja mantida.

3.2.3 Soluções de Escalonamento Preocupadas Com a Atividade de Roteamento

Um mecanismo de escalonamento que se preocupa com a interação com o processo de roteamento realizado na rede sensor é proposto em (VAIDEHI et al., 2011). O mecanismo, chamado *SUBCLUST*, proposto em (VAIDEHI et al., 2011), realiza a formação de um grupo de nós responsáveis pela realização do roteamento das informações sensorizadas e de vários subconjuntos escalonados durante o tempo de vida da rede. Os subconjuntos são formados pelos nós que não foram escolhidos para realizar roteamento. Os nós responsáveis pelo roteamento permanecem sempre ativos e formam um grupo chamado *Active Routing Nodes* (ARN). Os nós em ARN são escolhidos pela estação-base assim que os sensores são lançados no ambiente a ser monitorado. Já os subconjuntos são formados por meio de um algoritmo que garante a cobertura da rede. O escalonamento dos subconjuntos é feito de forma que apenas um subconjunto fique ativo por vez. Os dados sensorizados por um dos nós pertencentes a um subconjunto é encaminhado para um dos nós ARN, para que possam ser roteados, por intermédio do menor caminho, para a estação-base.

A seleção dos nós ARN é feita de forma randômica, com o envio de uma mensagem *Active Node Request* (ANR). Depois de escolhido para participar do grupo ARN, o nó começa a identificar os nós que estão na sua vizinhança. A identificação é feita pela mensagem *Neighbour* (NBR) enviada por todos os nós que estão no seu alcance de transmissão.

Os subconjuntos são formados baseados na distância entre dois nós e nas suas áreas de sensoriamento. Inicialmente, o algoritmo coloca um nó n_j que não esteja no grupo ARN no primeiro subconjunto S_i . Depois, é calculada a distância entre n_j e todos os nós cuja área de sensoriamento se sobrepõe à área de sensoriamento do nó n_j . O algoritmo então procura um nó para ser adicionado ao subconjunto S_i . O nó a ser adicionado precisa obedecer duas condições. A primeira é não pertencer ao grupo ARN. A segunda é ter a maior distância calculada, tal que essa distância seja menor do que $2r$, onde r é o raio de sensoriamento. O processo se repete até que todos os subconjuntos sejam formados. Com os subconjunto definidos, o escalonamento começa a ser realizado por meio da escolha de um subconjunto para ficar ativo por um tempo predefinido. Quando o tempo de permanência ativa de um subconjunto expira, um novo subconjunto é escolhido.

Apesar de bastante simples e eficiente quanto a questão de cobertura da rede, o mecanismo proposto em (VAIDEHI et al., 2011) não se mostra aplicável à maioria das aplicações reais de RSSF. A escolha de forma randômica dos nós ARN, dado que os autores assumem que esses nós possuem maior energia, não faz sentido em um cenário onde bateria extra não pode ser

adicionada a alguns nós especiais depois que a instalação da rede é realizada. Dessa forma, supondo que todos os nós tem mesma energia, a escolha de alguns nós para permanecerem ativos durante todo tempo não é uma opção indicada para maximizar o tempo de vida da rede, que é, como mostrado em (WANG; XIAO, 2006), o objetivo principal e comum aos mecanismos de escalonamento. O não revezamento dos nós que compõem o grupo ARN limita o tempo de vida da rede, ao tempo em que o primeiro nó do grupo RNA morre, e esse tempo será curto, uma vez que somente esses nós realizam o encaminhamento de todos os pacotes da rede.

Em (SARAVANAKUMAR; SUSILA; RAJA, 2010), um mecanismo de escalonamento de atividades foi integrado ao protocolo de roteamento LEACH para aumentar o tempo de vida da rede. O LEACH é um protocolo da camada de rede que usa formação de *clusters* e agregação de dados para minimizar o desperdício de energia no roteamento dos dados proposto em (HEINZELMAN; CHANDRAKASAN; BALAKRISHNAN, 2000). O LEACH é dividido em *rounds* e cada *round* é dividido em duas fases, a fase de inicialização do protocolo e a de transmissão das informações. Na primeira, é realizada a formação dos *clusters* e também é feita a eleição do *cluster head*, e, na segunda, o roteamento dos dados sensorizados. No LEACH, a eleição é feita de forma probabilística e distribuída para que todos os nós sejam eleitos *cluster head* pelo menos uma vez. A cada *round*, antes da formação dos *clusters*, todo nó gera um número randômico entre 0 e 1. Se esse número for menor do que um limite $T(n)$, o nó se elege *cluster head*.

O cálculo do limite $T(n)$ leva em consideração três informações, uma porcentagem p de *clusters head*, que é pré-configurada inicialmente, o *round* atual e o conjunto de nós que ainda não foram eleitos nos últimos $\frac{1}{p}$ *rounds*. Dessa forma, a cada intervalo de $\frac{1}{p}$ *rounds*, todos os nós são eleitos *cluster head* e um novo ciclo, em que todos os nós são novamente elegíveis, se inicia. O mecanismo de escalonamento proposto é implementado dentro de cada *cluster*, fazendo com que metade dos nós que formam um *cluster* fiquem no estado ativo e a outra metade no estado dormindo. Os nós permanecem em cada estado por um intervalo de tempo t e depois trocam de estado; quem estava dormindo fica ativo e quem estava ativo dorme.

O mecanismo proposto é simples e apresenta uma vantagem importante em decorrência da sua simplicidade - não gera sobrecarga ao protocolo de roteamento. Como não foi detalhado nenhuma maneira específica para realizar a divisão dos nós em ativos e dormindo, subentende-se, no entanto, que a escolha é feita de forma aleatória. O fato de não ser usada uma heurística na escolha dos nós que permanecem ativos limita o ganho no aumento do tempo de vida da rede, proporcionado pelo mecanismo implementado. Além disso, não é autônomo, uma vez que o próprio nó não tem poder de decisão na definição do seu estado, e ainda carrega as

limitações inerentes ao LEACH, como a restrição de localização da estação-base no centro da região monitorada.

Em (SWAIN; HANSDAH; CHOUHAN, 2010), outro mecanismo de escalonamento integrado a um protocolo de roteamento é apresentado. O roteamento é realizado por meio da construção de uma árvore de *broadcast*, onde a raiz é o nó *sink*. A árvore, construída periodicamente, possui sempre dois caminhos de todo nó ao nós *sink* como forma de prover tolerância a falhas. A construção da árvore se baseia em achar os melhores caminhos em termos de energia residual entre um nó e a estação-base. Dessa forma, tem-se que os nós folha da árvore são os nós com menor energia no momento da construção da árvore. Com base nessa informação, o mecanismo de escalonamento coloca os nós folhas de cada árvore gerada para dormir, enquanto os nós internos permanecem acordados.

O algoritmo de construção da árvore assume que cada nó da rede armazena o id dos dois pais e o custo mínimo associado ao caminho por meio de cada pai até o nó *sink*. Os nós que estão ligados diretamente ao nó *sink* armazenam o id do *sink* no campo dos dois pais. Além disso, o nó guarda outras informações utilizadas no processo de construção da árvore, como seu id, sua energia residual e o custo de ser adicionado ao caminho. A construção da árvore de *broadcast* é dividida em duas fases, na fase I é definido o primeiro caminho e, na fase II, o segundo.

Na fase I, o nó *sink* envia uma mensagem ADV1 por inundação para os seus vizinhos. Quando um nó recebe uma mensagem ADV1, ele seta um tempo de *back-off* para que possa armazenar e/ou atualizar as informações dos seus pais, que pode mudar de acordo com o novo custo calculado toda vez que um nó recebe uma ADV1. O novo custo é calculado com base na informação de custo do pai que vem na mensagem ADV1, somado ao custo do nó ser adicionado ao caminho. Se o nó é vizinho do *sink*, esse custo já é mínimo e não é alterado; caso contrário, o nó verifica se o valor de custo armazenado é maior do que o novo custo. Se for, ele é atualizado. Depois que o tempo de *back-off* expira, o nó envia por inundação a sua mensagem ADV1 com seu próprio id, o id do seu pai e o custo associado ao id do pai armazenado. Depois que um nó envia sua ADV1 aos vizinhos, quando ele recebe outra ADV1, ele compara seu id com o id do pai que está armazenado na mensagem. Se forem iguais, o nó sabe que é um nó interno e portanto permanecerá acordado. No caso de não receber nenhuma mensagem ADV1 com id do pai igual ao seu id, ele se classifica como um nó-folha.

Na segunda fase, o procedimento é basicamente o mesmo - uma mensagem ADV2 será utilizada para achar o segundo melhor caminho entre os nós e o nó *sink*. Na segunda fase, depois que um nó recebe sua primeira ADV2, se ele se declarou um nó interno na primeira fase,

ele atualiza as informações do seu segundo pai e envia por inundação sua mensagem ADV2. Caso receba outras ADV2, elas são descartadas. Se o nó se declarou como nó folha na primeira fase, quando o seu tempo de *back-off* expira, ele começa a dormir até que uma nova árvore seja construída. Depois que a árvore é gerada, os dados podem ser transmitidos para estação-base. Os dados coletados são enviados alternando os pais escolhidos em cada subárvore para que o gasto de energia seja distribuído. Caso os dados sejam sensorizados por um nó-folha, ele acorda, encaminha o dado e volta a dormir.

A ideia utilizada no mecanismo proposto, de primeiro construir as rotas de encaminhamento dos dados e depois colocar os nós com menor energia para dormir, consegue garantir a conectividade da rede. Além disso, o fato dos nós que dormem continuarem realizando o sensoriamento garante também a qualidade de cobertura da área monitorada. A execução do algoritmo de construção da árvore de *broadcast*, no entanto, realizada periodicamente, pode não ser eficiente quando aplicado em redes muito densas e de larga escala. Numa rede muito densa, a grande quantidade de mensagens ADV1 e ADV2 gerada, além de causar desperdício de energia, aumenta o risco de colisão e diminui a taxa de entrega dos dados. Outro aspecto que influencia no desempenho do mecanismo é o intervalo de tempo definido para a construção de novas árvores. Em cenários onde certos eventos acontecem concentrados em um lugar específico, se o tempo for muito grande, os dois nós escolhidos como pais do nó que está monitorando o evento podem morrer. Se isso acontecer, cria-se um buraco na rede, que compromete a sua conectividade, impedindo que as informações coletadas sejam transferidas ao nó *sink*.

Depois de feita a análise dos trabalhos expostos nesta seção, fica evidente que é difícil encontrar uma solução para RSSF que seja completamente distribuída, autonômica, descentralizada e independente de informações, que não possam ser alcançadas com a troca de apenas uma mensagem, e, ainda, eficiente em termos de gasto energético. No próximo capítulo, no entanto, será apresentada uma solução que otimiza o tempo de vida da rede, realizando o roteamento por meio do uso de formigas, e apresenta as características citada há pouco.

4 *BiO4SeL*

O roteamento eficiente é muito importante para se conseguir economizar energia no processo de transmissão dos pacotes de dados. Como exposto no capítulo anterior, muitos trabalhos estão usando colônia de formigas para obter um roteamento dinâmico, descentralizado e autônomo, que produz, como principal consequência, a otimização do tempo de vida da rede. Neste capítulo, será mostrado o protocolo *Biologically-inspired Optimization for Sensor network Lifetime* (BiO4SeL) (RIBEIRO; CASTRO, 2010a) que consegue, quando usado em redes esparsas, otimizar o tempo de vida da rede por meio do roteamento baseado em formiga.

O BiO4SeL funciona de modo autônomo e distribuído, aumentando o tempo de vida da rede, melhorando a distribuição do gasto energético e mantendo a taxa de entrega de pacotes. Quando empregado em redes densas, no entanto, ele não consegue maximizar o ganho no tempo de vida da rede. A proposta deste trabalho, que será apresentada no próximo capítulo, é implementar um gerenciamento de energia, através do escalonamento dos nós, no BiO4SeL, para otimizar seu ganho no tempo de vida da rede quando aplicado em redes densas. Por essa razão, serão expostos detalhes do seu funcionamento neste capítulo.

Os autores do BiO4SeL implementam um protocolo que consegue incorporar o comportamento autônomo e distribuído das formigas e ainda utilizar uma heurística com base na energia remanescente dos nós para controlar a deposição feromônica na rede. Com a utilização dessas ideias, o BiO4SeL apresenta-se como uma opção eficiente a ser empregada nas RSSF quando comparado com outros protocolos de roteamento, como o *Ant Routing Algorithm for Mobile Ad-hoc networks* (ARAMA) (HUSSEIN; SAADAWI, 2003) e o *Ad-hoc On Demand Distance Vector Routing* (AODV) (PERKINS; ROYER, 1999). Além disso, as características das formigas que foram incorporadas ao BiO4SeL fazem com que esse protocolo esteja apto a ser utilizado nas redes sensores empregadas no mundo real, dada sua capacidade de se autoconfigurar e de se adaptar às mudanças frequentes de topologia da rede.

Para a simulação realizada com o protocolo, os autores assumem o emprego de uma rede homogênea, plana, com nós estacionários e com apenas uma estação base e uma fonte, embora

seu algoritmo possa ser aplicado em redes com n estações-base e m fontes. O BiO4SeL assume também que os nós não tenham conhecimento prévio da rede e da localização da estação-base. Ele usa as formigas para estabelecer múltiplos caminho entre fonte e destino e implementa tabelas de roteamento probabilísticas inversas em todo nó, para que, com base na energia residual e no feromônio calculado localmente, o próximo nó no caminho seja escolhido.

Os algoritmos de roteamento baseados em colônias de formigas trabalham, de forma geral, usando formigas para encontrar caminhos ótimos que serão identificados pelas outras formigas por meio do feromônio acumulado no caminho. Para descobrir os caminhos da fonte ao destino e atualizar o feromônio no melhor caminho, são utilizadas, respectivamente, as formigas de envio e de retorno. Boa parte das soluções de roteamento baseadas em colônias de formiga fazem uso dessas duas formigas, como pode ser encontrado em (CHEN et al., 2006) e (YANG et al., 2009). O uso da formiga de retorno, no entanto, gera uma grande sobrecarga na rede e causa desperdício de energia na transmissão dos pacotes de formigas. Por esse motivo, o BiO4SeL utiliza formigas de retorno apenas na fase de inicialização do algoritmo para que a formação de ciclos seja evitada.

O BiO4SeL é dividido em três fases: a de reconhecimento (*Bootstrap*), a de descoberta inicial de rotas e a de troca de dados. Na fase de reconhecimento, cada nó envia uma mensagem *ihello* por inundação para seus vizinhos. Cada mensagem *ihello* tem tempo de vida de um salto, pois ela é usada apenas para fazer o reconhecimento da vizinhança e inicializar a tabela de roteamento em cada nó. Quando um nó recebe uma *ihello*, ele guarda na sua tabela de roteamento o id do seu vizinho e sua energia inicial (*InitEn*).

Como no BiO4SeL assume-se a ideia de que os nós não possuem nenhuma informação sobre a estação-base, a descoberta de caminhos é realizada por inundação a partir da estação na fase de descoberta inicial de rotas. As formigas utilizadas para realizar a descoberta das rotas são as formigas de inicialização (*iant*). As *iants* são enviadas com tempo de vida grande para que possam alcançar a rede inteira e assim os melhores caminhos sejam encontrados. O pacote *iant* contém o tipo de pacote (*iaTp*) e o identificador de inundação (*broadId*). O *broadId* é utilizado para evitar que formigas repetidas sejam reencaminhadas.

O procedimento realizado nessa fase é de suma importância para autonomia do protocolo que, sem informação nenhuma da rede, consegue estabelecer as melhores rotas de cada nó à estação-base. As melhores rotas, inicialmente, são os caminhos mais curtos que, no BiO4SeL, são encontrados por meio do número de saltos que as formigas carregam a partir da estação-base. Quando um nó recebe uma *iant*, o número de saltos feitos pela formiga da estação-base até o nó é salvo na sua tabela. Se um nó recebe uma *iant* com mesmo identificador de uma

formiga já recebida, o pacote é descartado. No caso de ser um identificador diferente, o nó avalia se o número de saltos da nova *iant* é menor do que a distância salva na sua tabela. Se a nova *iant* chegou por um caminho mais curto, a tabela é atualizada, caso contrário, a formiga faz uma atualização negativa do feromônio no salto entre o nó anterior e o nó onde ela acabou de chegar. A diminuição do feromônio faz com que na fase de troca de dados os melhores caminhos tenham mais chance de serem escolhidos.

A redução feromônica é realizada pela seguinte fórmula:

$$F_d^v = F_d^v - (\min((K \times (2 - R_d^v)); 0, 9)) \times (F_d^v - F_{min}), \quad (4.1)$$

onde F_d^v é o feromônio para o destino d por meio do vizinho v , K é um coeficiente de redução feromônica cujo valor é um número arbitrário entre 0 e 1, R_d^v é a relação de salto para o vizinho v com destino d e F_{min} é o feromônio mínimo para um salto qualquer. Para evitar que o feromônio de um caminho decresça demais, fazendo com que ele nunca seja escolhido, ou cresça demais, impedindo que outros caminhos sejam escolhidos, os autores definem um limite mínimo F_{min} e um limite máximo F_{max} para o feromônio acumulado em um salto.

O cálculo da relação de salto para o vizinho v com destino d (R_d^v) é feito com base na contagem atual de saltos da formiga (S^f) e a distância mínima de saltos entre todos os vizinhos para o destino d (S_d^{min}), de acordo com a seguinte fórmula:

$$R_d^v = S_d^{min} / S^f. \quad (4.2)$$

Outro cálculo que é realizado pelo BiO4SeL é a estimativa da energia gasta pelo tráfego realizado entre os vizinhos a cada envio ou recebimento de qualquer pacote após a fase de reconhecimento. O cálculo é feito de acordo com a função que o nó vai executar, se é recepção ou envio de pacotes. Quando um nó está enviando um pacote para seu vizinho v , ele vai estimar a energia do vizinho, levando em consideração que v está recebendo o pacote. O cálculo da estimativa E^v é feito de acordo com a fórmula:

$$E^v = E^v - G_r^v \times FC_r \times L_c, \quad (4.3)$$

onde G_r^v é o coeficiente energético para o estado de recepção do vizinho v , FC_r é um fator de correção de estimativa para recepção (calculado com base na energia real do v que é obtida através de uma mensagem *hello* utilizada na fase de troca de dados) e L_c é o tamanho do pacote em *bits*.

A mesma estimativa é feita quando um nó recebe um pacote de um vizinho v . A diferença está no coeficiente energético e no fator de correção utilizado que, neste caso, serão trocados pelo coeficiente energético para o estado de envio do vizinho v (G_e^v) e pelo fator de correção de estimativa para envio (FC_e). A fórmula atualizada segue:

$$E^v = E^v - G_e^v \times FC_e \times L_c. \quad (4.4)$$

Além da energia gasta estimada por uma dessas duas fórmulas, o nó calcula também uma estimativa da energia despendida durante o tempo em que seu vizinho ficou, supostamente, ocioso. O cálculo é feito com base no coeficiente energético para o estado inativo do vizinho v (G_i^v) e no intervalo de tempo da suposta inatividade do vizinho, como mostra a seguinte fórmula:

$$E^v = E^v - G_i^v \times FC_i \times (T_c - T_l), \quad (4.5)$$

onde T_c e T_l são, respectivamente, os marcadores de tempo de chegada do pacote anterior e do pacote atual.

Além disso, o BiO4SeL implementa um sistema de reenvio de *iant*s. Como as formigas são enviadas por inundação, em ambientes densamente povoados, as *iant*s podem colidir e prejudicar o estabelecimento das rotas. Para evitar esse problema, durante a chegada de cada *iant* a cada nó, um contador com tempo T é disparado. O nó reencaminha a formiga normalmente e espera receber uma formiga reencaminhada por um de seus vizinhos. Se o tempo expirar e o nó não receber nenhuma formiga, a formiga é reenviada. O reenvio só é feito uma vez para evitar sobrecarregar a rede.

Depois que as rotas iniciais são estabelecidas, inicia-se a fase de troca de dados. É nesta fase que os dados são encaminhados do nó-fonte à estação-base e que formigas de anúncios são enviadas a cada intervalo de tempo $\delta\tau$. As formigas de anúncio são chamadas de *hello* e são enviadas de todos os nós da rede para informar os vizinhos que estão vivos e para atualizar o valor de suas energias remanescentes na tabela dos vizinhos. As formigas *hello* possuem tempo de vida de um salto e apenas dois campos, um para guardar o tipo do pacote e outro com a energia atual do nó que a está enviando.

Quando um nó recebe um *hello*, ele verifica se o id do vizinho v que lhe enviou a formiga já está na sua tabela. Se v está na tabela, ele tem sua energia atualizada e seu tempo de expiração reiniciado. Para cada nó numa tabela de roteamento existe um tempo de expiração t associado. Esse tempo t é utilizado para manter a tabela de um nó atualizada com relação aos vizinhos que morrem, apresentam defeito ou saem do seu alcance de transmissão. Se o t de um vizinho

expira antes que um *hello* chegue, ele é retirado da tabela de roteamento.

Além disso, a cada *hello* enviado, a energia remanescente é utilizada para calcular o fator de erro na estimativa de energia gasta, calculada toda vez que um nó envia ou recebe um pacote. O cálculo da estimativa de energia gasta é importante porque é utilizada no cálculo do feromônio que será depositado no caminho escolhido. Dessa forma, o BiO4SeL procura manter essa estimativa o mais próximo possível da realidade, atualizando o fator de erro toda vez que um *hello* é recebido.

O autor define ainda outro tipo de pacote utilizado no BiO4SeL, que é o *RequestHello*. O *RequestHello* é necessário quando um nó recebe um *hello* de um nó que não está na sua tabela de roteamento. Assim, *RequestHello* é enviado ao nó desconhecido com o objetivo de inseri-lo no processo de roteamento. Para isso, no pacote, seguem as informações de gasto energético (enviadas na fase de iniciação por meio do *iHello*), o identificador da estação-base e a distância, em quantidade de salto, até ela. Quando um nó desconhecido recebe um *RequestHello*, ele envia por inundação para os seus vizinhos, a um salto de distância, um pacote *respRqHello* com as mesmas informações do *RequestHello*.

Outra formiga empregada no BiO4SeL é a formiga de carona (*piggyback*), que caminha pela rede de carona no pacote de dados. A formiga de carona (*pbAnt*) possui apenas um campo que é preenchido com número mínimo de saltos entre o nó atual e a estação-base quando o pacote de dados é criado. Essa informação será utilizada no cálculo de deposição feromônica.

O caminho que um pacote de dados percorre da fonte ao destino é escolhido probabilisticamente com base no feromônio acumulado em cada salto. Não existe um caminho pré-configurado; cada nó que recebe um pacote de dados tem que encaminhá-lo e, para isso, ele tem que escolher um de seus vizinhos como o próximo nó no caminho. A escolha é feita levando em consideração o feromônio associado a todos os saltos entre o nó e seus vizinhos. O valor do feromônio acumulado nos saltos pode variar entre um valor $maxF$ e um $minF$ que são predefinidos. Na escolha do próximo nó no caminho, é utilizada uma porcentagem de escolha e a divisão feita entre os valores de feromônio acumulado em um salto pela soma do feromônio de todos os saltos. Isso faz com que o salto com maior feromônio tenha maior probabilidade de ser escolhido, embora exista a chance de novos caminhos serem escolhidos em razão da porcentagem de escolha gerada.

Para evitar, porém, que um caminho seja escolhido sempre, causando a morte dos nós, existe uma probabilidade (pE) da escolha do próximo nó ser feita com base na energia remanescente nos vizinhos. Isso acontece quando a quantidade de energia do nó no salto com maior feromônio é igual ou menor do que a metade da energia do nó com maior energia remanescente.

Como, após a escolha do próximo nó, o feromônio do salto escolhido é atualizado, essa medida contribui para que novos caminhos sejam utilizados e os caminhos com pouca energia sejam poupados.

A atualização do feromônio, depois de escolhido o próximo nó no caminho, é feita de acordo com a seguinte fórmula:

$$F_d^v = F_d^v + ((\max F - F_d^v) \times (c + e) \times fc), \quad (4.6)$$

onde F_d^v é o feromônio do vizinho v com relação ao destino d e $\max F$ é o valor de feromônio máximo acumulado em um salto. c , e e fc são, respectivamente, a constante de caminho, constante de energia e constante arbitrária associadas ao cálculo do feromônio. A constante de caminho é calculada com base no peso atribuído ao caminho no cálculo do feromônio e na relação entre as distâncias mínimas e percorridas até o destino. A constante de energia é calculada com base no peso atribuído à energia e na relação entre energia atual do nó e a sua energia inicial. Mais informações a respeito do cálculo dessas constantes podem ser encontradas em (RIBEIRO, 2009).

Após a atualização feromônica de um salto, o valor de feromônio total até um destino d também deve ser atualizado. Isso é feito como mostrado a seguir:

$$F_d = F_d - (Fa_d^v - F_d^v), \quad (4.7)$$

onde F_d é o feromônio total para o destino d , F_d^v o feromônio ao destino d através do vizinho v e Fa_d^v é o feromônio do salto antes da atualização.

Para distribuir melhor o gasto energético entre os nós da rede, o autor implementa também um processo de evaporação. A evaporação ocorre quando um mesmo salto é escolhido um determinado número de vezes. Se um nó recebe uma quantidade EvC (variável obtida em testes do protocolo) de pacotes de dados vindos de um vizinho v em direção a uma estação-base d , o feromônio desse salto é evaporado de acordo com a seguinte fórmula:

$$F_d^v = \min F + (Ev^{EvI} \times (F_d^v - \min F)), Ev^{EvI} = E^v / EI^v, \quad (4.8)$$

onde F_d^v é o feromônio do vizinho v com relação ao destino d , $\min F$ é o mínimo feromônio possível, Ev é a porcentagem de evaporação, EvI o índice de evaporação, E^v a estimativa de energia atual do vizinho v e EI^v a energia inicial do vizinho v . Depois da evaporação, o valor de feromônio total precisa ser atualizado novamente.

Para evitar o descarte de pacotes, o BiO4SeL implementa ainda uma formiga negativa (*ne-*

gAnt) utilizada para diminuir o feromônio quando ciclos ou caminhos ruins forem detectados. A *negAnt* segue pelo caminho inverso percorrido pelo pacote, seguindo os identificadores guardados na mensagem recebida. A formiga *negAnt* tem apenas dois campos, identificador da formiga e do pacote que causou o ciclo.

O BiO4SeL apresenta uma solução autonômica eficiente em termos de ganho no tempo de vida e em média de energia restante no tempo de vida da rede. Ainda, porém, que tenham sido apresentados mecanismos que incentivem uma distribuição mas uniforme do gasto de energia, esse objetivo não foi totalmente alcançado. É possível aumentar ainda mais o tempo de vida da rede e incentivar uma distribuição melhor do gasto energético, mantendo os benefícios já alcançados pelo BiO4SeL. Uma proposta que busca alcançar esse objetivo, por via da implementação de um mecanismo de escalonamento no BiO4SeL, é apresentada no próximo capítulo.

5 *Mecanismo de Escalonamento no BiO4SeL*

As RSSF é um assunto que foi muito explorado na última década, no entanto, como visto no capítulo 3, muitos trabalhos continuam sendo publicados nessa área. Segundo (ZHANG; MAO; ANDERSON, 2010), a energia e seu consumo otimizado constituem ainda o principal desafio na área de RSSF. Por esse motivo, novas soluções continuam sendo publicadas na literatura. O protocolo BiO4SeL é o resultado de um trabalho de mestrado, realizado no Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat), que produziu bons resultados, conseguindo otimizar o tempo de vida da rede quando comparado com outros protocolos de roteamento. Ele alcançou o objetivo de melhorar o tempo de vida da rede mediante a implementação de um mecanismo autônomo e distribuído baseado em formigas. Detalhes do desenvolvimento desse trabalho podem ser encontrado em (RIBEIRO, 2009).

O BiO4SeL utiliza colônia de formigas para, mediante o uso do feromônio, implementar um sistema que possibilita distribuir o gasto energético entre os nós da rede. Além disso, o emprego de colônia de formigas no estabelecimento das rotas de encaminhamento dos pacotes de dados permite a adaptação eficiente do protocolo ante a ocorrência de problemas no roteamento dos pacotes. Como, na maioria dos casos, as RSSF são utilizadas em ambientes inóspitos, logo sujeitas a falhas, a capacidade de adaptação a problemas inesperados é de fundamental importância no cumprimento dos requerimentos de qualidade de serviço das aplicações.

As características descritas no parágrafo anterior contribuíram para o desempenho eficiente do protocolo nas simulações realizadas usando redes esparsas, como pode ser comprovado em (RIBEIRO, 2009). Por meio do código-fonte disponibilizado pelo autor do trabalho, foi possível realizar alguns testes com BiO4SeL em cenários utilizando redes muito densas. Após algumas simulações do BiO4SeL nesse novo panorama, constatou-se que ele continua desempenhando bem suas funções e alcançando o objetivo de aumentar o tempo de vida da rede. No entanto, Como nas redes muito densas existe desperdício de energia em razão do grande número de nós redundantes, surgiu a ideia de aproveitar o bom desempenho do BiO4SeL nessas redes e

adicionar um mecanismo de escalonamento, de forma que o tempo de vida da rede pudesse ser ainda mais otimizado.

Neste capítulo, será apresentada uma extensão do protocolo BiO4SeL, dividida em duas versões, o BiO4SeL v2 e o BiO4SeL v3, que implementam um mecanismo de escalonamento com o objetivo de maximizar o tempo de vida de redes sensores muito densas. Na seção 5.1, será descrito o problema abordado neste trabalho e, na seção 5.2, será exibida uma descrição detalhada da solução proposta nesta dissertação.

5.1 Descrição do Problema

Como mencionado no capítulo 2, o escalonamento de atividades dos nós de uma rede é uma das principais soluções utilizadas para maximizar seu tempo de vida. Sua aplicação nas RSSF pode ser feita utilizando-se diversas técnicas e pode ser implementada tanto no nível da camada MAC como no nível de camada de rede. Os mecanismos de escalonamento, no entanto, se baseiam em conjuntos de restrições que dificultam sua aplicação nas RSSF encontradas no mundo real.

A tarefa de escolher alguns nós para dormir resulta em uma economia de energia indiscutível, no entanto, dependendo da forma como seja feita a eleição dos nós que vão dormir, questões essenciais para o bom desempenho da rede, como a sua conectividade, podem ser seriamente comprometidas. Por outro lado, a rigidez de algumas soluções, como 100% de garantia de cobertura e conectividade, e suposições muito restritivas, podem tornar inviável a utilização desses mecanismos.

Toda solução projetadas para RSSF deve ser simples para que o objetivo de economia de energia seja alcançado e a rede não seja sobrecarregada. Da mesma forma, para um mecanismo de escalonamento ser bom, ele deve se pautar nesse mesmo princípio. A simplicidade é de fundamental importância para os mecanismos de escalonamento, porque eles são utilizados em redes muito densas. Assim, as perdas geradas pela utilização de um mecanismo muito complexo podem não compensar o ganho de energia obtido pelo mecanismo, caso ele exista. Segundo (DIETRICH; DRESSLER, 2009), todo mecanismo de escalonamento tem que ter como objetivo três características básicas: simplicidade, escalabilidade e robustez.

Apesar de existirem muitos trabalhos propondo soluções para o escalonamento dos nós, na pesquisa feita para realização deste trabalho, não foi encontrada nenhuma solução que se preocupe em apresentar essas três característica juntas. A simplicidade é raramente encontrada na maioria das soluções estudadas. A escalabilidade é a mais fácil de ser identificada e depende

da sobrecarga adicionada pelo mecanismo. A robustez é uma característica fácil de ser alcançada, quando o escalonamento dos nós está diretamente relacionado com o procedimento de roteamento realizado na rede. Como a robustez de um mecanismo é definida como sua capacidade de contornar problemas inesperados, de forma que o desempenho da rede não seja afetado (definido no capítulo 2), se o mecanismo de escalonamento é integrado em um algoritmo de roteamento robusto, ele pode aproveitar a estrutura fornecida pelo protocolo para realizar o escalonamento, fazendo com que ele herde a robustez já inerente ao processo de roteamento realizado na rede.

Por isso, é indicado ao mecanismo de escalonamento que ele seja integrado ao roteamento, evitando que o escalonamento dos nós prejudique o cumprimento da tarefa mais importante das RSSF, que é fazer a informação desejada chegar à estação-base. Vaidehi *et al.* (2011) e Swain; Hansdah; Chouhan (2010) apresentam alguns exemplos de mecanismos de escalonamento integrados a atividade de roteamento das RSSF. É fácil notar a simplicidade desses mecanismos, quando comparados àqueles que não levam em consideração o roteamento. Já a escalabilidade e a robustez ficam dependentes do procedimento de roteamento utilizado.

Portanto, os mecanismos de escalonamento podem se beneficiar se levarem em consideração os protocolos de roteamento utilizado. Esses, por sua vez, devem ser autônomos, dinâmicos e descentralizados, pois essas são as características essenciais para o desempenho eficiente das soluções propostas para RSSF.

O autonomismo caracterizado pela capacidade de autoconfiguração e autoadaptação é essencial nas RSSF, dado que, normalmente, as RSSF não utilizam um administrador responsável pela realização dessas tarefas. Por essas mesmas razões, as soluções de RSSF também precisam ser dinâmicas. O fato de não existir um administrador de rede faz com que seja essencial a capacidade de adaptação e reconfiguração da rede em tempo de execução. Por exemplo, o processo de adaptação da rede, quando um nó responsável por fazer o encaminhamento de um pacote morre, deve ser feito de forma dinâmica. Nesse caso, o procedimento de escolha de um novo nó no momento do roteamento caracteriza o dinamismo do protocolo utilizado, evitando que um pacote que seria encaminhado pelo nó morto seja perdido.

O BiO4SeL, protocolo onde será implementado o mecanismo proposto nesta dissertação, apresenta as características citadas no parágrafo anterior, pois, ele é autônomo, dinâmico e descentralizado. Com o uso das formigas, ele implementa essas características e consegue apresentar bons resultados, principalmente com relação ao aumento no tempo de vida rede. Por meio da análise dos gráficos apresentados em (RIBEIRO, 2009) pode-se constatar, entretanto, que a distribuição do gasto energético entre os nós não está balanceada de modo uniforme,

concentrando-se principalmente nos caminhos mais curtos entre a fonte e o destino.

Com efeito, o objetivo geral deste trabalho é a implementação e a integração de um mecanismo de escalonamento no protocolo BiO4SeL. Como consequência do cumprimento desse objetivo, almeja-se alcançar os seguintes objetivos específicos:

- otimizar o tempo de vida da rede;
- melhorar a distribuição do gasto de energia; e
- escalonar os nós de maneira simples, robusta e escalável.

5.2 BiO4SeL v2 e BiO4SeL v3

O objetivo deste trabalho é implementar um mecanismo de escalonamento no protocolo BiO4SeL, que seja simples, robusto e escalável. Para alcançá-lo, nenhuma alteração complexa podia ser realizada no BiO4SeL. Com isso, o mecanismo de escalonamento foi pensado para ser completamente integrado à estrutura já utilizada pelo protocolo. Diferentemente da maioria dos mecanismos encontrados na literatura, o mecanismo proposto neste trabalho não está interessado em colocar a maior quantidade possível de nós para dormir, mas sim em poupar os nós que estão sendo muito utilizados, forçando outros nós a participarem do roteamento dos dados e distribuindo melhor o gasto de energia entre os nós da rede.

A solução proposta foi dividida em duas versões, o BiO4SeL v2 e o BiO4SeL v3. O BiO4SeL v2 implementa uma heurística no escalonamento dos nós, de forma que a energia e a carga de trabalho de cada nó sejam levadas em consideração. Nessa heurística, dormem os nós com pouca energia e que são muito utilizados no roteamento de pacote de dados. Já o BiO4SeL v3, que é uma extensão do BiO4SeL v2, foi proposto com o intuito de aumentar ainda mais o ganho no tempo de vida da rede, envolvendo também no processo de escalonamento dos nós, aqueles que têm carga de trabalho nula por não serem usados no roteamento dos dados. Os detalhes de cada versão serão apresentados a seguir.

No escalonamento BiO4SeL v2, o nó pode estar em três estados diferentes: *dormindo*, *ativo* e *indeciso*. No estado ativo e no estado indeciso, o nó está apto a realizar suas funções de sensoriamento e transmissão de pacotes. No estado dormindo, o nó fica completamente desligado e não realiza nenhuma função dentro da rede. Inicialmente, todos os nós estão no estado ativo. Quando os dados começam a ser encaminhados da fonte para o destino, é que o nó pode passar para o estado indeciso, com o intuito de decidir se vai ou não dormir. Estando no estado

indeciso, o nó pode voltar para o estado ativo ou mudar para o estado dormindo. O diagrama de estado, mostrando as transições de estado possíveis durante a execução do escalonamento BiO4SeL v2, é mostrado na Figura 5.1.

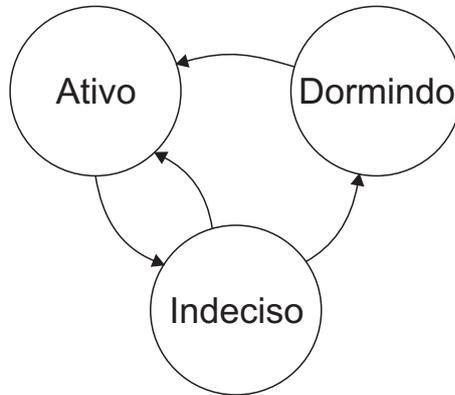


Figura 5.1: Diagrama de Estados.

No BiO4SeL v2, o escalonamento dos nós é realizado pelo próprio nó, de forma descentralizada e distribuída, que se baseando na sua carga de trabalho, decide se pode tentar dormir ou não. A carga de trabalho de cada nó é definida pela quantidade de vezes que ele foi escolhido para encaminhar um pacote de dados, e é representada pela variável *contPktDados*. Quando os dados começam a ser gerados e enviados a partir da fonte, todos os nós estão com o contador de encaminhamento de pacote, o *contPktDados*, zerados. A cada vez que um nó é escolhido como o próximo nó no caminho, ele tem seu *contPktDados* incrementado. Quando a carga de trabalho de um nó é maior do que 15, ele passa para o estado indeciso, onde ele verifica se as condições da rede permitem que ele durma. O valor da carga de trabalho que dispara a mudança de estado do nó, e todas as outras constantes utilizadas no escalonamento BiO4SeL v2, foram definidas por meio de testes.

A decisão de dormir de cada nó é feita pela análise da sua lista de vizinhos e de um conjunto de condições descritas a seguir. O nó estará apto a dormir se ele tiver pelo menos dez vizinhos que estejam no estado ativo e dentre eles pelo menos um que obedeça as condições impostas pelo escalonamento BiO4SeL v2. Eis as condições impostas.

1. Ter energia remanescente maior do que o nó.
2. Estar no estado ativo.
3. Ser diferente da fonte, do nó antecessor e do destino.
4. Estar mais próximo da estação-base do que o nó antecessor.
5. Ser vizinho também do nó antecessor.

Um nó antecessor é o nó que enviou o último pacote de dados para o nó que quer dormir; no caso em que um nó recebe o pacote de dados direto da fonte, a fonte é também o nó antecessor. As condições que um nó indeciso deve obedecer para ir para o estado dormindo procuram evitar que o funcionamento da rede seja afetado por problemas de cobertura e de conectividade. No BiO4SeL v2, a garantia de cobertura é definida segundo dados encontrados em (WU et al., 2005), onde um modelo matemático é definido e utilizado para encontrar a quantidade de nós redundantes de acordo com a porcentagem de garantia requerida pela aplicação. A Tabela 3.2.2 foi mostrada no capítulo 3 e os autores definiram que se um nó tem mais do que 4 vizinhos, mais de 90% do seu alcance de transmissão pode ser coberto por esses vizinhos. No BiO4SeL v2, o número mínimo de vizinhos ativos foi definido como 10 e, de acordo com a tabela apresentada em (WU et al., 2005), isso garante cobertura de 98.86% da área onde a rede está implantada. A conectividade, entretanto, não é 100% garantida em razão da possibilidade de perda da mensagem de atualização do estado enviada quando um nó muda para o estado dormindo. Como será mostrado no próximo capítulo, no entanto, os resultados mostram que o BIO4SeL v2 continua garantindo elevada taxa de entrega.

A Figura 5.2 ilustra a situação onde um nó no estado indeciso tem que decidir se dorme ou não. O nó que está no estado indeciso é o nó que está em vermelho. Na situação apresentada, ele tem dez vizinhos ativos e precisa ter um vizinho que o possa substituir. Suponha-se que todos os vizinhos do nó vermelho estejam acordados e tenham maior energia remanescente do que ele. Dessa suposição, tem-se que as condições 1 e 2 são obedecidas. Assim, todos os vizinhos continuam aptos a desempenhar o papel que o nó vermelho faz no roteamento dos dados; no entanto, os vizinhos 1, 4, 6, 7, 8, 9 e 10 não são vizinhos do nó antecessor, que, nesse caso, é a própria fonte; logo, esses vizinhos não obedecem a condição 3 e não são contados como um possível substituto. Os vizinhos 2, 3 e 5 são vizinhos comuns à fonte e ao nó vermelho, mas, como os vizinhos 2 e 5 estão mais distantes do destino (em quantidade de saltos), num caminho que parte da fonte, eles também não podem desempenhar o papel do nó vermelho, pois não obedecem a condição 4. Portanto, o único vizinho que satisfaz todas as condições e pode fazer o papel do nó vermelho sem resultar em problemas de cobertura e sensoriamento para a rede é o vizinho 3. Como o nó tem pelo menos um vizinho que obedece a todas as condições, ele pode mudar para o estado dormindo.

Depois que o nó decide dormir, ele precisa determinar o período em que ficará no estado dormindo. A fórmula que calcula esse tempo é mostrada a seguir:

$$T = \left(\frac{media_en}{energy} \right) \times sleeptime; \quad (5.1)$$

onde T é o tempo em que o nó ficará dormindo; $media_en$ é a média da energia de todos os vizi-

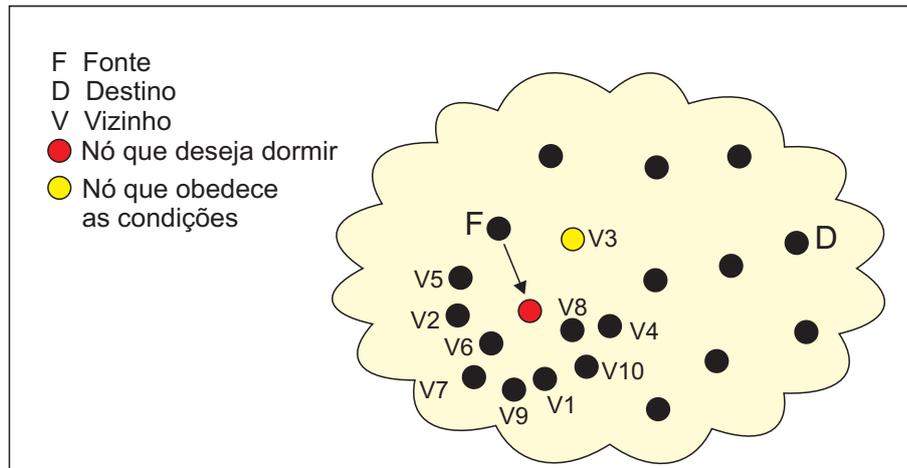


Figura 5.2: Escolha do nó substituto.

nhos do nó e *energy* é a energia atual do nó. Como um dos objetivos deste trabalho é aumentar o tempo de vida da rede, o tempo em que um nó ficará no estado dormindo é definido de forma que quanto menor sua energia, comparada com a energia dos seus vizinhos, maior o tempo que ele permanecerá dormindo. Dessa forma, o tempo que um nó fica dormindo é diretamente proporcional à razão entre a média de energia dos seus vizinhos e a sua energia atual. Outro elemento usado na fórmula que calcula o tempo que um nó permanece no estado dormindo, é a constante *sleeptime*, usada para delimitar o tempo de dormida dos nós. O *sleeptime* foi incluído na Fórmula 5.1 para estabelecer um parâmetro de tempo médio durante o qual os nós dormem. A variação no tempo definido pelo *sleeptime* vai depender da diferença de energia entre os nós e seus vizinhos.

A constante *sleeptime* foi definida com o tempo de 20s depois de vários testes realizados modificando esse valor, que variou de 1s a 30s. Os resultados mostraram que colocar os nós para dormir por um espaço de tempo tão curto como 1s, acarreta grande sobrecarga na rede, provocando perda de pacote e desperdício de energia em decorrência das muitas mensagens de atualização produzidas. Usando o tempo de 30s, o ganho no tempo de vida da rede também é prejudicado. Isto porque, apesar de gerar menos sobrecarga com envio de mensagens de atualização, como os nós passam muito tempo dormindo, um nó que está no estado ativo pode ser utilizado até esgotar sua bateria porque todos os seus vizinhos que o podiam substituir estão dormindo, desfavorecendo a distribuição do gasto de energia e antecipando a morte do primeiro nó da rede. Entre os tempos 5 e 20 segundos, os resultados não apresentaram muitas diferenças, mas com 20s foram obtidos os melhores resultados. Por isso, o *sleeptime* foi definido com o tempo de 20s. Isso acontece porque foi empregado um delimitador de tempo intermediário, que representa tempo suficiente para se obter economia de energia, sem comprometer o gasto de energia dos nós vizinhos. O intervalo de tempo, de média 20s, diminui o número de mensagens

de atualização enviadas e evita que o gasto de energia se concentre em poucos nós. Evitar a concentração do gasto de energia é muito importante para que um dos objetivos deste trabalho, a distribuição mais uniforme do gasto de energia, seja alcançado.

Após calcular o tempo em que ficará dormindo, o nó zera a variável *contPktDados*, que define a carga de trabalho do nó, para que, quando o nó voltar para o estado ativo, ele passe um tempo acordado disponível para encaminhar pacotes, dando aos vizinhos que foram usados enquanto ele dormia a oportunidade de dormir. Depois de realizar esses procedimentos, o nó espera um tempo de *back-off* e então envia uma mensagem de atualização, a *UpdateStatus*, por inundação para todos seus vizinhos. O tempo de *back-off* é utilizado para evitar a perda dos pacotes de dados que podem ser transferidos pelo nó. Mais detalhes sobre a mensagem de atualização serão descritos juntamente com as outras mensagens utilizadas para implementação do mecanismo.

No caso de o nó (que se encontra no estado indeciso) não encontrar um vizinho que obedeça a todas as condições impostas, ele simplesmente volta para o estado ativo. Nesse caso, sua carga de trabalho acumulada é mantida e, desse momento em diante, sempre que sua carga de trabalho for incrementada, ele passará para o estado indeciso para tentar encontrar um vizinho que obedeça todas as condições impostas pelo BiO4SeL v2. Quando ele encontrar esse vizinho, o procedimento descrito nos parágrafos anteriores é realizado.

Quando um nó que está no estado dormindo acorda, ele volta para o estado ativo, sem enviar nenhuma mensagem de atualização. A atualização de estado do nó que acordou é feita por meio da troca de *Hello*s já estabelecida no BiO4SeL. Detalhes de como essa atualização é feita e da integração do mecanismo de escalonamento no BiO4SeL serão fornecidos nos próximos parágrafos.

No escalonamento BiO4SeL v2, o nó tem total controle sobre sua mudança de estado. Ele escolhe se dorme ou não com base na sua carga de trabalho e na quantidade de vizinhos que obedecem algumas condições impostas para assegurar a funcionalidade da rede. As informações sobre a vizinhança de um nó, utilizadas no momento de realizar a decisão de mudança de estado, são obtidas por intermédio das trocas de mensagens já implementadas no BiO4SeL.

Como enunciado no Capítulo 4, o BiO4SeL é dividido em três fases: a de reconhecimento, a de descoberta inicial das rotas e a de troca de dados. No BiO4SeL v2, o escalonamento ocorre durante a troca de dados, mas também utiliza mensagens implementadas pelo BiO4SeL na fase de reconhecimento. Dessa forma, nenhuma mensagem nova é utilizada nas duas primeiras fases do BiO4SeL, apenas alguns novos campos são adicionados às mensagens existentes. No caso da mensagem *iHello* empregada no BiO4SeL para inicializar a tabela de roteamento dos nós,

os campos *status* e *times* foram adicionados. O campo *status* guarda o estado dos vizinhos e o *times* o tempo em que os vizinhos que estão dormindo devem acordar.

Na fase de troca de dados, o BiO4SeL usa três mensagens diferentes: o *Hello*, o *RequestHello* e o *respRqHello*. O *Hello* é uma mensagem utilizada para avisar que o nó está vivo e para atualizar sua energia nas tabelas dos vizinhos. No escalonamento BiO4SeL v2, ela continua desempenhando essa função e tem novos campos adicionados à mensagem para que o processo de decisão feito pelo nó, quando está no estado indeciso, seja realizado com sucesso. São adicionados ao *Hello* quatro novos campos, que são: *status*, *times*, *cntnb* e *vector_nb*. O dois primeiros são os mesmos campos adicionados à mensagem *iHello*. O *cntnb* é usado para guardar a quantidade de vizinhos do nó e o *vector_nb* para guardar o id de todos seus vizinhos. Essa lista de id's é necessária para que um nó no estado indeciso possa verificar se seu candidato a substituto obedece à condição 5, que diz que: o nó substituto deve ser também vizinho do nó antecessor.

Durante o tempo de vida da rede, se um vizinho que está na tabela de roteamento de um nó passar muito tempo sem enviar *Hello*, ele é deletado da tabela. As mensagens *RequestHello* e *respRqHello* foram implementadas para permitir que um nó que foi deletado de uma tabela possa ser adicionado novamente. Detalhes da utilização dessas mensagens já foram descritos no capítulo 4. A *RequestHello* é usada quando recebe um *Hello* de um nó que não está na sua tabela. Nessa situação, a *RequestHello* é enviada para pedir as informações de energia do nó que está sendo adicionado na tabela e a *respRqHello* é apenas a resposta do vizinho a ser incluído na tabela com as informações requeridas. No BiO4SeL v2, elas recebem apenas dois campos, o *status* e o *times*, os mesmos adicionados a *iHello*.

Além dos campos adicionados às mensagens já implementadas no BiO4SeL, no escalonamento BiO4SeL v2, é usada uma nova mensagem que foi implementada para atualizar o estado do nó quando ele decide dormir. A *UpdateStatus* possui quatro campos que são: o identificador do tipo de pacote, o estado do nó, o tempo em que o nó deverá acordar e a sua energia atual. Quando um nó recebe uma mensagem *UpdateStatus*, ele atualiza, além do novo estado do vizinho, sua energia e o tempo em que ele ficará dormindo. O formato da mensagem *UpdateStatus* é ilustrado na Figura 5.3.

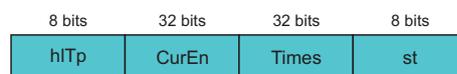


Figura 5.3: Mensagem *UpdateStatus*.

Quando um nó acorda, volta para o estado ativo sem enviar nenhuma mensagem avisando os vizinhos de que acordou e voltou para o estado ativo. A atualização do seu estado nas tabelas

dos vizinhos é feita por meio da mensagem *Hello*. Isso é feito para evitar a sobrecarga da rede com o envio de muitas mensagens de atualização. Essa medida, no entanto, pode fazer com que um nó que está no estado ativo e apto a encaminhar pacote de dados não seja contado como opção no momento de escolher o próximo no caminho de roteamento de um pacote. Para evitar esse tipo de situação, antes da escolha do próximo nó, o nó que realiza essa escolha atualiza o estado de seus vizinhos usando a informação armazenada no campo *times*. Se o tempo definido para o vizinho dormir tiver expirado, o nó atualiza o seu estado na sua tabela. Caso contrário, o vizinho continua fora da lista dos nós que podem ser escolhidos para rotar o pacote de dados.

Como no BiO4SeL v2 só quem apresenta uma carga de trabalho elevada pode mudar para o estado indeciso e ter a chance de dormir, a quantidade de nós que permanecem ativos, mesmo com o escalonamento, é muito alta. Com o objetivo de economizar ainda mais energia, diminuindo o número de nós ativos na rede, foi implementada uma extensão do BiO4SeL v2, chamada BiO4SeL v3, onde, além dos nós que encaminham dados, os nós que raramente são escolhidos também podem dormir.

Todos os estados, constantes, sequência de atividades e mensagens definidas no BiO4SeL v2 continuam sendo válidas no BiO4SeL v3. O procedimento adicionado no BiO4SeL v3 é implementado com o envio das mensagens *Hello*. No BiO4SeL v3, quando chega o momento de um nó enviar um *Hello*, ele muda para o estado indeciso para decidir se ele pode dormir. Para que o nó esteja apto a dormir, ele precisa ter sua variável *contPktDados* igual a zero e no mínimo dez vizinhos ativos. Dessa forma, se no momento antes de enviar o *Hello* ele obedece a essas duas condições, em vez de mandar a mensagem *Hello*, ele envia uma mensagem *UpdateStatus*, avisando aos seus vizinhos que está mudando de estado. O tempo em que ele permanecerá dormindo é calculado seguindo a mesma fórmula definida no BiO4SeL v2. Caso o nó não obedeça as duas condições impostas, ele envia a mensagem *Hello* normalmente. O escalonamento dos nós, seguindo a condição que foi adicionada, faz com que no BiO4SeL v3, tanto os nós que possuem carga de trabalho elevada quanto os que mostram carga de trabalho nula, possam dormir. O procedimento realizado quando um nó acorda é o mesmo seguido pelos nós que têm muita carga de trabalho, pois eles voltam para o estado ativo e esperam o momento de enviar o *Hello* para atualizar seu estado nas tabelas dos vizinhos. O comportamento da fase de troca de dados do BiO4SeL, com as implementações propostas no BiO4SeL v2 e v3, é mostrado no Algoritmo 5.2. A parte escrita em vermelho é executada tanto no BiO4SeL v2 como no BiO4SeL v3, já a parte em azul, apenas no BiO4SeL v3.

Como foi expresso no início do capítulo, almejava-se desenvolver um mecanismo simples, robusto e escalável. A partir da descrição do mecanismo, é fácil perceber a simplicidade da

solução apresentada, dado que ela não utiliza nenhuma técnica complexa que precise de informação de localização e de uma visão geral da rede. A ideia utilizada, baseando-se apenas na carga de trabalho de cada nó, facilitou a integração do mecanismo ao BiO4SeL. A simplicidade do mecanismo decorre do fato de que todo processo de escolha do nó que vai dormir é controlado pelo próprio nó e realizado em todos os nós da rede, de forma descentralizada e autonômica. Além disso, como as soluções propostas neste trabalho foram completamente integradas ao BiO4SeL e este, por sua vez, é um protocolo capaz de se adaptar às situações inesperadas e facilmente escalável para redes de larga escala, o escalonamento implementado no Bio4SeL herda sua robustez e escalabilidade.

Os outros objetivos da proposta descrita neste capítulo, como o aumento no tempo de vida da rede e a melhora na distribuição dos gastos energéticos, serão discutidos no próximo capítulo.

Algoritmo 1 Fase de troca de dados do BiO4SeL com mecanismo de escalonamento BiO4seL v2 e v3. (RIBEIRO; CASTRO, 2010a)

```

1: eventualmente, cada nó envia hello
2: if contPkt = 0 then
3:   entra no estado Indeciso
4:   if obedece condições do BiO4SeL v3 then
5:     envia UpdateStatus
6:     entra no estado Dormindo
7:   else
8:     volta para o estado Ativo
9:   end if
10: end if
11: eventualmente, cada nó envia dados
12: while nó recebe pacote do
13:   decrementa estimativa energética dos vizinhos
14:   if pacote = hello then
15:     if previamente na tabela de roteamento then
16:       conserta estimativa energética dos vizinhos
17:     else
18:       envia requestHello
19:     end if
20:   end if
21:   if pacote = requestHello then
22:     guarda informação de bateria
23:     envia respRqHello
24:   end if
25:   if pacote = respRqHello then
26:     guarda informação de bateria
27:   end if
28:   if pacote = dados then
29:     if previamente recebido then
30:       descarta
31:       envia formiga negativa no caminho inverso
32:     else
33:       calcula próximo salto
34:       ajusta feromônio
35:       encaminha dados
36:       entra no estado Indeciso
37:       Verifica carga de trabalho
38:       if contPktDados >= 15 then
39:         if pelo menos um vizinho obedece condições impostas do BiO4seL v2 then
40:           contPktDados = 0
41:           times = T
42:           espera tempo de back-off
43:           envia UpdateStatus
44:           entra no estado Dormindo
45:         end if
46:       else
47:         volta para estado Ativo
48:       end if
49:     end if
50:   if EB then
51:     descarta
52:   end if
53: end if
54: if pacote = formiga negativa then
55:   decrementa feromônio
56:   if primeiro nó do laço then
57:     descarta
58:   else
59:     encaminha
60:   end if
61: end if
62: end while

```

6 *Resultados e Discussões*

Neste capítulo, serão especificados os experimentos realizados e todos os parâmetros utilizados na simulação da proposta externada no capítulo anterior. Na primeira seção, descrever-se-ão as especificações dos experimentos. Os parâmetros do simulador utilizado para validação do trabalho e os parâmetros definidos no protocolo Bio4SeL e no escalonamento BiO4SeL v2 e v3 são descritos na Seção 6.2. Para finalizar, na Seção 6.3, serão demonstrados os resultados e procedida a discussão a respeito do comportamento do BiO4SeL v2 e v3 comparado com o BiO4SeL.

6.1 **Especificação dos experimentos**

A escolha dos experimentos realizados foi feita de forma que pudesse ser verificado se o BiO4SeL v2 e v3 alcançaram os objetivos definidos inicialmente. O principal objetivo dos mecanismos de escalonamento propostos é otimizar o ganho no tempo de vida da rede quando usado em redes muito densas. Como o tempo de vida da rede é definido como o tempo em que morre o primeiro nó da rede, o experimento 6.3.1 calcula os tempo de morte do primeiro nó para cada um dos protocolos simulados, mostrando que quanto maior o tempo da primeira morte, maior o ganho obtido. Outro objetivo do BiO4SeL v2 e v3 é conseguir distribuir melhor o gasto de energia entre os nós da rede. Os experimentos 6.3.4, 6.3.3, 6.3.8, usando informações diferentes, mostram como é feita a distribuição do gasto energético na rede.

O BiO4SeL v2 e v3, no entanto, não buscam apenas melhorar os aspectos relacionados a energia, eles foram desenvolvidos de forma que as melhorias obtidas não influenciasses negativamente no desempenho de outros aspectos do BiO4SeL. Os experimentos 6.3.5 e 6.3.6 mostram o comportamento do BiO4SeL v2 e v3 com relação a sobrecarga produzida na rede por meio dos pacotes de sinalização e a taxa de entrega de dados obtida com o escalonamento dos nós. Já os experimentos 6.3.2 e 6.3.7 espelham o comportamento dos mecanismos propostos. Detalhes dos resultados apontados nos experimentos serão mostrados na Seção 6.3.

6.2 Parâmetros da Simulação

O Network Simulator (NS-2) (NIST, 2007) foi a ferramenta de simulação utilizada para fazer a avaliação do mecanismo de escalonamento proposto neste trabalho. O NS-2 é uma ferramenta de código aberto e gratuito e por isso bastante utilizada na validação de trabalhos acadêmicos. Porém, ela foi escolhida para simular esta proposta porque os códigos-fontes do protocolo utilizado neste trabalho, o BiO4seL, foram programados para simular no NS-2. A versão do NS-2 utilizada foi a 2.33, pois foi a versão usada para implementação original do BiO4SeL. A seguir serão apontados os parâmetros do NS-2 e a definição dos cenários utilizados na simulação.

6.2.1 Parâmetros do NS

Os escalonamentos BiO4SeL v2 e v3 foram propostos para serem empregados em redes muito densas. Dessa forma, os cenários definidos na simulação variam apenas em quantidade de nós e na densidade da rede. Existem muitas definições para densidade da rede na literatura. Di Francesco *et al.* (2011) definem que uma rede é densa quando o roteamento pode ser realizado utilizando múltiplos caminhos. Em (ZHAO; GOVINDAN, 2003), uma rede é considerada densa se a vizinhança de um nó alcança a casa das dezenas, onde a vizinhança compreende todos os sensores que estão no raio de transmissão de um nó. Já em (YE *et al.*, 2002), assumindo a ideia que é realizado o escalonamento dos nós, uma rede muito densa é definida como uma rede onde o número total de nós da rede é de ordens de magnitude maior do que a quantidade de nós que permanecem ativos.

Nesta simulação, será utilizada a definição de densidade encontrada em (ZHAO; GOVINDAN, 2003), e o grau de densidade será variado de pouco denso para muito denso. Como a definição de densidade usada neste trabalho independe da dimensão do cenário utilizado, não ocorre variação na dimensão da rede durante as simulações. Dessa forma, é fixado um valor de 50mx50m para dimensão dos cenários simulados. Na tabela mostrada a seguir, são apresentados os parâmetros utilizados e seus respectivos valores.

Quantidade de nós	Densidade ((média de viz)/nós)
100	20
200	40
300	60
400	80

Tabela 6.1: *Cenários utilizados nos testes*

Nos experimentos realizados, assume-se o uso de uma RSSF homogênea, estacionária e com apenas uma fonte e uma estação-base. Os nós assumem identificadores singulares que são atribuídos consecutivamente do zero ao $n-1$, onde n é o número de nós da rede. A fonte e o destino são identificados pelos valores $n-2$ e $n-1$, respectivamente. A posição da fonte e da estação-base é predeterminada. Assumindo uma região quadrada e com $50m^2$ de dimensão, definida para os cenários de simulação, a fonte está sempre localizada na posição (5,25) e a estação-base na posição (45,25). A margem de 5m foi definida para aumentar o número de opções de caminhos, partindo da fonte e chegando à estação-base. Essa decisão foi tomada porque, no cenário utilizado nas simulações originais do BiO4SeL, considerado um cenário com dimensões de $50m^2$, a fonte e o destino ficavam localizados, respectivamente, nos pontos (1,1) e (49,49). Esse posicionamento levava à concentração do gasto de energia nos menores caminhos e limitava a quantidade de vizinhos da fonte e da estação-base. Portanto, para evitar esse comportamento, o posicionamento desses nós no cenário utilizado neste trabalho foi modificado.

Os testes foram realizados como explicado a seguir. Para cada quantidade de nós especificada na Tabela 6.2.1, foram gerados, aleatoriamente, 30 cenários diferentes. Como o BiO4SeL é probabilístico, para cada cenário gerado foram executados 30 testes com o objetivo de gerar uma amostra significativa de dados a ser usada no cálculo das médias utilizadas para gerar os gráficos. Além disso, para se gerar uma média confiável, foi definido um intervalo de confiança (α) de 0,95%. O valor 30, definido para quantidade de amostras geradas, foi escolhido porque, segundo (JAIN, 1991), 30 é o valor mínimo de amostra necessárias para o funcionamento confiável do procedimento realizado para encontrar o intervalo de confiança.

O modelo de tráfego utilizado foi o mesmo utilizado pelo BiO4SeL originalmente, o *Constant Bit Rate* (CBR). Conforme definido em (RIBEIRO; CASTRO, 2010a), na fase de troca de dados, é enviado um pacote de 70Kb a cada 0,4 segundo. O tempo de início do envio dos dados também foi mantido aos 28s de simulação, como definido originalmente. Como estão, porém, sendo utilizados cenários muito densos neste experimento, foi preciso aumentar o tempo necessário para que os nós fizessem o reconhecimento de seus vizinhos, passando de 5s para 7s. O tempo total de simulação nos testes realizados neste conjunto de experimentos também foi modificado. No trabalho que apresentou o BiO4SeL, foi definido o tempo de 200s, mas como os mecanismos de escalonamento propostos conseguem otimizar o tempo de vida da rede, os gráficos produzidos com tempo de 200s não deixavam explícito o ganho ocasionado pela implementação desta proposta. Por isso, foi definido o tempo de simulação de 500s.

Outras configurações padrão do NS-2, que não foram citadas, foram mantidas iguais às con-

figurações utilizada pelo BiO4SeL e são mostradas na Tabela 6.2.1, encontrada em (RIBEIRO, 2009). Como mostrado na tabela, o valor da energia da fonte e da estação-base é maior do que a energia dos outros nós. Essa configuração foi usada para não limitar a simulação ao tempo de vida desse dois nós.

Característica	Valor
Canal	Sem fio
Propagação	<i>TwoRayGround</i>
Camada física	IEEE 802.15.4
Fila de interface	<i>PriQueue</i>
Tipo de camada de enlace	<i>LL</i>
Antena	<i>OmniAntenna</i>
Tamanho máximo da fila de interface	50
Parâmetro energético para o envio	0.660
Parâmetro energético para a recepção	0.395
Parâmetro energético para ficar ligado	0
Alcance da antena	15m
Modelo energético	<i>EnergyModel</i>
Energia inicial dos nós	1
Energia inicial da EB e da fonte	2

Tabela 6.2: Configurações do NS. (RIBEIRO, 2009).

6.2.2 Parâmetros dos Protocolos

Para o bom desempenho do BiO4SeL, foram definidos diversos parâmetros em (RIBEIRO, 2009). Como o objetivo deste trabalho é verificar o ganho que o mecanismo de escalonamento proposto acarreta ao protocolo BiO4SeL, todos os parâmetros definidos originalmente foram mantidos. A Tabela 6.2.2 mostra os valores utilizados pelo BiO4SeL e os novos valores, em vermelho, definidos para implementação do BiO4SeL v2 e v3. Os parâmetros adicionados, assim como os parâmetros já definidos pelo BiO4SeL, foram obtidos por meio de testes, realizados com o objetivo de encontrar a configuração do parâmetros que resultasse no melhor resultado.

6.3 Resultados

Nesta seção, anotou-se os resultados obtidos com o gerenciamento autônomo de energia realizado pelo BiO4SeL v2 e v3. Os gráficos vão mostrar se os objetivos, de aumento no tempo de vida da rede e de uma distribuição mais uniforme do gasto energético, foram alcançados.

Parâmetro	Valor
Tempo de envio de cada <i>ihello</i>	$t = 0,01 \times id$
Quantidade de formigas de inicialização (<i>iant</i>) enviadas	$iant_{cnt} = 5$
Intervalo entre cada envio de <i>iant</i>	0,5s
Intervalo entre cada envio de <i>hello</i>	7,5s a 12,5s
Tempo de expiração do contador de <i>hello</i>	13s
Tempo de expiração das entradas de <i>unicast</i>	0,5s
Tempo de expiração do contador de entradas de <i>unicast</i>	1s
Mínimo de feromônio (<i>minF</i>)	0.000001
Máximo de feromônio (<i>maxF</i>)	0.01
Feromônio inicial (<i>iF</i>)	0.0001
Intervalo de envio de <i>hello</i> (<i>hlInt</i>)	10s
Expiração do contador de pacotes (<i>EvC</i>)	2 pacotes
Índice de evaporação (<i>EvI</i>)	3
Peso do caminho (<i>pc</i>)	0,1
Constante arbitrária associada ao cálculo feromônico (<i>fc</i>)	0,1
Coefficiente de diminuição feromônica (<i>K</i>)	0,6
<i>Sleeptime</i>	20s
<i>contPktDados</i>	15
<i>cnt_nb</i>	1

Tabela 6.3: *Parâmetros do protocolo. Adaptada de (RIBEIRO, 2009).*

Além disso, será mostrado o impacto do mecanismo de escalonamento na entrega dos pacotes de dados e na sobrecarga produzida na rede pelos pacotes de sinalização.

6.3.1 Experimento 1: Tempo de Vida pela Quantidade de Nós

Como já definido neste trabalho, o tempo de vida da rede representa o tempo em que o primeiro nó da rede esgota sua energia.

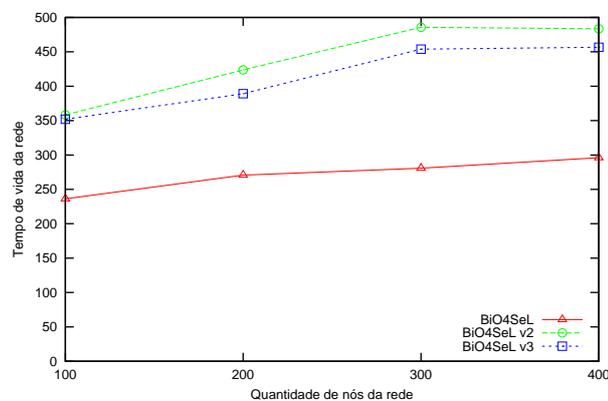


Figura 6.1: Comparação entre os tempos de vida dos protocolos.

O gráfico da Figura 6.1 mostra claramente o ganho, no tempo de vida da rede, obtido pelas duas versões do mecanismo de escalonamento implementado no BiO4SeL. Outro indicativo

expresso na Figura 6.1 é o crescente aumento no tempo de vida da rede em função do aumento da sua densidade. Isso acontece porque, em cenários onde a densidade é baixa, são poucos os nós que mudam para o estado dormindo em decorrência das restrições estabelecidas para evitar problemas de conectividade e cobertura. Além disso, esse resultado já era esperado, dado que os mecanismos implementados foram propostos para redes muito densas. Portanto, fica provado o bom desempenho do BiO4SeL v2 e v3 com relação ao tempo de vida da rede quando empregado em redes muito densas.

Um fato interessante apresentado neste gráfico é o ganho de energia no tempo de vida da rede do BiO4SeL v2 ser maior ou igual que o do BiO4SeL v3, em todos os cenários. Como no BiO4SeL v3 um número maior de nós muda para o estado dormindo, esperava-se que o ganho apresentado por ele fosse maior; no entanto, ainda que por uma pequena diferença, o BiO4SeL v2 tem maior tempo de vida. Esse fato deixa claro que fazer a escolha dos nós aptos a dormir se baseando em uma boa heurística, assim como é realizado no BiO4SeL v2, é mais importante do que colocar o maior número de nós possíveis para dormir.

6.3.2 Experimento 2: Média de Bateria na Rede

Na Figura 6.2, nos 30 primeiros segundos (em média), a média de energia é a mesma nos três protocolos. Isso acontece porque, até ser iniciada a fase de troca de dados, o comportamento dos três protocolos é igual. Além disso, com o aumento no número de nós, e por consequência da densidade, percebe-se um gasto maior de energia durante os primeiros segundos da simulação, decorrente do maior número de mensagens trocadas entre os vizinhos durante a fase de inicialização.

Na Figura 6.2(a), como a densidade da rede com 100 nós ainda é baixa, a diferença entre o BiO4SeL e o BiO4SeL v2 não é tão grande. Com relação ao BiO4SeL v3, é um pouco maior, mas sem muito contraste. Quando se observam, porém, as Figuras 6.2(b), 6.2(c) e 6.2(d), onde, respectivamente, a densidade vai aumentando, pode-se notar um aumento gradual, na média de energia, quando é utilizado o BiO4SeL v2. Já o BiO4SeL v3, no cenário de redes muito densas, como os das Figuras 6.2(c) e 6.2(d), consegue aumentar consideravelmente sua média de energia, acentuando ainda mais a diferença entre a sua média de energia e as apresentadas pelo BiO4SeL e o BiO4SeL v2. Isso ocorre em razão da grande quantidade de nós que mudam para o estado dormindo no BiO4SeL v3.

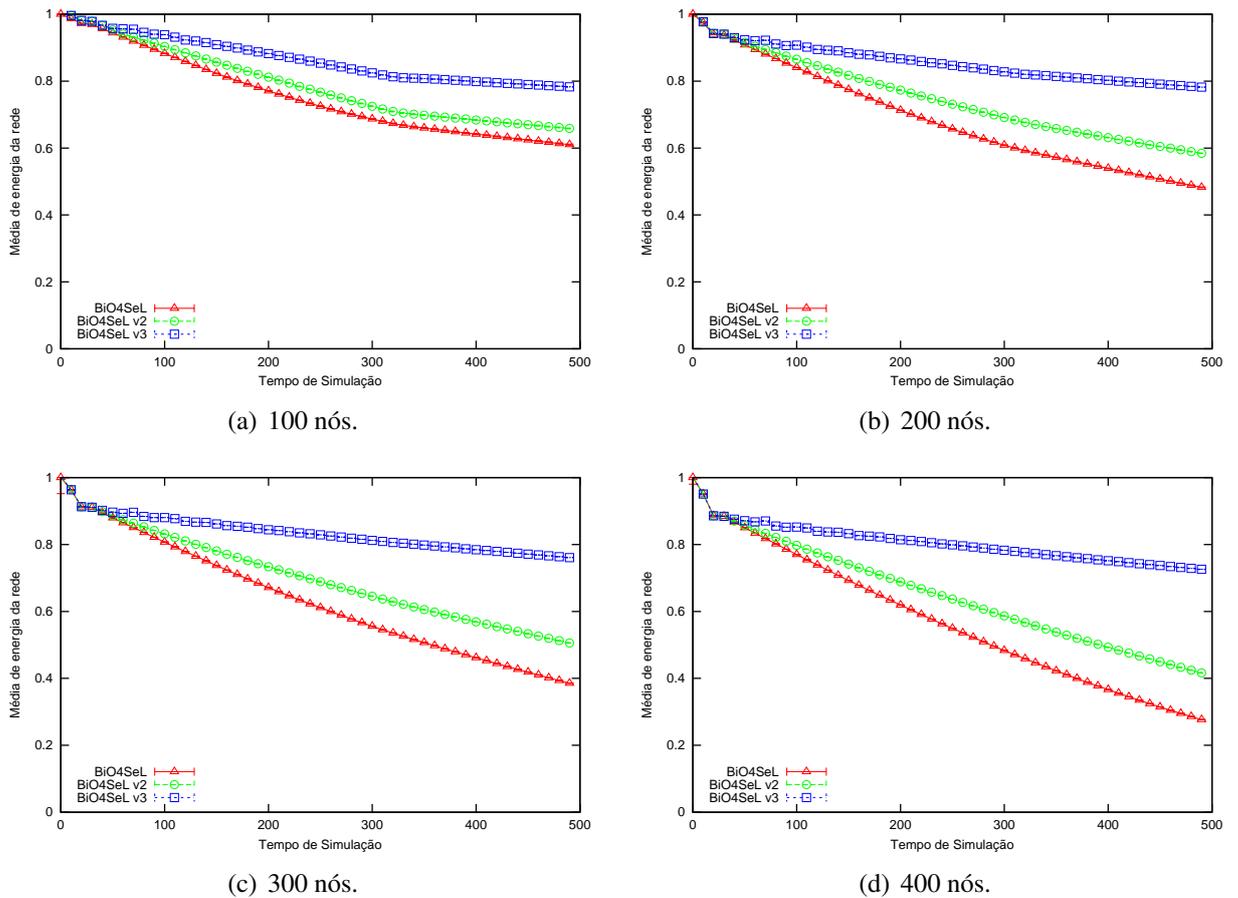


Figura 6.2: Média de bateria na rede

6.3.3 Experimento 3: Distribuição Energética no Tempo de Vida da Rede

Os gráficos desta subseção mostram a distribuição energética entre os nós no tempo de vida da rede, em todos os cenários definidos. Nas distribuições apresentadas, os picos do gráfico representam os nós com maior energia e as depressões os nós com menor energia. Dessa forma, quanto menor a diferença entre os picos e depressões representados no gráficos, melhor a distribuição apresentada pelo protocolo. A fonte e o destino estão localizados, respectivamente, nos pontos (5,25) e (45,25).

Nas Figuras 6.3(a), 6.3(b), 6.3(c) e 6.3(d), pode-se observar que, mesmo realizando uma boa distribuição entre os nós, no BiO4SeL, existe uma leve concentração do gasto de energia nos nós dos caminhos mais curtos. Isso pode ser percebido, nos picos e depressões que existem em alguns nós desses caminhos. Nos gráficos utilizados nesta subseção, os caminhos mais curtos são os caminhos nos arredores do ponto $(w,25)$, onde w representa o intervalo de pontos do eixo x .

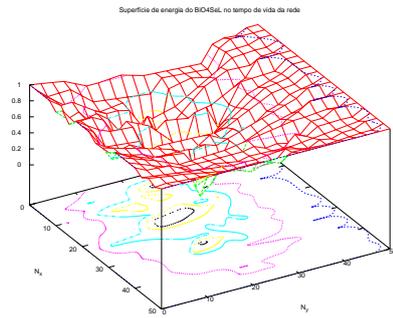
O BiO4SeL v2 consegue mostrar uma distribuição um pouco mais uniforme nos cenários mais densos, como os mostrados nas Figuras 6.3(g) e 6.3(h). Além de evitar os picos e depressões nos caminhos mais curtos, o BiO4SeL v2 consegue fazer com que caminhos poucos utilizados no BiO4SeL participem mais ativamente da transmissão dos dados. Isso pode ser percebido por meio das marcações azul, rosa, verde e amarela, que aparecem nos gráficos, mostrando os níveis de energia remanescentes, respectivamente, do maior para o menor. Pode-se ver, por exemplo, que na Figura 6.3(h) as marcações verde e amarela alcançam nós mais distantes das rotas mais curtas do que as marcações verde e amarela na Figura 6.3(d), evidenciando que o BiO4SeL v2 consegue provocar gasto de energia em nós pouco utilizados pelo BiO4SeL.

O BiO4SeL v3, por sua vez, não consegue realizar uma distribuição uniforme em toda a rede. Em razão da grande quantidade de nós colocados para dormir, o gasto de energia passa a se concentrar também nos caminhos inicialmente mais utilizados. Como, porém, o procedimento implementado no BiO4SeL v2 também foi implementado no BiO4SeL v3, ele melhora a distribuição de energia entre os caminhos mais escolhidos. Se isso não acontecesse, os gráficos apresentariam grande diferença de energia entre os pontos mais altos e os mais baixos. Esse comportamento pode ser observado, nos cenários mais densos, representados nas Figuras 6.4(c) e 6.4(d).

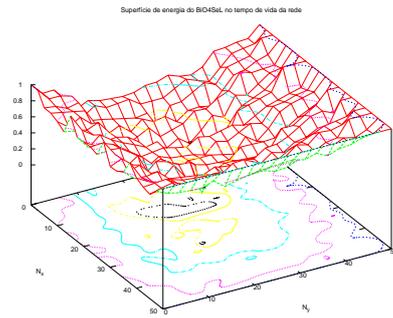
6.3.4 Experimento 4: Coeficiente de Variação da Média Energética

Segundo (RIBEIRO, 2009), o coeficiente de variação é dado pela variância dividido pela média. O coeficiente de variação, neste trabalho, é utilizado para verificar quanto, em média, a energia dos nós se distanciam da média de energia. Com esse experimento pode-se observar se a distribuição de energia da rede está sendo realizada de maneira mais uniforme. Nos gráficos apresentados, quanto menor o coeficiente de variação, melhor a distribuição do gasto energético entre os nós da rede.

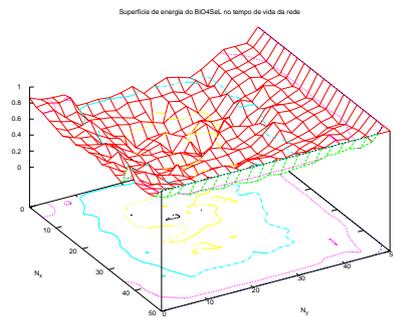
Analisando as Figuras 6.5(a), 6.5(b), 6.5(c) e 6.5(d), nota-se a redução do coeficiente de variação do BiO4SeL v2 comparado ao BiO4SeL, o que confirma o resultado representado nas Figuras 6.3(g) e 6.3(h), de que o BiO4SeL v2 distribui melhor o gasto de energia. Percebe-se também que o BiO4SeL, com o aumento da densidade, aumenta o coeficiente de variação. Isso acontece porque, em cenários muito densos, o BiO4SeL faz com que a média de energia diminua, pois ainda que uma grande quantidade de nós não seja utilizada no roteamento, esses nós continuam vivos e gastando energia com as mensagens de sinalização. Assim, como a média de energia é menor e a distribuição menos igualitária, com o aumento da densidade, o coeficiente de variação tende a crescer.



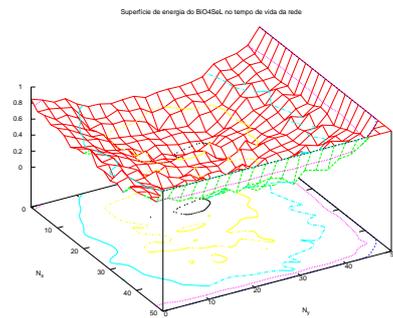
(a) BiO4SeL, 100 nós.



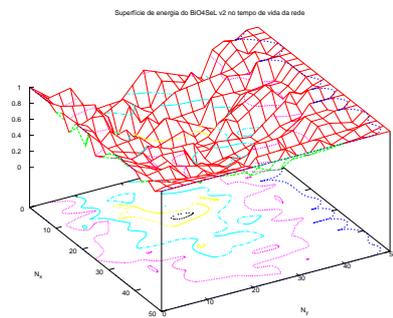
(b) BiO4SeL, 200 nós.



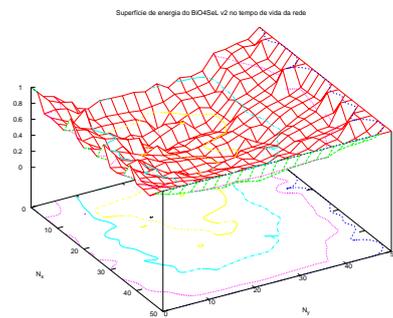
(c) BiO4SeL, 300 nós.



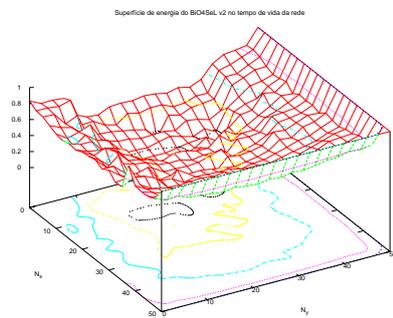
(d) BiO4SeL, 400 nós.



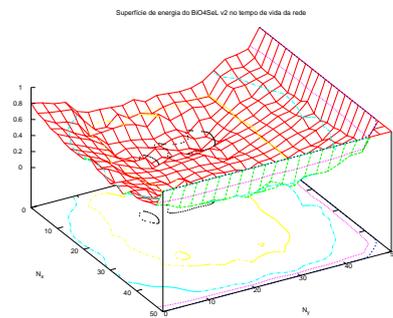
(e) BiO4SeL v2, 100 nós.



(f) BiO4SeL v2, 200 nós.



(g) BiO4SeL v2, 300 nós.



(h) BiO4SeL v2, 400 nós.

Figura 6.3: Distribuição energética na rede, no tempo de vida

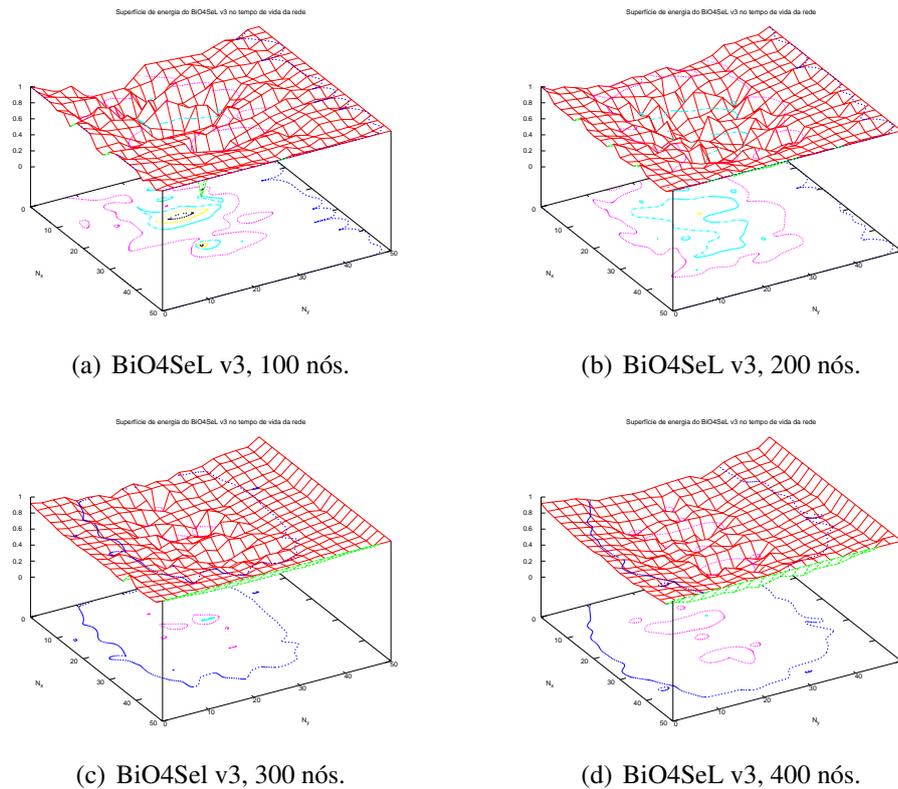


Figura 6.4: Distribuição energética na rede, no tempo de vida, continuação

Já o BiO4SeL v3, ainda que não apresente a melhor distribuição energética, é o protocolo com menor coeficiente de variação. Fato explicado pela média de energia elevada do protocolo, já que muitos nós dormem durante o tempo da simulação economizando bastante energia.

6.3.5 Experimento 5: Pacotes de Sinalização pelo Tempo

Os pacotes de sinalização aumentam conforme cresce a densidade da rede para todos os protocolos. Como os mecanismos de escalonamento, BiO4SeL v2 e v3, funcionam da mesma forma que o BiO4SeL nas fases de inicialização e descoberta de rotas, os três protocolos produzem as mesmas quantidades de pacotes de sinalização durante a execução das duas primeiras fases do algoritmo BiO4SeL. Isso pode ser confirmado por meio das Figuras 6.6(a), 6.6(b), 6.6(c) e 6.6(d).

Apesar de ser implementada uma nova mensagem no BiO4SeL v2 e v3, não é gerada sobrecarga na rede, porque o escalonamento dos nós faz com que muitas mensagens de sinalização do BiO4SeL não sejam enviadas. O BiO4SeL v2, como coloca poucos nós para dormir, consegue manter a produção de pacotes de sinalização no mesmo nível que o BiO4SeL. Já o BiO4SeL v3, ocasiona menor quantidade de pacotes de sinalização em todos os cenários, quando comparado

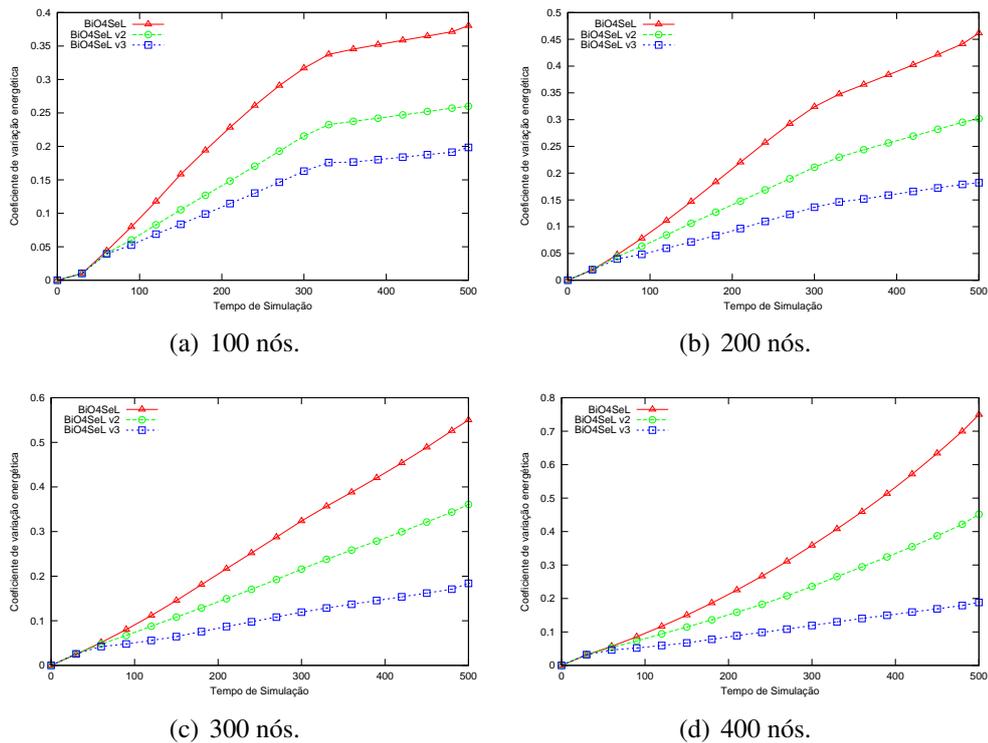


Figura 6.5: Coeficiente de variação da média energética pelo tempo

com o BiO4SeL e o BiO4SeL v2. Isso acontece em decorrência da grande quantidade de nós que ficam no estado dormindo no BiO4SeL v3.

6.3.6 Experimento 6: Taxa de Entrega pela Quantidade de Nós

A taxa de entrega foi dividida em dois gráficos, um que representa a taxa de entrega até o tempo de vida de rede e o outro é a taxa de entrega depois que o primeiro nó da rede morre. Na Figura 6.7(a), onde a taxa de entrega antes do tempo de vida da rede é representada, pode-se notar um pequeno decréscimo na taxa de entrega do BiO4SeL com o aumento da densidade. O oposto é observado para o BiO4SeL v2 e v3, que, com o aumento da densidade, garante uma taxa maior de entrega dos dados. Isso acontece porque, para que os mecanismos de escalonamento propostos funcionem da melhor forma, eles têm que ser aplicados em redes muito densas para eludir problemas de conectividade, evitando também a perda de pacotes. Como expresso no Capítulo 5, o BiO4SeL v2 e v3 não garante 100% de conectividade da rede em razão da possibilidade de atraso e perda da mensagem que atualiza, nas tabelas dos seus vizinhos, o estado do nó que vai dormir. O gráfico da taxa de entrega mostra, todavia, que, em ambientes muito densos, a diferença entre a taxa de entrega do BiO4SeL para o BiO4SeL v2 e v3 é pequena, fazendo com que, em cenários que não exijam 100% de confiabilidade na entrega dos dados, o

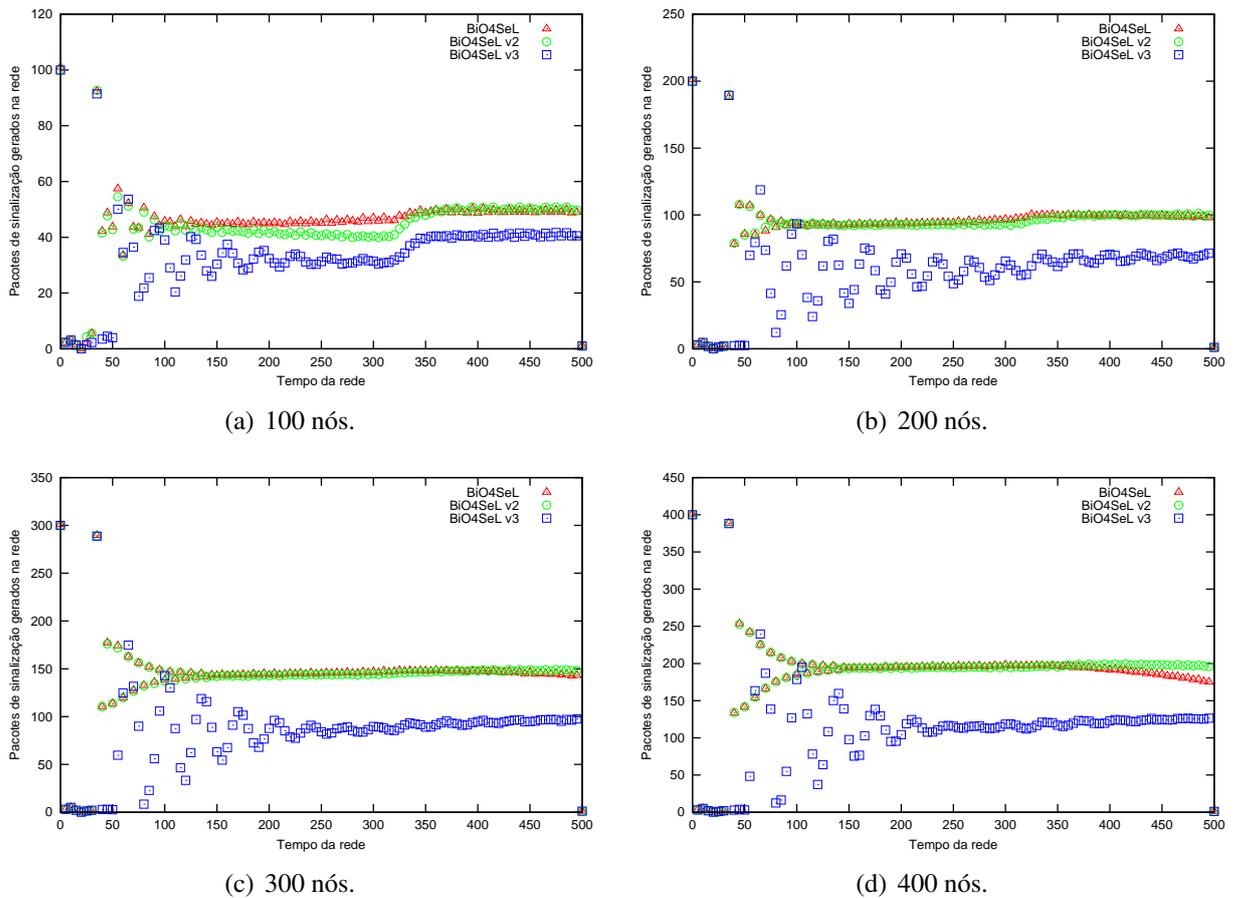


Figura 6.6: Histograma de envio de pacotes de sinalização

ganho no tempo de vida compense as poucas perdas de pacotes que acontecem no BiO4SeL v2 e v3.

Na Figura 6.7(b), que representa a taxa de entrega depois do tempo de vida da rede, observa-se o mesmo comportamento identificado na Figura 6.7(a). Isso é explicado pela densidade das redes simuladas. Ainda que ocorra a morte do primeiro nó da rede e de muitos outros nós depois disso, a grande quantidade de nós utilizados garante que continuem existindo rotas para os pacotes serem encaminhados até a fonte. O uso das redes densas melhorou também o resultado do BiO4SeL na taxa de entrega depois do tempo de vida da rede quando comparado com o resultado apresentado originalmente. Quando ele foi desenvolvido, foi simulado em redes esparsas, e nessas redes, depois das primeiras mortes, a taxa de entrega do protocolo caía significativamente.

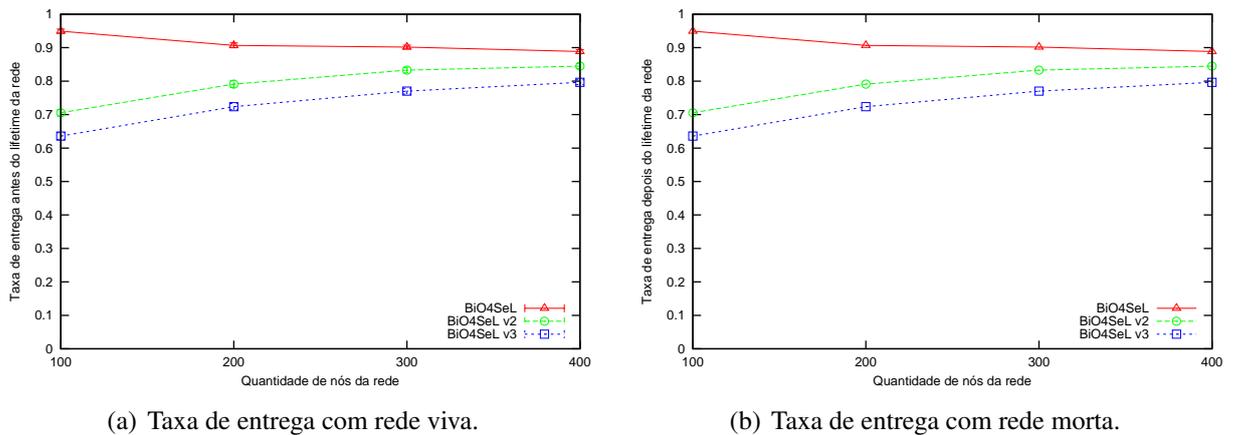


Figura 6.7: A taxa de entrega de pacotes.

6.3.7 Experimento 7: Quantidade de Nós Ativos pelo Tempo de Simulação

Este experimento foi realizado para mostrar a quantidade de nós que ficam no estado ativo pelo tempo de simulação. Como no BiO4SeL não é feito escalonamento dos nós, ele não é levado em consideração neste experimento.

As Figuras 6.8(a), 6.8(b), 6.8(c) e 6.8(d) mostram que, quando se usa o BiO4SeL v2, a quantidade de nós que permanecem ativos é basicamente a maioria dos nós da rede. Isso é explicado pelo fato de, no BiO4SeL v2, a escolha dos nós que mudam para o estado dormindo ser realizada levando em consideração somente a carga de trabalho de cada nó. Escolhendo os nós com base na carga de trabalho, o BiO4SeL v2 faz com que os nós que estejam sendo muito utilizados economizem sua energia, evitando que esse nós morram prematuramente e, conseqüentemente, aumentando o tempo de vida da rede. Como nestes cenários estão sendo usadas redes densas, a quantidade de nós dormindo representa uma taxa muito pequena dos nós da rede, fazendo com que a porcentagem de nós ativos fique extremamente perto dos 100%, como pode ser visto nos gráficos.

Já no BiO4SeL v3, que além de colocar para dormir os nós muito utilizados no roteamento, coloca também aqueles que nunca são utilizados; o número de nós dormindo é bem maior do que no BiO4SeL v2. Isso é mostrado nas figuras, que conseguem evidenciar bem a diferença entre o número de nós ativos do BiO4SeL v2 e v3. Como discutido na Seção ??, no entanto, colocar uma grande quantidade de nós para dormir nem sempre é a melhor solução e produz os melhores resultados. O BiO4SeL v2 mostra isso, pois, mesmo permanecendo com a maior parte dos nós da rede ativos, ele consegue apresentar os melhores resultados, como melhor tempo de

vida e distribuição do gasto energético.

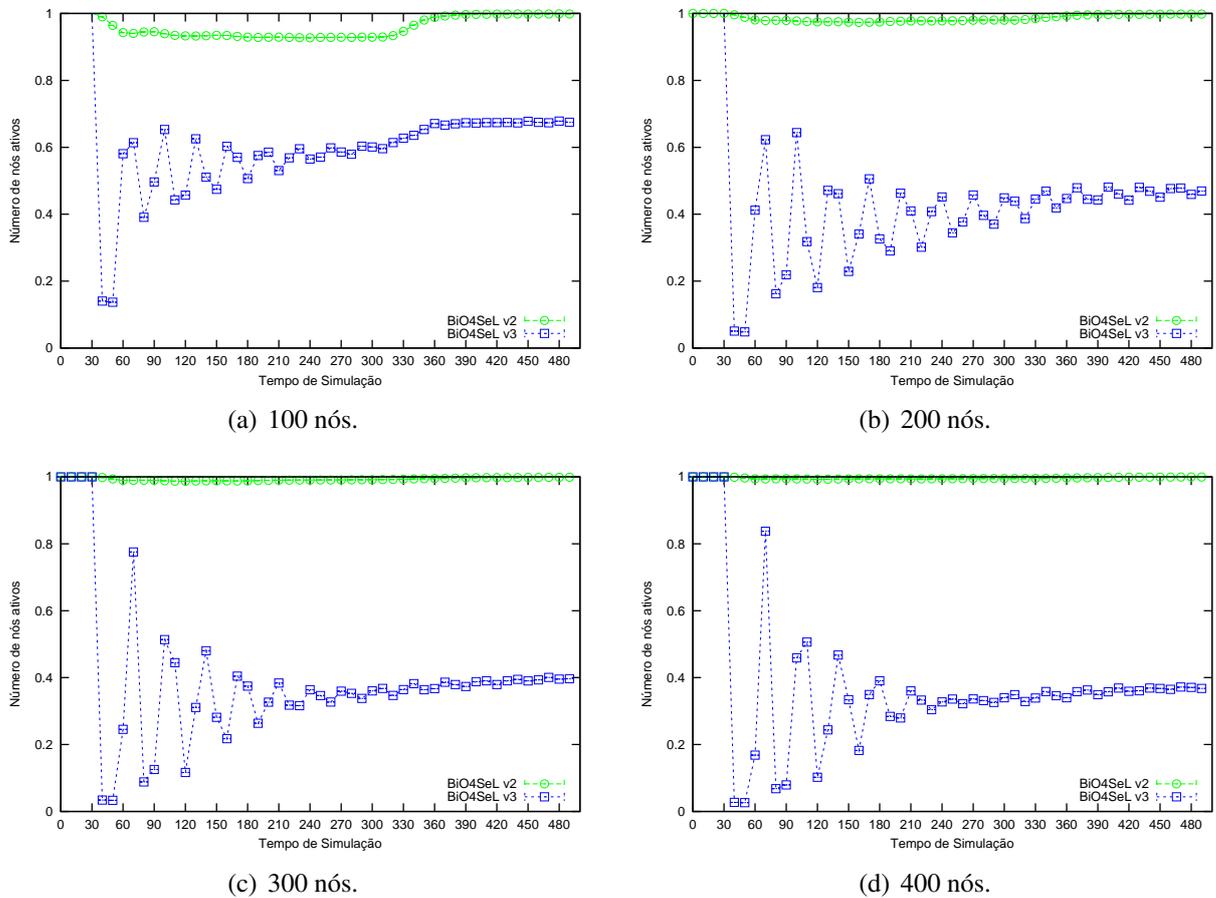


Figura 6.8: Quantidade de nós ativos pelo tempo.

6.3.8 Experimento 8: Energia dos Nós pelos Saltos à Estação Base

Os resultados deste experimento mostram um mapa da energia dos nós com relação à distância que eles estão do destino. Os valores de energia representados nos gráfico foram coletados no tempo de vida da rede e mostram como a energia é consumida até o tempo de morte do primeiro nó. Por meio dos gráficos, pode-se ver como a bateria é gasta entre os nós da rede de acordo com sua localização.

Como o cenário utilizado nas simulações é pequeno, $50m^2$, e o alcance de cada nó é de 15m, observa-se um gasto maior de energia nos nós que estão depois dos 15m e antes dos 40m. Isso acontece porque o BiO4SeL prioriza os caminhos mais curtos, de forma que os nós mais próximos à fonte e ao destino, que pertencem aos caminhos mais longo, são menos utilizados. Como o BiO4SeL v2 e v3 realizam o roteamento dos dados da mesma forma que o BiO4SeL, pode-se observar nas Figuras 6.9(a), 6.9(b), 6.9(c) e 6.9(d) que eles também priorizam os ca-

minhos mais curtos. No entanto, como eles realizam o escalonamento dos nós, conseguem distribuir melhor o gasto energético entre os nós localizados no intervalo (15,40), de forma que os nós neste intervalo apresentem maior energia quando comparado ao BiO4SeL. Nos gráficos, isso é percebido na observação da diferença entre os nós com menor e maior energia, que não é tão grande no BiO4seL v2 e v3 como é no BiO4SeL.

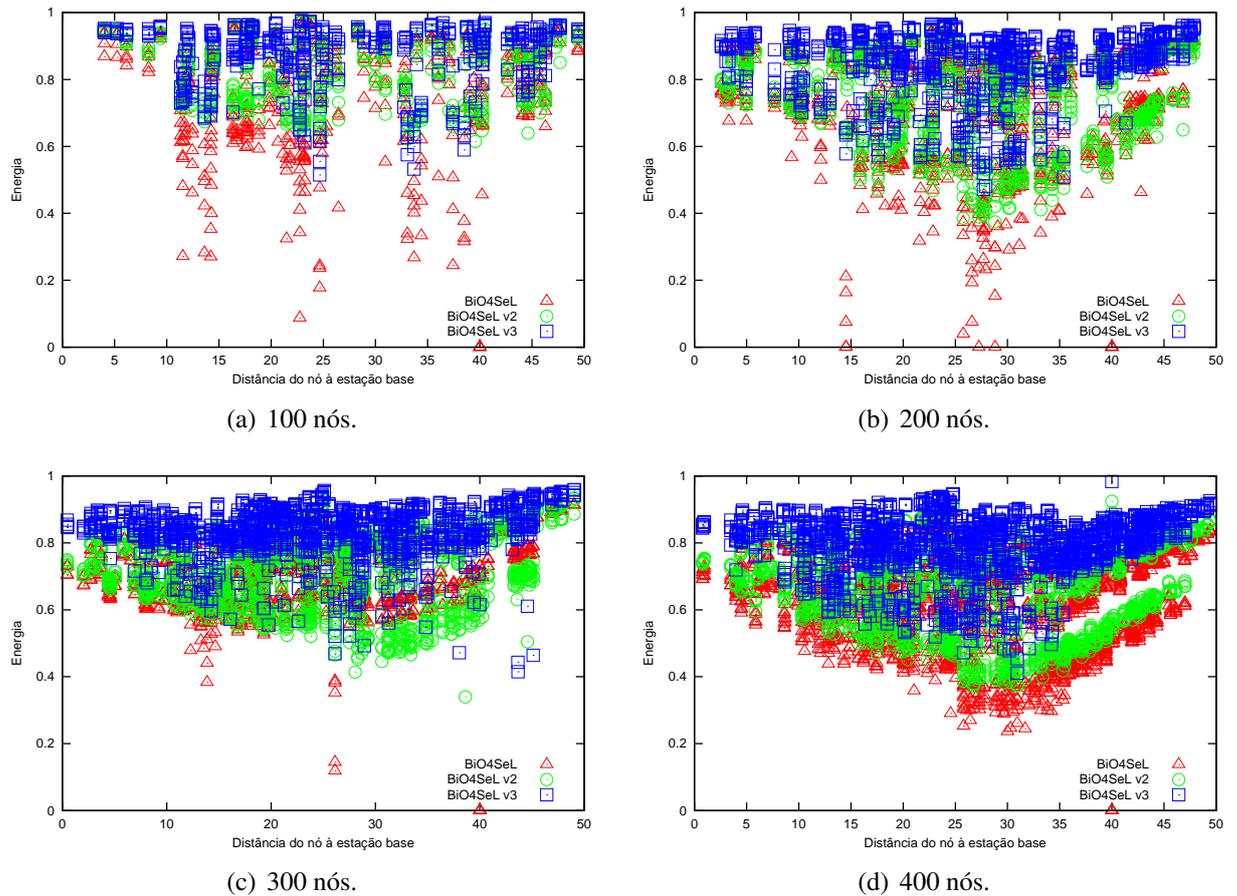


Figura 6.9: Energia pela distância ao destino.

7 Conclusão e Trabalhos Futuros

Esta dissertação propôs o BiO4SeL v2 e v3, mecanismos que realizam o gerenciamento autônomo da energia, mediante o escalonamento de atividade dos nós, no protocolo BiO4SeL. Na Seção 7.1, serão apresentadas as conclusões obtidas da análise feita no capítulo anterior. Na Seção 7.2, serão listadas as contribuições alcançadas com o desenvolvimento do trabalho apresentado nesta dissertação. E, para finalizar, na Seção 7.3, indicar-se-ão alguns trabalhos que podem ser realizados para melhorar ainda mais o desempenho dos mecanismos propostos.

7.1 Conclusões

Este trabalho foi proposto com o objetivo de implementar um mecanismo que realizasse o gerenciamento de energia nas RSSF, de forma autônoma, por via do escalonamento dos nós, no protocolo BiO4SeL. O mecanismo foi proposto para ser empregado em redes densas, que é onde o escalonamento de atividade obtém melhor resultado. A solução proposta usa a carga de trabalho de cada nó para realizar a escolha dos nós que mudarão de estado. Foi desenvolvida uma solução simples e autônoma para que não fosse gerada sobrecarga, e o tempo de vida da rede fosse maximizado.

Para isso, primeiramente, foi projetado o mecanismo de escalonamento, o BiO4SeL v2, para que os nós que estivessem sendo muito utilizados no encaminhamento de dados fossem poupados. Depois, o escalonamento BiO4SeL v2 foi integrado ao protocolo BiO4SeL, de forma que fosse acrescentado à rede o mínimo de sobrecarga. Como a proposta era desenvolver um mecanismo autônomo, o BiO4SeL v2 foi pensado para aproveitar toda a estrutura implementada pelo BiO4SeL, que é um protocolo, baseado em colônias de formigas, que faz o roteamento autônomo dos pacotes de dados.

Com o BiO4SeL v2, no entanto, a quantidade de nós que não estavam tendo sua energia gerenciada era muito grande. Para obter uma economia de energia ainda maior, o BiO4SeL v2 foi estendido, gerando o BiO4SeL v3. No BiO4SeL v3, a carga de trabalho continua sendo

o critério utilizado na escolha dos nós que vão mudar para o estado dormindo. O que mudou no BiO4SeL v3 é que, além dos nós que possuem uma carga de trabalho elevada, os nós que possuem carga de trabalho nula também podem mudar de estado.

Para avaliar o BiO4SeL v2 e v3, foram realizados vários experimentos fazendo-se a simulação dos três protocolos envolvidos neste trabalho. Com a análise dos experimentos, ficou evidente que o BiO4SeL v2 e v3 alcançaram seus principais objetivos, que eram melhorar o tempo de vida da rede e distribuir melhor o gasto de energia entre os nós. Além disso, o BiO4SeL v2 e v3 se mostraram simples, robusto e escalável, sem ocasionar sobrecarga na rede e ainda mantendo alta taxa de entrega do dados nos cenários muito densos. Uma conclusão importante obtida dos resultados é que a escolha dos nós que podem mudar de estado é um passo que exerce forte influência no desempenho do escalonamento dos nós. Isso foi mostrado pelo melhor desempenho do BiO4SeL v2 quando comparado ao BiO4SeL v3, na maioria dos experimentos realizados.

7.2 Contribuições

Os mecanismos de escalonamento de atividade dos nós usam diversas técnicas para realizar o gerenciamento de energia da rede. Dentre as técnicas mais utilizadas, encontram-se as que determinam o conjunto máximo de nós disjuntos, as baseadas em teoria dos grafos, as baseadas no alcance de sensoriamento etc. Estas soluções, no entanto, fazem uso de muitas informações não disponíveis nos nós, como localização e visão geral da rede. Além disso, muitas são implementadas de forma centralizada. Então, a primeira contribuição do BiO4SeL v2 e v3 está na sua capacidade de realizar o escalonamento de forma distribuída e autônoma, requisitos fundamentais para soluções de RSSF. O BiO4SeL v2 e v3 são distribuídos porque a mudança de estado é feita em cada nó, e é autônomo porque cada nó tem capacidade de decidir se pode ou não mudar de estado.

Outra contribuição do BiO4SeL v2 e v3 é sua simplicidade, robustez e escalabilidade. Na pesquisa bibliográfica feita para o desenvolvimento deste trabalho, encontrou-se alguns mecanismos que apresentam uma dessas características, porém não foi encontrado um trabalho preocupado em apresentar as três características. Como os mecanismos propostos foram completamente integrados ao BiO4SeL, eles herdam as características de robustez e escalabilidade apresentadas pelo BiO4SeL. Já a simplicidade está na forma como os nós que podem mudar de estado são escolhidos. O fato de não precisarem de informações de todos os nós da rede, e de cada nó se basear apenas na sua própria carga de trabalho, caracteriza a simplicidade dos

protocolos propostos.

Como consequência dessas contribuições, o BiO4SeL v2 e v3 conseguiram aumentar significativamente o tempo de vida da rede e melhorar a distribuição do gasto energético do BiO4SeL. Otimizar o tempo de vida da rede é a contribuição mais importante de um mecanismo de gerenciamento de energia e, como esse objetivo foi alcançado pela proposta deste trabalho, essa é também a principal contribuição desta dissertação.

As contribuições descritas nesta seção foram comprovadas por meio dos resultados dos experimentos realizados. O ganho no tempo de vida da rede pode ser visto claramente na Figura 6.1, resultante do experimento do tempo de vida pelo tempo de simulação. Já a melhora na distribuição do gasto energético pode ser verificada no experimento da distribuição energética pelo tempo de vida da rede. Além disso, o experimento que avalia os pacotes de sinalização gerados mostra que, no caso do BiO4SeL v3, a sobrecarga produzida pelos pacotes de sinalização é reduzida. O BiO4SeL v2, por sua vez, se mantém, em média, gerando a mesma quantidade de sinalização que o BiO4SeL; e ainda garante uma taxa de entrega de dados elevada, mesmo sem garantir 100% de conectividade da rede.

7.3 Trabalhos Futuros

No desenvolvimento desta dissertação, algumas questões foram negligenciadas em função do tempo disponível para sua implementação. Dessa forma, seguem alguns pontos que podem ser trabalhados para melhorar ainda mais o desempenho da solução proposta.

- Analisar o desempenho dos protocolos em cenários com mobilidade, pois eles oferecem suporte para mobilidade dos nós. Nem o BiO4SeL original, no entanto, foi testado ainda nesse cenário.
- Avaliar o comportamento do escalonamento dos nós quando realizado em redes heterogêneas.
- Realizar uma parametrização com as constantes definidas no BiO4SeL, de forma que sejam encontrados os valores que resultem numa distribuição do gasto energético mais igualitária, pois, mesmo sendo apresentada uma melhora no BiO4SeL v2 e v3, ainda está longe do ideal.
- Implementar uma heurística mais elaborada para realizar a escolha dos nós que vão mudar de estado quando apresentam carga de trabalho nula, pois, foi mostrado que a falta de uma boa heurística influencia diretamente na economia de energia obtida pelo mecanismo.

Referências Bibliográficas

ABDELGAWAD, A.; BAYOUMI, M. Remote measuring for sand in pipelines using wireless sensor network. *Instrumentation and Measurement, IEEE Transactions on*, v. 60, n. 4, p. 1443–1452, april 2011. ISSN 0018-9456.

ABE, R.; HONIDEN, S. Adaptive geographic routing in wireless sensor networks. In: *Proceedings of the 13th ACM international conference on Modeling, analysis, and simulation of wireless and mobile systems*. New York, NY, USA: ACM, 2010. (MSWIM '10), p. 91–100. ISBN 978-1-4503-0274-6. Disponível em: <<http://doi.acm.org/10.1145/1868521.1868538>>.

AKKAYA, K.; YOUNIS, M. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, v. 3, n. 3, p. 325 – 349, 2005. ISSN 1570-8705. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1570870503000738>>.

AKYILDIZ, I. et al. A survey on sensor networks. *Communications Magazine, IEEE*, v. 40, n. 8, p. 102 – 114, aug 2002. ISSN 0163-6804.

AKYILDIZ, I. et al. Wireless sensor networks: a survey. *Computer Networks*, v. 38, n. 4, p. 393 – 422, March 2002.

AL-KARAKI, J. N.; KAMAL, A. E. Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, v. 11, p. 6–28, 2004.

ALSALIH, W.; HASSANEIN, H.; AKL, S. Placement of multiple mobile data collectors in wireless sensor networks. *Ad Hoc Networks*, v. 8, n. 4, p. 378 – 390, 2010. ISSN 1570-8705. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1570870509000985>>.

AROLKAR, H. A.; SHETH, S. P.; TAMHANE, V. P. Ant colony based approach for intrusion detection on cluster heads in wsn. In: *Proceedings of the 2011 International Conference on Communication, Computing & Security*. New York, NY, USA: ACM, 2011. (ICCCS '11), p. 523–526. ISBN 978-1-4503-0464-1. Disponível em: <<http://doi.acm.org/10.1145/1947940.1948048>>.

AROLKAR, H. A.; SHETH, S. P.; TAMHANE, V. P. Ant colony based approach for intrusion detection on cluster heads in wsn. In: *Proceedings of the 2011 International Conference on Communication, Computing & Security*. New York, NY, USA: ACM, 2011. (ICCCS '11), p. 523–526. ISBN 978-1-4503-0464-1. Disponível em: <<http://doi.acm.org/10.1145/1947940.1948048>>.

AVOKH, A.; MIRJALILY, G. Dynamic balanced spanning tree (dbst) for data aggregation in wireless sensor networks. In: *Telecommunications (IST), 2010 5th International Symposium on*. [S.l.: s.n.], 2010. p. 391 –396.

- AZAD, A.; CHOCKALINGAM, A. Mobile base stations placement and energy aware routing in wireless sensor networks. In: *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*. [S.l.: s.n.], 2006. v. 1, p. 264 –269. ISSN 1525-3511.
- BAEG, S.-H. et al. Building a smart home environment for service robots based on rfid and sensor networks. In: *Control, Automation and Systems, 2007. ICCAS '07. International Conference on*. [S.l.: s.n.], 2007. p. 1078 –1082.
- BARBANCHO, J. et al. Using artificial intelligence in routing schemes for wireless networks. *Computer Communications*, v. 30, n. 14-15, p. 2802 – 2811, 2007. ISSN 0140-3664. Network Coverage and Routing Schemes for Wireless Sensor Networks. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0140366407002095>.
- BARRICO, C. M. C. S. *Optimização Evolucionária Multi-Objectivo em Ambientes Incertos - Pesquisa de Soluções Robustas*. Tese (Doutorado) — Universidade de Coimbra, Faculdade de Ciência e Tecnologia, 2007.
- BASHEER, I.; HAJMEER, M. Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods*, v. 43, n. 1, p. 3 – 31, 2000. ISSN 0167-7012. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167701200002013>.
- BHUIYAN, M.; GONDAL, I.; KAMRUZZAMAN, J. Lacar: Location aided congestion aware routing in wireless sensor networks. In: *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*. [S.l.: s.n.], 2010. p. 1 –6. ISSN 1525-3511.
- BING, S. et al. An ant-based on-demand energy routing protocol for ad hoc wireless networks. In: *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*. [S.l.: s.n.], 2007. p. 1516 –1519.
- BULUT, E.; KORPEOGLU, I. Sleep scheduling with expected common coverage in wireless sensor networks. *Wirel. Netw.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 17, p. 19–40, January 2011. ISSN 1022-0038. Disponível em: <http://dx.doi.org/10.1007/s11276-010-0262-2>.
- BUR, K.; OMIYI, P.; YANG, Y. Wireless sensor and actuator networks: Enabling the nervous system of the active aircraft. *Communications Magazine, IEEE*, v. 48, n. 7, p. 118 –125, July 2010. ISSN 0163-6804.
- C., L. M. A.; NASSER, N. Cluster-based routing protocol for mobile sensor networks. In: *Proceedings of the 3rd international conference on Quality of service in heterogeneous wired/wireless networks*. New York, NY, USA: ACM, 2006. (QShine '06). ISBN 1-59593-537-1. Disponível em: <http://doi.acm.org/10.1145/1185373.1185404>.
- CHEN, G. et al. An improved ant-based routing protocol in wireless sensor networks. In: *Collaborative Computing: Networking, Applications and Worksharing, 2006. CollaborateCom 2006. International Conference on*. [S.l.: s.n.], 2006. p. 1 –7.
- CHEN, J. et al. Modeling and extending lifetime of wireless sensor networks using genetic algorithm. In: *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2009. (GEC '09), p. 47–54. ISBN 978-1-60558-326-6. Disponível em: <http://doi.acm.org/10.1145/1543834.1543842>.

CHOI, W.-J.; CHOI, J.-K.; YOO, S.-J. Energy efficient channel sensing scheduling in cognitive radio networks. In: *Ubiquitous and Future Networks (ICUFN), 2011 Third International Conference on*. [S.l.: s.n.], 2011. p. 353 –357.

DIETRICH, I.; DRESSLER, F. On the lifetime of wireless sensor networks. *ACM Trans. Sen. Netw.*, ACM, New York, NY, USA, v. 5, p. 5:1–5:39, February 2009. ISSN 1550-4859. Disponível em: <<http://doi.acm.org/10.1145/1464420.1464425>>.

DILMAGHANI, R. et al. Wireless sensor networks for monitoring physiological signals of multiple patients. *Biomedical Circuits and Systems, IEEE Transactions on*, v. 5, n. 4, p. 347 –356, aug. 2011. ISSN 1932-4545.

DORIGO, M.; MANIEZZO, V.; COLORNI, A. Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, v. 26, n. 1, p. 29 –41, feb 1996. ISSN 1083-4419.

DRESSLER, F.; AKAN, O. Bio-inspired networking: from theory to practice. *Communications Magazine, IEEE*, v. 48, n. 11, p. 176 –183, november 2010. ISSN 0163-6804.

FRANCESCO, M. D.; DAS, S. K.; ANASTASI, G. Data collection in wireless sensor networks with mobile elements: A survey. *ACM Transactions on Sensor Networks*, ACM, New York, NY, USA, v. 8, n. 1, aug 2011. ISSN 1550-4859. Disponível em: <<http://dx.doi.org/10.1145/1993042.1993049>>.

GARCÍA-HERNÁNDEZ PABLO H. IBARGÜENGOYTIA-GONZÁLEZ, J. G.-H. J. A. P.-D. C. F. Wireless sensor networks and applications: a survey. *IJCSNS International Journal of Computer Science and Network Security*, v. 7. ISSN 1738-7906.

GHAFFARIYAN, P. An effective data aggregation mechanism for wireless sensor networks. In: *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on*. [S.l.: s.n.], 2010. p. 1 –4.

GUAN, X. et al. A novel routing algorithm based on ant colony system for wireless sensor networks. In: *Proceedings of the 2009 Proceedings of 18th International Conference on Computer Communications and Networks*. Washington, DC, USA: IEEE Computer Society, 2009. (ICCCN '09), p. 1–5. ISBN Bad - remove. Disponível em: <<http://dx.doi.org/10.1109/ICCCN.2009.5235245>>.

GUO, W.; ZHANG, W.; LU, G. A comprehensive routing protocol in wireless sensor network based on ant colony algorithm. In: *Proceedings of the 2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing - Volume 01*. Washington, DC, USA: IEEE Computer Society, 2010. (NSWCTC '10), p. 41–44. ISBN 978-0-7695-4011-5. Disponível em: <<http://dx.doi.org/10.1109/NSWCTC.2010%-.18>>.

HAAG, C. R. et al. An artificial immune system-inspired multiobjective evolutionary algorithm with application to the detection of distributed computer network intrusions. In: *Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation*. New York, NY, USA: ACM, 2007. (GECCO '07), p. 2717–2724. ISBN 978-1-59593-698-1. Disponível em: <<http://doi.acm.org/10.1145/1274000.1274035>>.

- HEIMFARTH, T.; JANACIK, P. Experiments with biologically-inspired methods for service assignment in wireless sensor networks. In: HINCHEY, M. et al. (Ed.). *Biologically-Inspired Collaborative Computing*. Springer Boston, 2008, (IFIP International Federation for Information Processing, v. 268). p. 71–84. ISBN 978-0-387-09654-4. 10.1007/978-0-387-09655-1_7. Disponível em: http://dx.doi.org/10.1007/978-0-387-09655-1_7.
- HEINZELMAN, W. R.; CHANDRAKASAN, A.; BALAKRISHNAN, H. Energy-efficient communication protocol for wireless microsensor networks. In: *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8 - Volume 8*. Washington, DC, USA: IEEE Computer Society, 2000. (HICSS '00), p. 8020–. ISBN 0-7695-0493-0. Disponível em: <http://dl.acm.org/citation.cfm?id=820264.820485>.
- HEINZELMAN, W. R.; KULIK, J.; BALAKRISHNAN, H. Adaptive protocols for information dissemination in wireless sensor networks. In: *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. New York, NY, USA: ACM, 1999. (MobiCom '99), p. 174–185. ISBN 1-58113-142-9. Disponível em: <http://doi.acm.org/10.1145/313451.313529>.
- HOFMEYR, S. A.; FORREST, S. A. Architecture for an artificial immune system. *Evol. Comput.*, MIT Press, Cambridge, MA, USA, v. 8, p. 443 – 473, December 2000. ISSN 1063 - 6560. Disponível em: <http://dx.doi.org/10.1162/106365600568257>.
- HUANG, C.-F.; TSENG, Y.-C. The coverage problem in a wireless sensor network. In: *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*. New York, NY, USA: ACM, 2003. (WSNA '03), p. 115–121. ISBN 1-58113-764-8. Disponível em: <http://doi.acm.org/10.1145/941350.941367>.
- HUSSEIN, O.; SAADAWI, T. Ant routing algorithm for mobile ad-hoc networks (arama). In: *Performance, Computing, and Communications Conference, 2003. Conference Proceedings of the 2003 IEEE International*. [S.l.: s.n.], 2003. p. 281 – 290. ISSN 1097-2641.
- ISHMANOV, F.; KIM, S. W.; KIM, B.-S. Energy consumption balancing in wireless sensor networks. In: *Computer Applications and Industrial Electronics (ICCAIE), 2010 International Conference on*. [S.l.: s.n.], 2010. p. 255 –260.
- IWANICKI, K.; STEEN, M. van. On hierarchical routing in wireless sensor networks. In: *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*. Washington, DC, USA: IEEE Computer Society, 2009. (IPSN '09), p. 133–144. ISBN 978-1-4244-5108-1. Disponível em: <http://dl.acm.org/citation.cfm?id=1602165%-.1602179>.
- JAIN, R. Book. *The Art of Computer Systems Performance Analysis : Techniques for Experimental Design, Measurement, Simulation, and Modeling / Raj Jain*. Wiley, New York :, 1991. xxvii, 685 p. : p. ISBN 0471503363. Disponível em: <http://www.loc.gov/catdir/toc-onix04/90045479.html>.
- JIANG, P. et al. A study of routing protocols in wireless sensor networks. In: *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*. [S.l.: s.n.], 2006. v. 1, p. 266 –270.

- KANDRIS, D. et al. Energy efficient and perceived qos aware video routing over wireless multimedia sensor networks. *Ad Hoc Netw.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 9, p. 591–607, June 2011. ISSN 1570-8705. Disponível em: <http://dx.doi.org/10.1016/j.adhoc.2010.09.001>.
- KASIRAJAN, P.; LARSEN, C.; JAGANNATHAN, S. A new adaptive compression scheme for data aggregation in wireless sensor networks. In: *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*. [S.l.: s.n.], 2010. p. 1–6. ISSN 1525-3511.
- KONSTANTINIDIS, A. et al. Energy-aware topology control for wireless sensor networks using memetic algorithms. *Comput. Commun.*, Butterworth-Heinemann, Newton, MA, USA, v. 30, p. 2753–2764, October 2007. ISSN 0140-3664. Disponível em: <http://dx.doi.org/10.1016/j.comcom.2007.05.013>.
- KOTOV, V. D.; VASILYEV, V. Immune model based approach for network intrusion detection. In: *Proceedings of the 3rd international conference on Security of information and networks*. New York, NY, USA: ACM, 2010. (SIN '10), p. 233–237. ISBN 978-1-4503-0234-0. Disponível em: <http://doi.acm.org/10.1145/1854099.1854146>.
- KOTOV, V. D.; VASILYEV, V. I. Artificial immune system based intrusion detection system. In: *Proceedings of the 2nd international conference on Security of information and networks*. New York, NY, USA: ACM, 2009. (SIN '09), p. 207–212. ISBN 978-1-60558-412-6. Disponível em: <http://doi.acm.org/10.1145/1626195.1626247>.
- KULKARNI, R.; ANDRSTER, A. F.; VENAYAGAMOORTHY, G. Computational intelligence in wireless sensor networks: A survey. *Communications Surveys Tutorials, IEEE*, v. 13, n. 1, p. 68–96, quarter 2011. ISSN 1553-877X.
- KUMAR, K. Imcc protocol in heterogeneous wireless sensor network for high quality data transmission in military applications. In: *Parallel Distributed and Grid Computing (PDGC), 2010 1st International Conference on*. [S.l.: s.n.], 2010. p. 339–343.
- KWONG, K. et al. Implementation of herd management systems with wireless sensor networks. *Wireless Sensor Systems, IET*, v. 1, n. 2, p. 55–65, june 2011. ISSN 2043-6386.
- LEE, A.; RA, I. Adaptive-gossiping for an energy-aware routing protocol in wireless sensor networks. In: *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*. New York, NY, USA: ACM, 2010. (IWCMC '10), p. 1131–1135. ISBN 978-1-4503-0062-9. Disponível em: <http://doi.acm.org/10.1145/1815396.1815655>.
- LI, D.; LIU, W.; CUI, L. Easidesign: An improved ant colony algorithm for sensor deployment in real sensor network system. In: *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*. [S.l.: s.n.], 2010. p. 1–5. ISSN 1930-529X.
- LI, Y. et al. Deploying wireless sensors for differentiated coverage and probabilistic connectivity. In: *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*. [S.l.: s.n.], 2010. p. 1–6. ISSN 1525-3511.
- LIAO, W.-H.; KAO, Y.; FAN, C.-M. Data aggregation in wireless sensor networks using ant colony algorithm. *J. Netw. Comput. Appl.*, Academic Press Ltd., London, UK, UK, v. 31, p. 387–401, November 2008. ISSN 1084-8045. Disponível em: <http://dl.acm.org/citation.cfm?id=1399643-.1399678>.

LIN, Y.; HU, X.-m.; ZHANG, J. An ant-colony-system-based activity scheduling method for the lifetime maximization of heterogeneous wireless sensor networks. In: *Proceedings of the 12th annual conference on Genetic and evolutionary computation*. New York, NY, USA: ACM, 2010. (GECCO '10), p. 23–30. ISBN 978-1-4503-0072-8. Disponível em: <http://doi.acm.org/10.1145/1830483.1830488>.

LINDSEY, S.; RAGHAVENDRA, C. Pegasus: Power-efficient gathering in sensor information systems. In: *Aerospace Conference Proceedings, 2002. IEEE*. [S.l.: s.n.], 2002. v. 3, p. 3–1125 – 3–1130 vol.3.

LOTF, J.; GHAZANI, S. Overview on routing protocols in wireless sensor networks. In: *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*. [S.l.: s.n.], 2010. v. 3, p. V3–610 –V3–614.

LOTF, J.; HOSSEINZADEH, M.; ALGULIEV, R. Hierarchical routing in wireless sensor networks: a survey. In: *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*. [S.l.: s.n.], 2010. v. 3, p. V3–650 –V3–654.

LUO, J.; HUBAUX, J.-P. Joint sink mobility and routing to maximize the lifetime of wireless sensor networks: The case of constrained mobility. *Networking, IEEE/ACM Transactions on*, v. 18, n. 3, p. 871 –884, june 2010. ISSN 1063-6692.

LUQIAO, Z.; QINXIN, Z. Interference and energy aware topology control. In: *Consumer Electronics, Communications and Networks (CECNet), 2011 International Conference on*. [S.l.: s.n.], 2011. p. 1357 –1359.

MAHFOUDH, S.; MINET, P. Survey of energy efficient strategies in wireless ad hoc and sensor networks. In: *Networking, 2008. ICN 2008. Seventh International Conference on*. [S.l.: s.n.], 2008. p. 1 –7.

MANOLOPOULOS, Y.; KATSAROS, D.; PAPADIMITRIOU, A. Topology control algorithms for wireless sensor networks: a critical survey. In: *Proceedings of the 11th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing on International Conference on Computer Systems and Technologies*. New York, NY, USA: ACM, 2010. (CompSysTech '10), p. 1–10. ISBN 978-1-4503-0243-2. Disponível em: <http://doi.acm.org/10.1145/1839379.1839381>.

MENG, F. et al. Energy-efficient and coverage-specific node scheduling for wireless sensor networks. In: *Proceedings of the 13th ACM international conference on Modeling, analysis, and simulation of wireless and mobile systems*. New York, NY, USA: ACM, 2010. (MSWIM '10), p. 368–375. ISBN 978-1-4503-0274-6. Disponível em: <http://doi.acm.org/10.1145/1868521.1868582>.

NAMI, M.; BERTELS, K. A survey of autonomic computing systems. In: *Autonomic and Autonomous Systems, 2007. ICAS07. Third International Conference on*. [S.l.: s.n.], 2007. p. 26.

NGUYEN, N. D. et al. Energy-efficient models for coverage problem using sensors with adjustable sensing ranges. In: *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*. [S.l.: s.n.], 2010. p. 1 –6. ISSN 1525-3511.

NIST. *The Network Simulator NS-2 NIST add-on IEEE 802.21 Model*. jan 2007. Develop by The National Institute of Standards and Technology (NIST). Disponível em: <<http://w3.antd.nist.gov/seamlessandsecure%20-%20pubtool.shtml>>.

PAONE, M. et al. A bio-inspired distributed routing protocol for wireless sensor networks: Performance evaluation. In: *Proceedings of the 2010 IEEE 30th International Conference on Distributed Computing Systems Workshops*. Washington, DC, USA: IEEE Computer Society, 2010. (ICDCSW '10), p. 247–255. ISBN 978-0-7695-4079-5. Disponível em: <<http://dx.doi.org/10.1109/ICDCSW.2010%20-%2066>>.

PAPADOPOULOS, A. et al. Vibe: An energy efficient routing protocol for dense and mobile sensor networks. *Journal of Network and Computer Applications*, In Press, Corrected Proof, p. –, 2011. ISSN 1084-8045. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1084804511001068>>.

PAVAI, K.; SIVAGAMI, A.; SRIDHARAN, D. Study of routing protocols in wireless sensor networks. In: *Advances in Computing, Control, Telecommunication Technologies, 2009. ACT '09. International Conference on*. [S.l.: s.n.], 2009. p. 522 –525.

PAZAND, B.; DATTA, A. An energy-efficient node-scheduling scheme for wireless sensor networks based on minimum dominating sets. *Int. J. Netw. Manag.*, John Wiley & Sons, Inc., New York, NY, USA, v. 19, p. 75–99, February 2009. ISSN 1099-1190. Disponível em: <<http://dx.doi.org/10.1002/nem.689>>.

PERKINS, C.; ROYER, E. Ad-hoc on-demand distance vector routing. In: *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*. [S.l.: s.n.], 1999. p. 90 –100.

RAJAGOPALAN, R.; VARSHNEY, P. Data-aggregation techniques in sensor networks: a survey. *Communications Surveys Tutorials, IEEE*, v. 8, n. 4, p. 48 –63, quarter 2006. ISSN 1553-877X.

REHMAN, A. et al. A review of wireless sensors and networks' applications in agriculture. *Computer Standards & Interfaces*, n. 0, p. –, 2011. ISSN 0920-5489. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0920548911000353>>.

RIBEIRO, L.; CASTRO, M. de. Bio4sel: A bio-inspired routing algorithm for sensor network lifetime optimization. In: *Telecommunications (ICT), 2010 IEEE 17th International Conference on*. [S.l.: s.n.], 2010. p. 728 –734.

RIBEIRO, L.; CASTRO, M. de. Bio4sel: A bio-inspired routing algorithm for sensor network lifetime optimization. In: *Telecommunications (ICT), 2010 IEEE 17th International Conference on*. [S.l.: s.n.], 2010. p. 728 –734.

RIBEIRO, L. B. *BiO4SeL: Uma Abordagem Baseada em Colônia de Formigas para a Otimização do Tempo de Vida de Redes de Sensores Sem Fio*. Dissertação (Mestrado) — Universidade Federal do Ceará, Centro de Ciência, Mestrado e Doutorado em Ciência da Computação (MDCC), 2009.

RODRIGUES, J. J.; PEREIRA, O. R.; NEVES, P. A. Biofeedback data visualization for body sensor networks. *Journal of Network and Computer Applications*, v. 34, n. 1, p. 151 – 158,

2011. ISSN 1084-8045. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1084804510001530>>.

RU, H.; XU, G. Swarm intelligence-inspired adaptive routing construction in wsn. In: *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on*. [S.l.: s.n.], 2010. p. 1 –5.

SALEEM, M.; CARO, G. A. D.; FAROOQ, M. Swarm intelligence based routing protocol for wireless sensor networks: Survey and future directions. *Inf. Sci.*, Elsevier Science Inc., New York, NY, USA, v. 181, p. 4597–4624, October 2011. ISSN 0020-0255. Disponível em: <<http://dx.doi.org/10.1016/j.ins.2010.07.005>>.

SALEEM, M.; CARO, G. A. D.; FAROOQ, M. Swarm intelligence based routing protocol for wireless sensor networks: Survey and future directions. *Inf. Sci.*, Elsevier Science Inc., New York, NY, USA, v. 181, p. 4597–4624, October 2011. ISSN 0020-0255. Disponível em: <<http://dx.doi.org/10.1016/j.ins.2010.07.005>>.

SANTI, P. Topology control in wireless ad hoc and sensor networks. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 37, p. 164–194, June 2005. ISSN 0360-0300. Disponível em: <<http://doi.acm.org/10.1145/1089733.1089736>>.

SARAVANAKUMAR, R.; SUSILA, S.; RAJA, J. An energy efficient cluster based node scheduling protocol for wireless sensor networks. In: *Solid-State and Integrated Circuit Technology (ICSICT), 2010 10th IEEE International Conference on*. [S.l.: s.n.], 2010. p. 2053–2057.

SONG, S.; CHOI, B.-Y.; KIM, D. Mr. bin: Multicast routing with branch information nodes for wireless sensor networks. In: *Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on*. [S.l.: s.n.], 2010. p. 1 –6. ISSN 1095-2055.

SUN, L.-Y.; CAI, W.; HUANG, X.-X. Data aggregation scheme using neural networks in wireless sensor networks. In: *Future Computer and Communication (ICFCC), 2010 2nd International Conference on*. [S.l.: s.n.], 2010. v. 1, p. V1–725 –V1–729.

SWAIN, A.; HANSDAH, R.; CHOUHAN, V. An energy aware routing protocol with sleep scheduling for wireless sensor networks. In: *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*. [S.l.: s.n.], 2010. p. 933–940. ISSN 1550-445X.

TANG, L. et al. Pw-mac: An energy-efficient predictive-wakeup mac protocol for wireless sensor networks. In: *INFOCOM, 2011 Proceedings IEEE*. [S.l.: s.n.], 2011. p. 1305 –1313. ISSN 0743-166X.

TIAN, D.; GEORGANAS, N. D. A node scheduling scheme for energy conservation in large wireless sensor networks. *Wireless Communications and Mobile Computing*, John Wiley & Sons, Ltd., v. 3, n. 2, p. 271–290, 2003. ISSN 1530-8677. Disponível em: <<http://dx.doi.org/10.1002/wcm.116>>.

VAIDEHI, V. et al. Redundant node deactivation by scheduling in wireless sensor networks. In: *Recent Trends in Information Technology (ICRTIT), 2011 International Conference on*. [S.l.: s.n.], 2011. p. 613 –617.

WANG, H.; LUO, N. An improved ant-based algorithm for data aggregation in wireless sensor networks. In: *Proceedings of the 2010 International Conference on Communications and Mobile Computing - Volume 03*. Washington, DC, USA: IEEE Computer Society, 2010. (CMC '10), p. 239–243. ISBN 978-0-7695-3989-8. Disponível em: <http://dx.doi.org/10.1109/CMC-2010.183>.

WANG, L.; SUN, Q.; MA, H. Energy consumption optimize based on ant colony algorithm for wireless sensor networks. In: *Proceedings of the 2nd international Asia conference on Informatics in control, automation and robotics - Volume 1*. Piscataway, NJ, USA: IEEE Press, 2010. (CAR'10), p. 17–21. ISBN 978-1-4244-5192-0. Disponível em: <http://dl.acm.org/citation.cfm?id=1843162%-.1843167>.

WANG, L.; XIAO, Y. A survey of energy-efficient scheduling mechanisms in sensor networks. *Mob. Netw. Appl.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 11, p. 723–740, October 2006. ISSN 1383-469X. Disponível em: <http://dx.doi.org/10.1007/s11036-006-7798-5>.

WANG, S.-Y. et al. Preserving target area coverage in wireless sensor networks by using computational geometry. In: *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*. [S.l.: s.n.], 2010. p. 1–6. ISSN 1525-3511.

WU, K. et al. Lightweight deployment-aware scheduling for wireless sensor networks. *Mob. Netw. Appl.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 10, p. 837–852, December 2005. ISSN 1383-469X. Disponível em: <http://dx.doi.org/10.1007/s11036-005-4442-8>.

WU, X.; CHEN, G.; CHEN, J. Energy-efficient and topology-aware routing for underwater sensor networks. In: *Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on*. [S.l.: s.n.], 2010. p. 1–6. ISSN 1095-2055.

XING, G. et al. Co-grid: an efficient coverage maintenance protocol for distributed sensor networks. In: *Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on*. [S.l.: s.n.], 2004. p. 414–423.

YANG, J. et al. Ant colony-based multi-path routing algorithm for wireless sensor networks. In: *Intelligent Systems and Applications, 2009. ISA 2009. International Workshop on*. [S.l.: s.n.], 2009. p. 1–4.

YANG, O.; HEINZELMAN, W. A better choice for sensor sleeping. In: *Proceedings of the 6th European Conference on Wireless Sensor Networks*. Berlin, Heidelberg: Springer-Verlag, 2009. (EWSN '09), p. 134–149. ISBN 978-3-642-00223-6.

YANG, Y.; FONOAGE, M. I.; CARDEI, M. Improving network lifetime with mobile wireless sensor networks. *Comput. Commun.*, Butterworth-Heinemann, Newton, MA, USA, v. 33, p. 409–419, March 2010. ISSN 0140-3664. Disponível em: <http://dx.doi.org/10.1016/j.comcom.2009.11.010>.

YARDIBI, T.; KARASAN, E. A distributed activity scheduling algorithm for wireless sensor networks with partial coverage. *Wirel. Netw.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 16, p. 213–225, January 2010. ISSN 1022-0038. Disponível em: <http://doi.org/10.1007/s11276-008-0125-2>.

- YE, F. et al. Peas: a robust energy conserving protocol for long-lived sensor networks. In: *Network Protocols, 2002. Proceedings. 10th IEEE International Conference on*. [S.l.: s.n.], 2002. p. 200 – 201. ISSN 1092-1648.
- YICK, J.; MUKHERJEE, B.; GHOSAL, D. Wireless sensor network survey. *Comput. Netw.*, Elsevier North-Holland, Inc., New York, NY, USA, v. 52, p. 2292–2330, August 2008. ISSN 1389-1286. Disponível em: <http://dl.acm.org/citation.cfm?id=1389582%-.1389832>.
- YOON, S. et al. Swats: Wireless sensor networks for steamflood and waterflood pipeline monitoring. *Network, IEEE*, v. 25, n. 1, p. 50 –56, january-february 2011. ISSN 0890-8044.
- YU, J.; NOEL, E.; TANG, K. Degree constrained topology control for very dense wireless sensor networks. In: *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*. [S.l.: s.n.], 2010. p. 1 –6. ISSN 1930-529X.
- YU, Y.; GOVINDAN, R.; ESTRIN, D. *Geographical and Energy Aware Routing: a recursive data dissemination protocol for wireless sensor networks*. [S.l.], 2001.
- ZHANG, Z.; MAO, G.; ANDERSON, B. On the effective energy consumption in wireless sensor networks. In: *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*. [S.l.: s.n.], 2010. p. 1 –6. ISSN 1525-3511.
- ZHAO, J.; GOVINDAN, R. Understanding packet delivery performance in dense wireless sensor networks. In: *Proceedings of the 1st international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2003. (SenSys '03), p. 1–13. ISBN 1-58113-707-9. Disponível em: <http://doi.acm.org/10.1145/958491.958493>.
- ZHAO, Y. et al. An energy-efficient and low-latency mac protocol with adaptive scheduling for multi-hop wireless sensor networks. *Computer Communications*, v. 33, n. 12, p. 1452 – 1461, 2010. ISSN 0140-3664. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0140366410001465>.