



UNIVERSIDADE ESTADUAL DO CEARÁ

ANTONIEL DA SILVA RÊGO

**BEE-C: UM ALGORITMO DE ROTEAMENTO
BIO-INSPIRADO PARA ECONOMIA DE ENERGIA EM
REDES DE SENSORES SEM FIO**

FORTALEZA - CEARÁ

2011

ANTONIEL DA SILVA RÊGO

**BEE-C: UM ALGORITMO DE ROTEAMENTO BIO-INSPIRADO PARA
ECONOMIA DE ENERGIA EM REDES DE SENSORES SEM FIO**

Dissertação apresentada no Curso de Mestrado Acadêmico em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Joaquim Celestino
Júnior

FORTALEZA - CEARÁ

2011

C824p	<p>Rêgo, Antoniel da Silva.</p> <p>BEE-C: Um Algoritmo de Roteamento Bio-inspirado para Economia de Energia em Redes de Sensores Sem Fio / Antoniel da Silva Rêgo. – Fortaleza, 2011.</p> <p>75 p.;il.</p> <p>Orientador: Prof. Dr. Prof. Dr. Joaquim Celestino Júnior</p> <p>Dissertacao (Mestrado Acadêmico em Ciência da Computação) - Universidade Estadual do Ceará, Centro de Ciências e Tecnologia.</p> <p>1. Sensores 2. Bioinspirados 3. Clusterização 4. Energia 5. I. Universidade Estadual do Ceará, Centro de Ciências e Tecnologia.</p>
-------	---

CDD:001.6

ANTONIEL DA SILVA RÊGO

**BEE-C: UM ALGORITMO DE ROTEAMENTO BIO-INSPIRADO PARA
ECONOMIA DE ENERGIA EM REDES DE SENSORES SEM FIO**

Dissertação apresentada no Curso de Mestrado Acadêmico em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial para obtenção do grau de Mestre.

Aprovada em: __/__/____

BANCA EXAMINADORA

Prof. Dr. Joaquim Celestino Júnior
Universidade Estadual do Ceará – UECE
Orientador

Prof. Dr. Gerardo Valdísio Rodrigues Viana
Universidade Estadual do Ceará – UECE

Prof. Dr. Elias Teodoro Silva Júnior
Instituto Federal de Educação, Ciência e
Tecnologia do Ceará – IFCE

Prof. Dr. Eduardo Coelho Cerqueira
Universidade Federal do Pará – UFPA

AGRADECIMENTOS

Ao meu orientador, professor Celestino, pois sua participação neste trabalho foi fundamental, com incentivo, sugestões, apoio moral. Agradeço por ter acreditado e sempre ter me dado suporte, que foi essencial para este trabalho. Sou grato pelo tempo dedicado, competência, correções, revisões e pelas discussões, que foram bastante produtivas.

Ao professor Gustavo pelo incentivo, colaboração e conhecimento transmitido na parte de algoritmos bio-inspirados.

A todos os professores, colegas e amigos do MACC (Mestrado Acadêmico em Ciência da Computação), pelo convívio diário e por proporcionarem um ambiente agradável de estudo.

À CAPES (Coordenação de Aperfeiçoamento Pessoal de Nível Superior), pela bolsa concedida pelo período de 24 meses, sem a qual não teria sido possível a realização deste trabalho.

À UECE (Universidade Estadual do Ceará), por me acolher em 2009, como mestrando, e contribuir muito para a minha evolução, não apenas como estudante, mas também como pessoa. Farei o possível para repassar para a sociedade todo o conhecimento aqui adquirido.

Ao professor Eduardo Cerqueira e à UFPA (Universidade Federal do Pará) por terem disponibilizado espaço no laboratório do GERCOM (Research Group on Computer Networks and Multimedia Communication), no qual pude finalizar este trabalho.

Ao meu amigo e colega de pesquisa Neuton, pelo convívio, por ter me ajudado no início deste trabalho com ideias, pesquisas, críticas e discussões. Foi muito valiosa sua participação.

Aos meus colegas e amigos Fábio, Alysson, Walisson, Alisson e Sergio Vieira, e a todos os colegas de pesquisa do LARCES (Laboratório de Redes de Computadores e Segurança) pelo convívio na UECE por estes dois anos de mestrado.

Aos meus pais, Raimunda e Raimundo, através dos quais recebi o dom mais precioso do universo: a vida. Somente por isso seria infinitamente grato. Agradeço por me ensinarem a viver com dignidade e respeito. Mesmo em meio a muitas dificuldades, abriram a porta do meu futuro, me ensinando que o estudo seria a maneira mais viável para meu crescimento pessoal e profissional. Trabalharam dobrado, sacrificaram seus sonhos em favor dos meus, não são somente pais, mas amigos e companheiros.

Aos meus tios, Expedita e João, que desde os tempos da graduação têm me dado força e me apoiado nos estudos.

À minha namorada e amiga, Pathy Ingrid, por seu companheirismo e apoio durante todo o mestrado. Sua ajuda foi essencial para a conclusão deste trabalho.

À minha amiga Maria José (dona Mazé), por ter me acolhido em Fortaleza no início do mestrado. Toda minha gratidão à sua prestatividade e amizade.

Às amigas feitas em Fortaleza, em especial à Lizabetha, Luís Filho, Dona Cristina e Ana Joyce, que me ajudaram a adaptar-me na cidade através de passeios, almoços e descontrações,

suas amizades foram formidáveis.

Ao meu amigo David, grande parceiro de futebol em Fortaleza.

A todos que direta e indiretamente tiveram participação neste trabalho.

Agradeço principalmente a Deus, por sua presença constante em minha vida, sem que eu precise pedir, pelo auxílio em minhas escolhas e pelo conforto e força nos momentos difíceis.

*“Entrega o teu caminho ao Senhor,
confia nele e o mais Ele fará’
Salmos 37:5*

RESUMO

Este trabalho apresenta um algoritmo de roteamento para Redes de Sensores Sem Fio (BEE-C) baseado no LEACH (Low-Energy Adaptive Clustering Hierarchy) e no LEACH-C (LEACH Centralizado). O BEE-C, assim como o LEACH-C, foi desenvolvido para redes de sensores que coletam informações periodicamente durante o tempo de vida. Para redes deste tipo, o tempo de vida da rede é um fator muito importante, principalmente pela dificuldade de manutenção e pela limitação de bateria, memória e processamento que os sensores possuem. O BEE-C é um protocolo que utiliza um algoritmo bio-inspirado em abelhas para realizar uma clusterização eficiente dos nós da rede. Os resultados das simulações, comparados ao LEACH e LEACH-C, mostram que o BEE-C obteve uma melhora significativa no que se refere ao tempo de vida e quantidade de pacotes enviados à estação base. Além disso, os resultados apresentam um consumo de energia mais eficiente por parte do BEE-C comparado com os demais algoritmos.

Palavras-Chave: Sensores. Bioinspirados. Clusterização. Energia.

LISTA DE FIGURAS

Figura 1	Arquitetura do nó sensor	19
Figura 2	Um exemplo de redes de sensores sem fio	19
Figura 3	Vantagem do uso de comunicação multi-hop	27
Figura 4	Comunicação direta dos nós para a estação base	27
Figura 5	Comunicação multihop dos nós para a estação base	28
Figura 6	Rede clusterizada, somente os líderes enviam os dados para a estação base ..	29
Figura 7	Exemplo de representação de uma solução	31
Figura 8	Algoritmo HBMO	32
Figura 9	Representação do algoritmo HBMO	33
Figura 10	Formação dos grupos no protocolo LEACH	37
Figura 11	Fase de comunicação do protocolo LEACH	38
Figura 12	Cenário em que o líder do cluster 1 se recusa a servir de ponte para o cluster 2.	41
Figura 13	Fase de formação de grupos do BEE-C	46
Figura 14	Algoritmo de formação de grupos do BEE-C	47
Figura 15	Exemplo de uma rede de sensores clusterizada	48
Figura 16	Solução para o problema de clusterização (conjunto de nós cluster-heads) ..	49

Figura 17	Representação de um gene	49
Figura 18	Algoritmo para população inicial e escolha da abelha rainha	53
Figura 19	Algoritmo de Seleção por Têmpera Simulada	54
Figura 20	Algoritmo de Seleção por Roleta	55
Figura 21	Cruzamento entre zangão e rainha por máscara binária	56
Figura 22	Algoritmo de cruzamento por máscara binária	56
Figura 23	Exemplo de mutação ou busca local em um cromossomo	56
Figura 24	Algoritmo de mutacao	57
Figura 25	Quantidade média de dados enviados à estação base em cada cenário	63
Figura 26	Quantidade de dados enviados à estação base no decorrer do tempo	65
Figura 27	Tempo médio de vida da rede nos diferentes cenários	66
Figura 28	Quantidade de nós vivos no decorrer do tempo	67
Figura 29	Tempo médio de cobertura total da rede	68
Figura 30	Quantidade de energia consumida pela rede no decorrer do tempo	70
Figura 31	Quantidade de pacotes enviados por energia consumida	71

LISTA DE TABELAS

Tabela 1	Características dos protocolos	43
Tabela 2	Cenários utilizados nas simulações	58
Tabela 3	Densidade da rede em cada cenário	58
Tabela 4	Características e parâmetros do rádio	60
Tabela 5	Quantidade de clusters possíveis para cada cenário	61
Tabela 6	Informações estatísticas para os dados enviados para a estação base	64
Tabela 7	Informações estatísticas para o tempo de vida total da rede	67
Tabela 8	Informações estatísticas para o tempo de cobertura total da rede	69

LISTA DE SIGLAS

ACQUIRE	Active QUery forwarding In sensoR nEtworks
APTEEN	Adaptive Periodic Threshold-sensitive Energy Efficient sensor Network protocol
BCDCP	Base Station Controlled Dynamic Protocol
CDMA	Code Division Multiple Access
GAF	Geographic Adaptive Fidelity
GBR	Gradient-Based Routing
GEAR	Geographic and Energy Aware Routing
GeoMote	Geographic Multicast for Networked Sensors
GOAFR	Greedy Other Adaptive Face Routing
HBMO	Honey Bee Mating Algorithm
HPAR	Hierarchical Power-aware Routing
ICA	Inter Cluster Routing Algorithm
MCFA	Minimum Cost Forwarding Algorithm
PEGASIS	Power-Efficient Gathering in Sensor Information Systems
PSO-C	Particle Swarm Optimization - Centralizado
RSSF	Redes Sensores Sem Fio
SOP	Self Organizing Protocol
SPIN	Sensor Protocols for Information via Negotiation
TBF	Trajectory based Forwarding
TDMA	Time Division Multiple Access
TEEN	Threshold-sensitive Energy Efficient sensor Network protocol

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Objetivos	16
1.1.1	Objetivo Geral	16
1.1.2	Objetivos Específicos	16
1.2	Contribuições	16
1.3	Organização dos Capítulos	16
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Redes de Sensores Sem Fio	18
2.1.1	O Nó Sensor	18
2.1.2	A Estação Base	18
2.2	Aspectos Específicos e Desafios em Redes de Sensores Sem Fio	20
2.3	Consumo de Energia em Redes de Sensores Sem Fio	21
2.3.1	Modelo de Dissipação de Energia	21
2.4	Roteamento em Redes de Sensores Sem Fio	22
2.4.1	Roteamento Plano	23
2.4.2	Roteamento Hierárquico	23
2.4.3	Roteamento Geográfico	24
2.5	Clusterização em Redes de Sensores Sem Fio	24
2.5.1	Análise Comparativa do Consumo de Energia: Comunicação Direta, Multi-hop e por Clusterização	26
2.6	Síntese do Capítulo	28
3	TRABALHOS RELACIONADOS	30
3.1	Algoritmos bio-inspirados em abelhas para clusterização	30
3.1.1	HBMO	30
3.1.2	Bee Algorithm	34
3.2	Algoritmos para Clusterização em Redes de Sensores	36
3.2.1	LEACH	36
3.2.2	LEACH-C	38

3.2.3	ICA	40
3.2.4	BCDCP	41
3.2.5	PSO-C	42
3.3	Síntese do Capítulo	44
4	BEE-C: ALGORITMO BIO-INSPIRADO EM ABELHAS PARA CLUSTERIZAÇÃO EM REDES DE SENSORES	45
4.1	Pressuposições	45
4.2	Funcionamento do Protocolo BEE-C	45
4.2.1	Representação da solução do BEE-C	48
4.2.2	Função Objetivo	49
4.2.2.1	Primeira Função Objetivo	49
4.2.2.2	Segunda Função Objetivo	51
4.2.3	População Inicial e Escolha da Abelha Rainha	52
4.2.4	Seleção dos Zangões	52
4.2.4.1	Seleção por Têmpera Simulada	52
4.2.4.2	Seleção por Roleta	54
4.2.5	Acasalamento ou Cruzamento	55
4.2.6	Busca Local ou Mutação	55
4.2.7	Substituição da Abelha Rainha e Novas Gerações	57
4.3	Síntese do Capítulo	57
5	SIMULAÇÕES E RESULTADOS	58
5.1	Ambiente de Utilizado	58
5.2	Cenário	58
5.2.1	Parâmetros da rede para as simulações	59
5.2.1.1	Quantidade de clusters da rede	60
5.2.1.2	Tempo para cada rodada	61
5.2.2	Parâmetros para o algoritmo baseado no HBMO	62
5.3	Métricas para Avaliação	62
5.4	Resultados	62
5.4.1	Quantidade de dados enviados à estação base	63
5.4.2	Tempo de vida total da rede	65

5.4.3	Tempo de cobertura total da rede	68
5.4.4	Consumo de energia	70
5.5	Síntese do Capítulo	71
6	CONCLUSÕES E TRABALHOS FUTUROS	72
	REFERÊNCIAS BIBLIOGRÁFICAS	73

1 INTRODUÇÃO

O resultado de avanços em tecnologia e comunicação sem fio, e redes de sensores, tem emergido como uma nova ferramenta importante e indispensável para detecção de contaminações em ambientes perigosos, monitoramento de habitat nas reservas naturais, inspeções de inimigos em ambientes de guerra, entre outras aplicações (JIANG; MANIVANNAN, 2004; MURUGANATHAN et al., 2005).

As Redes de Sensores Sem Fio (RSSF) são um tipo especial de redes ad hoc que possibilitam o monitoramento do mundo físico através de redes de pequenos sensores densamente ou esparsamente distribuídos (AKYILDIZ et al., 2002). Essas redes são formadas por centenas ou milhares de nós sensores multifuncionais com baixa carga de energia, operando de forma autônoma em um ambiente, com capacidades computacionais limitadas e uma estação base, responsável por receber os dados dos nós sensores.

Atualmente as RSSFs são alvos de muitos desafios. Um dos quais está relacionado com a escassez de energia disponível nos sensores, sendo que uma grande parte das pesquisas feitas atualmente procuram enfatizar maneiras eficientes de economizar energia dos sensores, fazendo com que a vida dessas redes seja prolongada. Outras pesquisas estão relacionadas com problemas de cobertura e roteamento.

Uma técnica muito utilizada para tornar eficiente o uso de energia na rede e, conseqüentemente, aumentar seu tempo de vida, é a técnica de clusterização. Uma rede clusterizada é uma rede dividida em grupos de nós, sendo que cada grupo tem um líder, responsável por receber os dados do grupo, fazer a agregação e enviá-los à estação base. Várias são as técnicas empregadas para clusterização, entre as quais destaca-se o algoritmo k-means (JAIN; DUBES, 1988), que utiliza centroides para formar os clusters. Centroides são os centros de cada cluster, o algoritmo k-means busca minimizar a soma dos quadrados das distâncias entre os pontos e os centroides.

Apesar do uso de clusterização em redes de sensores apresentar muitos benefícios, o problema de clusterização é NP-Difícil (AGARWAL; PROCOPIUC, 2002), ou seja, nenhum algoritmo determinístico conhecido é capaz de resolvê-lo em tempo polinomial. Assim, o uso de técnicas que encontrem boas soluções em pouco tempo é primordial, principalmente em redes de sensores, pois é importante não haver grandes atrasos nas transmissões das mensagens. Muitos são os algoritmos utilizados para clusterização (BERKHIN, 2006), no entanto, algoritmos bio-inspirados têm conseguido melhores resultados para clusterização de dados (SH-ELOKAR; JAYARAMAN; KULKARNI, 2004; KAO et al., 2008; FATHIAN; AMIRI; MAROOSI, 2007; PHAM et al., 2007a).

Algoritmos bio-inspirados são abordagens que procuram solucionar vários tipos de

problemas através da modelagem matemática de comportamentos inteligentes naturais. Muitos são os comportamentos biológicos abstraídos, como por exemplo, colônia de formigas, enxame de peixes, colméia de abelhas, vespas, cupins, bactérias, bando de pássaros, etc.

Assim, o uso de algoritmos bio-inspirados para clusterização em RSSF pode ser uma boa opção, sendo que estes têm sido muito utilizados em clusterização de dados, tendo bons resultados.

1.1 Objetivos

1.1.1 Objetivo Geral

Neste trabalho é proposto um algoritmo de roteamento, que visa a formação eficiente dos clusters na rede, através de um algoritmo baseado em colônia de abelhas, que busca minimizar o consumo de energia da rede, e como consequência, aumentar o tempo de vida dos nós. O algoritmo é específico para redes de disseminação contínua de dados e é baseado no protocolo LEACH-C.

1.1.2 Objetivos Específicos

- Aumentar o tempo de vida da rede, caso possível;
- Aumentar a quantidade de dados enviados dos nós para a estação base;
- Aumentar o tempo de cobertura total da rede (contabilizado até a morte do primeiro nó);
- Otimizar o consumo de energia da rede;

1.2 Contribuições

A elaboração deste trabalho traz as seguintes contribuições:

- Utilização de um algoritmo inspirado no comportamento de abelhas para clusterizar uma RSSF;
- Adaptação do algoritmo ao problema de clusterização em RSSF;
- Utilização de novas funções objetivo que buscam encontrar melhores clusters para a rede.

1.3 Organização dos Capítulos

Esta dissertação está organizada em sete capítulos. O capítulo 2 apresenta uma visão geral do roteamento e clusterização em RSSF, onde são descritos os componentes básicos de uma rede de sensores, são apresentadas suas peculiaridades. Logo após, é feito um pequeno estudo do consumo de energia em uma rede de sensores. Por fim, concluímos o capítulo descrevendo o uso e benefícios da clusterização em uma RSSF.

No capítulo 3 é feito um levantamento de alguns trabalhos relacionados com o proposto. Inicialmente neste capítulo são apresentados dois algoritmos bio-inspirados em abelhas: o HBMO (Honey Bee Mating Algorithm) e o Bee Algorithm. Em seguida, o capítulo apresenta alguns algoritmos de roteamento baseados clusterização para redes de sensores. Dentre estes, o LEACH-C teve uma grande importância para a formulação deste trabalho.

O algoritmo BEE-C é detalhado no capítulo 4, onde são mostrados as pressuposições para a utilização do algoritmo, além do detalhamento total de suas fases.

O capítulo 5 mostra as simulações, resultados e análises. Neste capítulo são definidos os cenários e parâmetros para as simulações.

Por fim, o capítulo 6 mostra as conclusões e dar sugestões de possíveis trabalhos futuros para este trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta um estudo detalhado sobre roteamento e clusterização em redes de sensores. Na seção 2.1 são descritos os componentes de uma RSSF. Na seção 2.2 são apresentados os aspectos específicos das redes de sensores, onde são descritas as características peculiares de uma RSSF. Em seguida, na seção 2.3, é feito um estudo sobre o consumo de energia em RSSF, no qual é apresentado um modelo de consumo de energia e alguns aspectos que devem ser levados em conta para projeção de um protocolo eficiente em energia. Logo após, na seção 2.4, é descrito o roteamento em RSSF, abordando a classificação e exemplos de protocolos. Na seção 2.5 é feito um estudo sobre clusterização em RSSF, caracterizando esta técnica e apresentando exemplos de algoritmos de roteamento que a utilizam.

2.1 Redes de Sensores Sem Fio

Uma RSSF é basicamente composta por vários nós sensores e, normalmente, uma estação base. Os nós sensores coletam as informações e as enviam para a estação base, que pode se comunicar com outras redes, para que os dados de interesse sejam analisados (LABRADOR; WIGHTMAN, 2009).

2.1.1 O Nó Sensor

Um nó sensor é composto por cinco principais componentes: bateria, memória, processador, transceptor e dispositivo de sensoriamento (KARL; WILLIG, 2007). A bateria armazena a energia do sensor, que além de ter capacidade limitada, há pouca possibilidade de reposição. A capacidade da memória e do processador são pequenas devido ao tamanho e preço do sensor, que deve ser barato. O rádio inclui sistema de transmissão, recepção, amplificador e antena. A Figura 1 ilustra a arquitetura de um nó sensor.

2.1.2 A Estação Base

Uma estação base é uma entidade para onde os nós sensores enviam as informações, ela pode ser um computador pessoal, um servidor ou um gateway, que fornece conexão física com a internet (KARL; WILLIG, 2007). A estação base não possui as restrições de um nó sensor, tendo grande poder de processamento, alta capacidade de armazenamento e sem restrições de bateria. A Figura 2 mostra um exemplo de RSSF na qual a estação base envia os dados para o usuário através da internet.

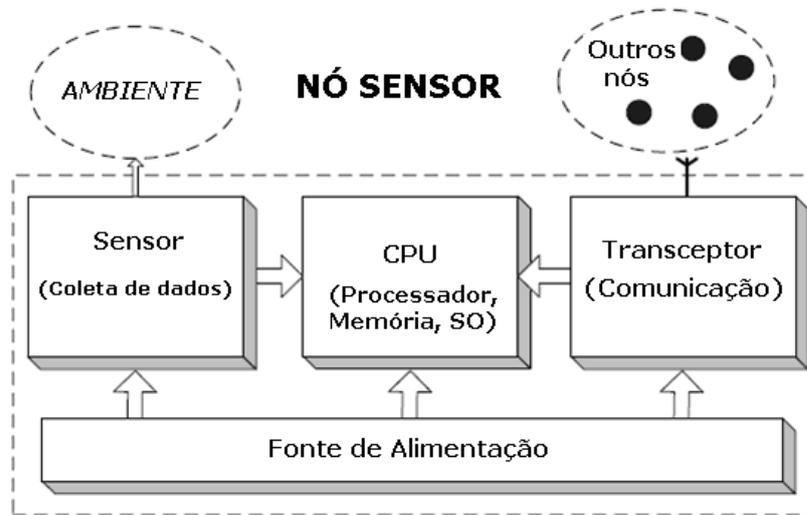


Figura 1: Arquitetura do nó sensor

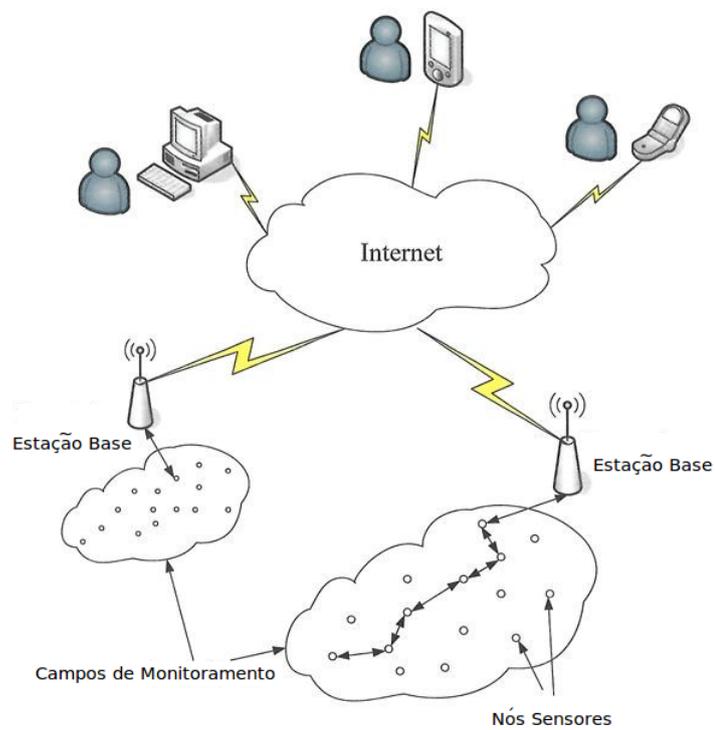


Figura 2: Um exemplo de redes de sensores sem fio

2.2 Aspectos Específicos e Desafios em Redes de Sensores Sem Fio

As RSSFs diferem das redes comuns, tendo algumas peculiaridades. Como foi mencionado anteriormente, cada nó sensor tem capacidades de processamento e armazenamento limitadas, por isso devem ter baixo custo a fim de serem adquiridos em quantidades elevadas. Ao projetar uma RSSF também devem ser consideradas as condições do ambiente no qual venha atuar, tais condições podem ser muito rigorosas, como tempestades, nevascas, queimadas, etc. Com isso, os nós podem falhar, além de possíveis falhas de comunicação.

Considerando que a configuração manual de uma grande rede de pequenos dispositivos seja impraticável, a rede deve ser auto-organizável (CULLER; ESTRIN; SRIVASTAVA, 2004).

Os protocolos de roteamento das redes comuns e das demais redes ad hoc não se aplicam ao contexto de RSSF, pois não consideram suas limitações.

Algumas restrições devem ser consideradas na projeção de protocolos para RSSF, sendo que tais restrições fazem das redes de sensores diferentes de outras redes ad hoc sem fio (AL-KARAKI; KAMAL, 2004):

- Recursos Limitados: Quantidade de energia, processamento e armazenamento;
- Topologia dinâmica: Podem existir falhas, e para garantir o bom funcionamento da rede, deve haver redundâncias. Os nós sensores podem ser programados para ficarem inativos quando o período é de baixa atividade, com a finalidade de economizar energia;
- Tempo de vida da rede: Para minimizar o custo de manutenção da rede, deve-se maximizar o tempo de vida da rede.

Os requisitos de um protocolo de roteamento para RSSF diferem de acordo com a característica de cada tipo de RSSF.

Os protocolos de roteamento para RSSF devem considerar as limitações da rede, minimizar os problemas existentes e otimizar o roteamento. A seguir são listados alguns desafios de pesquisa em RSSF (AL-KARAKI; KAMAL, 2004):

- Minimizar o consumo de energia;
- Maximizar o tempo de vida útil da rede;
- Garantir tolerância a falhas;
- Ser eficiente na comunicação;
- Garantir escalabilidade;

- Fazer uso de um bom método de agregação de dados.

O consumo de energia, principal desafio em RSSF, é abordado com mais detalhes na seção 2.3.

2.3 Consumo de Energia em Redes de Sensores Sem Fio

Como normalmente as redes de sensores são implantadas em locais de difícil acesso, como florestas, oceanos, etc., fica indesejável recarregar ou substituir as baterias dos nós sensores. Assim, a conservação da energia torna-se crucial para tornar o tempo de vida da rede longo. O desafio torna-se ainda maior, considerando que um nó sensor tem bateria com pouca energia.

O consumo de energia dos nós sensores pode ser indicado pela energia necessária para a transmissão e recepção das mensagens. A análise clássica dos algoritmos centra-se na comunicação, ou a contagem do número de rodadas de comunicação ou o número de mensagens enviadas. Isto pode ser estendido de forma direta a uma medida de complexidade de energia através da contagem da energia necessária para a transmissão das mensagens enviadas durante o tempo de vida da rede. No entanto, medições tem mostrado que a maior parte da energia da rede é gasta pelos nós quando escutam os sinais. Para lidar com este problema, técnicas de economia de energia ajustam os nós para o modo inativo, de acordo com um agendamento, a fim de economizar energia (WAGNER; WATTENHOFER, 2007).

As técnicas de economia de energia confiam no fato de que o processador dos nós tem a capacidade de ser fixado em um estado de sono, com um montante mínimo de energia. Geralmente, o dispositivo de comunicação é completamente desligado durante o estado de sono (WAGNER; WATTENHOFER, 2007).

Normalmente, a economia de energia é um equilíbrio entre a eficiência da rede - medida por exemplo pela taxa de envio de dados, conectividade ou latência - e a quantidade de energia consumida pela rede. Portanto, para aumentar o tempo de vida da rede sem perder a eficiência, protocolos eficientes em energia são desenvolvidos (WAGNER; WATTENHOFER, 2007).

2.3.1 Modelo de Dissipação de Energia

Segundo (HEINZELMAN et al., 2002), a dissipação de energia em um nó sensor é realizada de acordo com o modelo mostrado nas equações 2.1 e 2.2.

O consumo de energia para transmitir uma mensagem de k bits a uma distância d é:

$$E_{Tx}(k, d) = E_{Tx-elec}(k) + E_{Tx-amp}(k, d) = \begin{cases} E_{elec} * k + \epsilon_{fs} * k * d^2, & d < d_0 \\ E_{elec} * k + \epsilon_{mp} * k * d^4, & d \geq d_0 \end{cases} \quad (2.1)$$

e para receber a mensagem, o rádio gasta:

$$E_{Rx}(k) = E_{Rx-elec}(k) = E_{elec} * k, \quad (2.2)$$

onde:

$E_{Tx-elec}$ = Energia gasta na transmissão;

$E_{Rx-elec}$ = Energia gasta na recepção da mensagem;

E_{Tx-amp} = Energia do amplificador da transmissão;

d_0 = distância limite, calculada de acordo com os valores de E_{elec} , ϵ_{fs} e ϵ_{mp} ;

ϵ_{fs} = Parâmetro chamado de modelo de espaço livre (fs), é utilizado se a distância da origem ao destino for menor que d_0 ;

ϵ_{mp} = Parâmetro chamado de modelo multipath (mp), utilizado se a distância da origem ao destino for maior ou igual a d_0 ;

E_{elec} = Energia gasta por bit transmitido ou recebido.

Há dois elementos importantes na dissipação de energia em uma RSSF: o tamanho da mensagem e a distância entre o nó origem e o nó destino. Portanto, para melhorar o consumo de energia com o intuito de aumentar o tempo de vida da rede, é essencial prover um método que utilize distâncias mínimas entre a origem e o destino, e que use agregação e fusão de dados, para reduzir o tamanho das mensagens sem prejudicar a qualidade da informação.

2.4 Roteamento em Redes de Sensores Sem Fio

A classificação de roteamento em RSSF mais vista na literatura, divide o roteamento em RSSF, de acordo com a estrutura da rede, em roteamento plano, roteamento hierárquico ou roteamento geográfico (GARCÍA-HERNANDO et al., 2008; AL-KARAKI; KAMAL, 2004; AKKAYA; YOUNIS, 2005). Geralmente, no roteamento plano, os nós têm as mesmas funcionalidades. Já no roteamento hierárquico, os nós podem assumir diferentes papéis na rede.

No roteamento geográfico a posição geográfica dos nós é usada para rotear os dados adequadamente.

Além disso, os protocolos podem ser classificados em roteamento por múltiplos caminhos, consulta, negociação, qualidade de serviço, entre outros, dependendo do funcionamento do protocolo (GARCÍA-HERNANDO et al., 2008; AL-KARAKI; KAMAL, 2004; AKKAYA; YOUNIS, 2005).

Por último, o roteamento em RSSF pode ser classificado, de acordo com o método usado para adquirir a rota, em: proativo, reativo e híbrido (GARCÍA-HERNANDO et al., 2008; AL-KARAKI; KAMAL, 2004; AKKAYA; YOUNIS, 2005). No roteamento proativo, os protocolos adquirem as rotas antes de precisar delas, já nos protocolos reativos, as rotas são adquiridas sob demanda. Os protocolos híbridos usam a combinação das características proativas e reativas.

2.4.1 Roteamento Plano

Esse tipo de roteamento se aplica principalmente a redes planas, normalmente homogêneas. Nesse roteamento todos os nós realizam as mesmas funções. Cada nó é responsável por coletar dados e transmiti-los à estação base.

Dois dos principais protocolos de roteamento plano são o SPINSPIN (Sensor Protocols for Information via Negotiation) (HEINZELMAN; KULIK; BALAKRISHNAN, 1999) e o Difusão Direcionada (Directed Diffusion) (INTANAGONWIWAT; GOVINDAN; ESTRIN, 2000).

O SPIN faz o roteamento com base na informação sobre a quantidade de energia disponível em cada nó. Ele dissemina as informações de um nó entre os demais através de protocolos de negociação. Quando um nó está com a quantidade de energia próxima a um limite crítico predefinido, ele passa a participar menos da disseminação de dados.

O Difusão Direcionada procura estabelecer canais de comunicação eficientes entre os nós sensores e a estação base. Os nós enviam informação quando recebem consulta sobre dados dos quais têm conhecimento. Em nós intermediários, os dados podem ser agregados em um simples pacote, diminuindo o número de transmissões e o volume de dados transmitidos.

Outros exemplos de protocolos planos são: Rumor Routing (BRAGINSKY; ESTRIN, 2002), MCFA - Minimum Cost Forwarding Algorithm (YE et al., 2001), GBR - Gradient-Based Routing (SCHURGERS; SRIVASTAVA, 2001), COUGAR (YAO; GEHRKE, 2002), ACQUIRE - Active QUery forwarding In sensoR nEtworks (SADAGOPAN; KRISHNAMACHARI; HELMY, 2003), entre outros.

2.4.2 Roteamento Hierárquico

No roteamento hierárquico ou baseado em clusters, os nós podem se dividir em grupos (clusters). Existem duas classes de nós sensores, que são os nós fontes e os nós líderes de cada grupo, chamado de cluster-head, que coordena as atividades intra-cluster. Os nós fontes coletam as informações e enviam para o cluster-head, este realiza a fusão dos dados, diminuindo a redundância e retirando mensagens inúteis, e depois enviam para a estação base. As redes podem ser homogêneas ou heterogêneas. O roteamento hierárquico é mais complexo, mas segundo alguns autores possibilita maior escalabilidade (MANJESHWAR; AGRAWAL, 2001), além de tornar possível a fusão de dados.

Dentre os algoritmos hierárquicos destaca-se o LEACH (HEINZELMAN et al., 2000). Variações do LEACH têm sido propostas, como o ICA (MAIA; CÂMARA; LOUREIRO, 2004), em que a comunicação entre cluster-head e Estação-base passa a ser multihop, e o LEACH-C (HEINZELMAN et al., 2002), que realiza o processo de formação de clusters centralizado na estação base. Estes algoritmos serão vistos com mais detalhes na seção 4, que fala sobre protocolos que utilizam clusterização.

Outros exemplos de protocolos hierárquicos são: PEGASIS - Power-Efficient Gathering in Sensor Information Systems (LINDSEY; RAGHAVENDRA, 2002), TEEN - Threshold-sensitive Energy Efficient sensor Network protocol (MANJESHWAR; AGRAWAL, 2001), APTEEN - Adaptive Periodic TEEN (MANJESHWAR; AGRAWAL, 2002), SOP - Self Organizing Protocol (SUBRAMANIAN; KATZ, 2000), HPAR - Hierarchical Power-aware Routing (LI; ASLAM; RUS, 2001), entre outros.

2.4.3 Roteamento Geográfico

Baseia-se na localização dos nós sensores. Cada nó sensor deve possuir conhecimento sobre a localização de um grupo de sensores. Com esta informação, um nó origem pode determinar o melhor caminho até um nó destino.

Um exemplo deste tipo de roteamento é o Trajectory based Forwarding – TBF (NICULESCU; NATH, 2003), em que a rota de um pacote é escrita como uma função interpolada, ou seja, em vez de o pacote carregar os endereços dos nós pelos quais ele deve passar, ele traz uma função $f(x)$, e de acordo com as suas coordenadas (x,y) o nó sabe se ele deve encaminhar o pacote ou não.

Outro exemplo de protocolo baseado em roteamento geográfico é o GeoMote (Geographic Multicast for Networked Sensors) (BROADWELL; POLASTRE; RUBIN, 2001). Neste protocolo os destinatários das mensagens são endereçados por meio de polígonos, permitindo comunicações multicast localizadas. No GeoMote, existem três categorias de nós: os nós Geo-

Hosts, responsáveis por produzir dados, os GeoRouters, responsáveis por repassar os dados produzidos, e os GeoGateways, que atuam como ponto de entrada e saída dos nós. O caminho da propagação dos pacotes é definido através de um algoritmo guloso.

Outros protocolos geográficos são: GAF - Geographic Adaptive Fidelity (XU; HEIDEMANN; ESTRIN, 2001), GEAR - Geographic and Energy Aware Routing (YU; GOVINDAN; ESTRIN, 2001), GOAFR - Greedy Other Adaptive Face Routing (KUHN; WATTENHOFER; ZOLLINGER, 2003), entre outros.

2.5 Clusterização em Redes de Sensores Sem Fio

Embora comumente utilizadas para gerenciamento de dados e reconhecimento de padrões, técnicas de clusterização são empregadas em todos os domínios da ciência da computação. Focalizando em sistemas de comunicação, os métodos de clusterização podem ser explorados para melhorar grandemente a escalabilidade de métodos de comunicação, especialmente em redes ad hoc e redes de sensores (DRESSLER, 2007).

Os melhores domínios conhecidos para o uso eficiente de técnicas de agrupamento em redes são (DRESSLER, 2007):

- **Utilização otimizada de recursos:** Técnicas de clusterização foram usadas com sucesso para economia de tempo e energia. Estas otimizações essencialmente refletem o uso de algoritmos de clusterização para alocação de tarefas e recursos.
- **Escalabilidade Melhorada:** a clusterização ajuda a organizar redes ad hoc não-estruturadas de larga escala em grupos bem definidos, de acordo com as necessidades específicas da aplicação, sendo que as tarefas e os recursos necessários podem ser distribuídos nesta rede de maneira otimizada.

A clusterização fornece técnicas e medidas para reduzir o tamanho da rede a ser controlada, através do agrupamento dos nós em clusters de tamanho gerenciável. Em geral, as técnicas de clusterização podem ser classificadas em duas dimensões: de acordo com a medida de similaridade utilizada e de acordo com o comportamento básico do algoritmo. A seguir está a classificação segundo a medida de similaridade (DRESSLER, 2007):

- **Clusterização com base na distância:** na clusterização com base na distância a similaridade entre os objetos é representada pela distância medida. Assim, um número de objetos pertence ao mesmo grupo se eles estão próximos em termos de distância entre qualquer par dos objetos.

- **Clusterização conceitual:** em contrapartida, clusterização conceitual se refere a características particulares que devem ser preenchidas por objetos do mesmo grupo. Assim, neste caso, a similaridade define um conceito comum para todos os objetos do mesmo cluster.

Em uma segunda dimensão, o processo de clusterização pode ser classificado de acordo com a organização (DRESSLER, 2007):

- **Clusterização centralizada:** baseada em informação de estado global, ou seja, no conhecimento sobre todos os objetos no campo e suas características.
- **Clusterização distribuída:** na clusterização distribuída, os clusters são normalmente formados dinamicamente. Primeiro um objeto líder é escolhido, por exemplo, baseado em algum algoritmo conhecido de eleição de sistemas distribuídos. Depois, os membros do grupo e os recursos são gerenciados pelo líder.

Em RSSF, os algoritmos de clusterização são utilizados com a finalidade de garantir uma melhor escalabilidade, aumentar o tempo de vida da rede e diminuir a redundância dos dados.

2.5.1 Análise Comparativa do Consumo de Energia: Comunicação Direta, Multi-hop e por Clusterização

Em (ZHAO; GUIBAS, 2004), os autores realizam uma comparação do consumo de energia entre uma rede que utiliza transmissão direta dos nós para a estação base e uma rede que realiza a transmissão através de múltiplos saltos (multi-hop). Para isso eles consideraram um exemplo de uma rede de N saltos, assumindo que a distância total da transmissão é Nr , onde r é a distância de um salto. A energia mínima recebida para uma dada taxa de erro de transmissão é $P_{receive}$, e a energia da transmissão de um nó é P_{send} . Então o modelo de atenuação rádio-frequência (RF) é dado por:

$$P_{receive} \propto \frac{P_{send}}{r^\alpha} \quad (2.3)$$

onde α é o expoente de atenuação RF. Devido algumas interferências, o valor de α pode variar entre 2 e 5. Equivalentemente temos:

$$P_{send} \propto r^\alpha P_{receive} \quad (2.4)$$

Portanto a vantagem da transmissão em múltiplos saltos em relação a transmissão direta, tendo a mesma distância Nr , é:

$$\eta_{rf} = \frac{P_{send(Nr)}}{N \cdot P_{send(r)}} = \frac{(Nr)^\alpha P_{receive}}{N \cdot r^\alpha P_{receive}} = N^{\alpha-1} \quad (2.5)$$

A Figura 3 seguinte mostra a atenuação da energia para redes multi-hops e de transmissão direta.

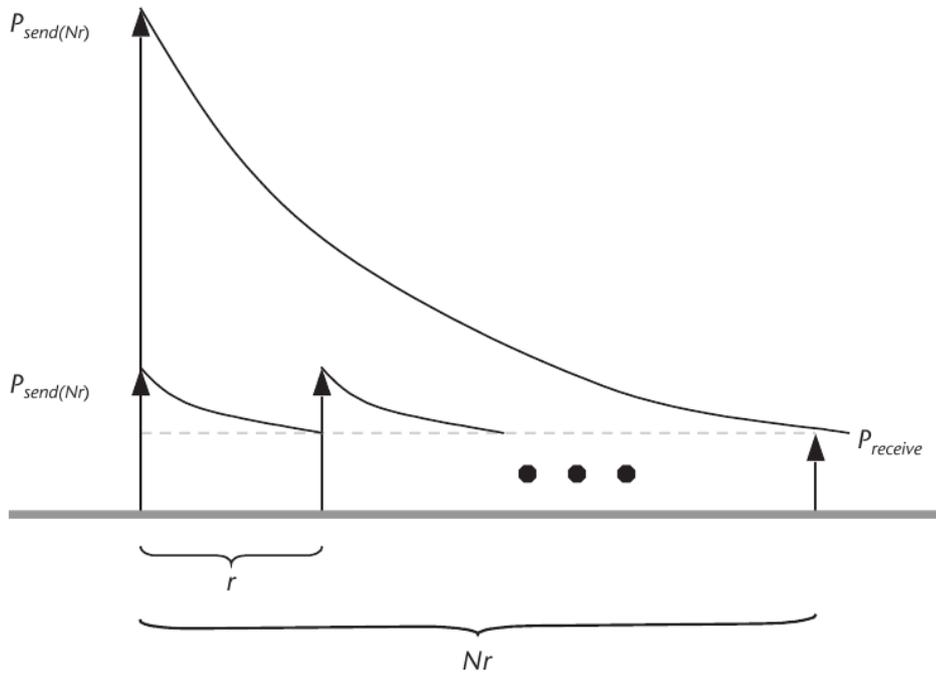


Figura 3: Vantagem do uso de comunicação multi-hop

Apesar de apresentar um ganho muito superior em transmissão multi-hop, esta análise desconsiderou o consumo de energia por outros componentes de um circuito rádio-frequencia. E também deve ser levado em consideração que apesar do gasto de energia em uma transmissão direta ser maior, a latência é menor, o que causa menos atraso nas transmissões.

Em (HEINZELMAN et al., 2000) são apresentados estudos que confirmam que a economia de energia em redes de transmissão multi-hop é maior do que em transmissão direta. As simulações realizadas mostram que na transmissão direta os nós morrem primeiro, principalmente os que estão mais distantes da estação base (Figura 4), e o tempo de vida da rede é menor do que na transmissão multi-hop.

No entanto, na comunicação multi-hop, apesar do ganho, os nós próximos à estação base morrem primeiro (Figura 5), fazendo com que áreas da rede não sejam cobertas.

Assim, mesmo havendo vantagem do uso de comunicação multihop em relação ao

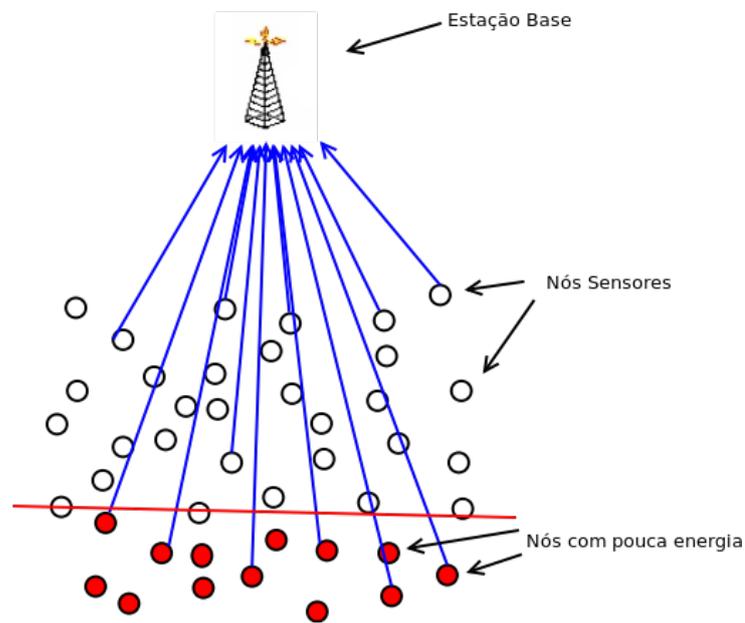


Figura 4: Comunicação direta dos nós para a estação base

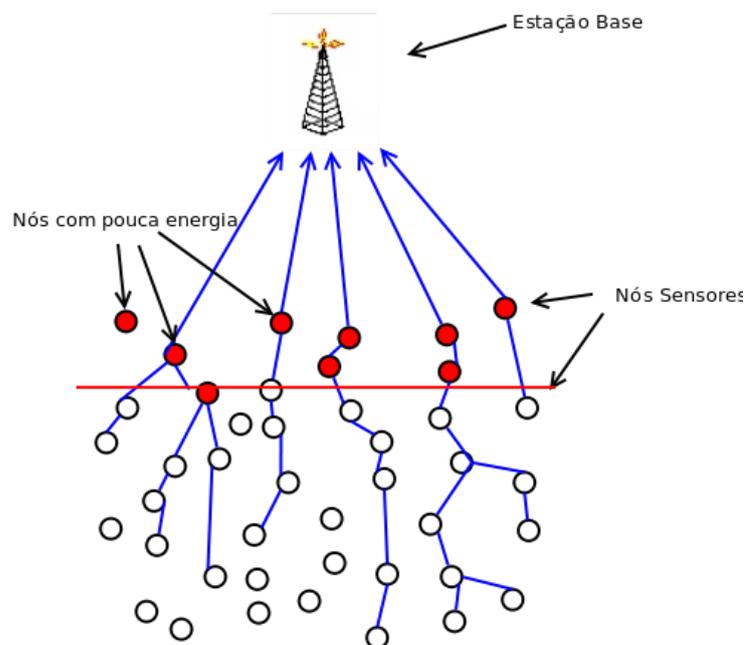


Figura 5: Comunicação multihop dos nós para a estação base

uso de comunicação direta em RSSF, o consumo de energia dos nós em ambas as abordagens é muito desproporcional. Além disso, estas técnicas não garantem escalabilidade, pois com o aumento dos nós, na comunicação direta haverá muita colisão na transmissão, e na comunicação multihop haverá muito atraso.

Uma boa técnica para um melhor consumo de energia em RSSF é a clusterização, onde os nós são organizados em grupos de nós sensores, os quais se comunicam com os nós líderes (cluster-heads), que transmitem os dados para a estação base. Isso reduz bastante a distância de envio dos nós, pois os cluster-heads normalmente ficam próximos dos nós do grupo (HEINZELMAN et al., 2000).

Dessa forma, a clusterização parece ser uma abordagem que realiza uma comunicação eficiente em energia, pois garante um consumo de energia mais distribuído. No entanto, o nó cluster-head deve estar bem localizado e ter alto nível de energia, pois será muito utilizado. Além disso, com o uso de clusters pode ser realizada a agregação dos dados, que diminui a redundância e os tamanhos das mensagens, fazendo com que haja mais economia de energia. A Figura 6 ilustra um exemplo de clusterização em RSSF.

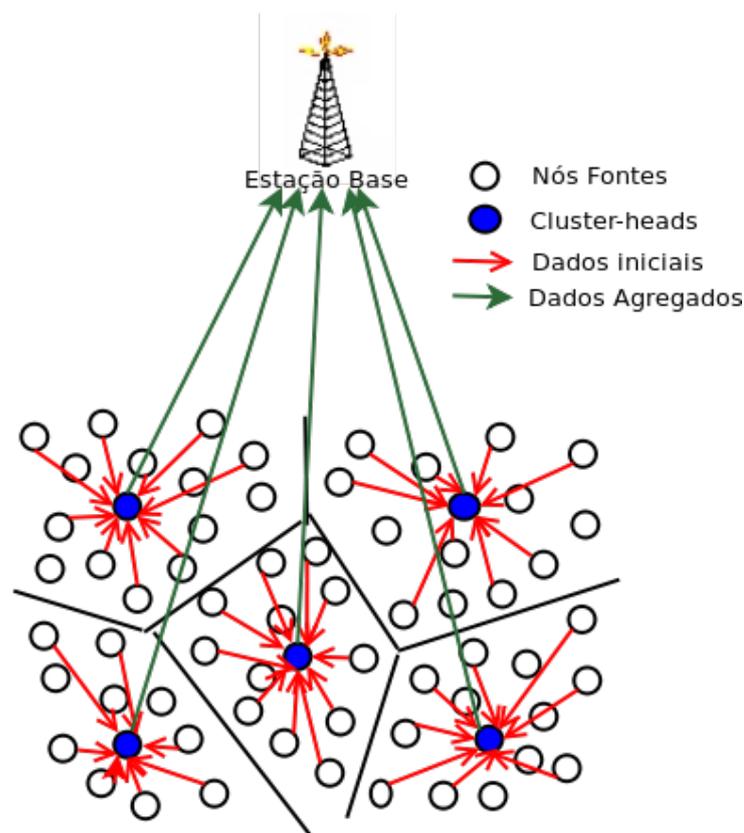


Figura 6: Rede clusterizada, somente os líderes enviam os dados para a estação base

2.6 Síntese do Capítulo

Neste capítulo foram apresentados os conceitos de redes de sensores, assim como as restrições e desafios apresentados por estas redes. Logo após, foi dedicada uma seção para um estudo sobre o consumo de energia em redes de sensores, pois se trata de um dos principais desafios para esse tipo de rede. Em seguida foi visto o roteamento em RSSF, apresentando as suas classificações e dando exemplos de protocolos. Por fim, a última seção se refere ao estudo de clusterização em redes de sensores sem fio.

3 TRABALHOS RELACIONADOS

Neste capítulo são apresentados os trabalhos relacionados, entre os quais, alguns com contribuição de grande valia para o trabalho proposto. Na seção 3.1 são descritos dois algoritmos inspirados no comportamento de abelhas utilizados para clusterização de dados. Na seção 3.2 são abordados alguns protocolos de roteamento para redes de sensores baseados em clusterização.

3.1 Algoritmos bio-inspirados em abelhas para clusterização

Nesta seção são apresentados dois algoritmos para clusterização baseados em comportamentos distintos de abelhas. O primeiro se baseia no comportamento de reprodução das abelhas e o segundo baseia-se no comportamento de busca de alimento.

3.1.1 HBMO

O algoritmo HBMO (Honey Bee Mating Optimization) (FATHIAN; AMIRI; MAROOSI, 2007) baseia-se no comportamento de reprodução das abelhas.

Em uma colmeia existem três tipos de abelhas: rainha, zangão e operária. Cada uma destas tem funções distintas. A abelha rainha é responsável apenas pela reprodução de novas abelhas, o zangão tem a única função de acasalar-se com a rainha, já a operária é responsável pela limpeza da colmeia, busca de alimentos, alimentação da rainha e do zangão, entre outras tarefas.

O algoritmo HBMO pode ser descrito através de cinco principais estágios:

1. O algoritmo inicia com o voo nupcial, onde a abelha rainha (melhor solução) seleciona zangões probabilisticamente para formar a espermateca (lista de zangões). Um zangão é então selecionado aleatoriamente para gerar os descendentes.
2. Criação de novos descendentes pelo cruzamento dos genótipos dos zangão com os da rainha.
3. Uso de operárias (heurísticas) para realizar uma busca local nos filhos gerados (soluções testes).
4. Adaptação da aptidão das operárias com base no valor de progresso dos filhos.
5. Substituição da rainha pelo melhor descendente.

As etapas descritas a seguir são mostradas na figura 8 e vistas com mais detalhes na figura 9.

Etapa 1: Representação da solução

Um cromossomo pode ser usado para representar uma solução para o problema, onde cada gene no cromossomo representa um parâmetro da solução. No problema de clusteração, o cromossomo é representado por um conjunto k de centros dos clusters, e cada gene são as dimensões dos centros. Especificadamente, um cromossomo pode ser representado como $C = [c_1 \dots c_j \dots c_k]$, onde c_j é o j ésimo gene e k é o total de número de genes. A figura 7 ilustra um exemplo de cromossomo. $C_1 = (2, 5, 1)$, $C_2 = (6, 3, 2)$, $C_3 = (5, 7, 4)$.

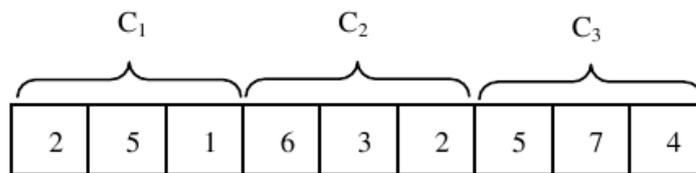


Figura 7: Exemplo de representação de uma solução

Etapa 2: Definição do modelo dos parâmetros de entrada

O algoritmo inicia com três parâmetros definidos pelo usuário e um pré-definido. O parâmetro pré-definido é o número de operárias (W), representando o número de heurísticas codificadas no programa. No entanto, o parâmetro pré-definido pode ser usado como um parâmetro para alterar o número de heurísticas ativas, se necessário, ou seja, o usuário pode escolher a primeira heurística, onde W é igual ou inferior ao número total de heurísticas codificadas no programa. Os parâmetros definidos pelo usuário são o número de rainhas, tamanho espermateca da rainha que representa o número máximo de descendentes que será suportado por todas as rainhas. A velocidade de cada rainha no início de cada voo de acasalamento é iniciada aleatoriamente.

Etapa 3: Geração aleatória de um conjunto de soluções iniciais

Nesta etapa, um conjunto inicial de centros de clusters é gerado aleatoriamente dos pontos do conjunto de dados. Cada solução representa um número k de centros dos clusters.

Etapa 4: Seleção da abelha rainha

Depois de gerar o conjunto inicial de soluções, é feita a ordenação das soluções com base na função de desempenho, e a melhor solução é determinada como rainha.

Etapa 5: Voo nupcial

Usar têmpera simulada para selecionar um conjunto de soluções (zangões) do espaço de busca para uma possível troca de informações com a melhor solução (rainha).

Etapa 6: Processo de reprodução

Gerar novo conjunto de soluções através do emprego de operadores de cruzamento e funções heurísticas predefinidas entre as soluções atuais (rainhas) e as soluções testes (zangões) de acordo com seus valores de função de aptidão.

Etapa 7: Alimentação dos filhos e da rainha com a geleia real

Melhorar as soluções geradas (descendentes) através de funções heurísticas e operadores de mutação de acordo com suas funções de aptidão.

Etapa 8: Substituição da abelha rainha

Se a melhor solução entre os filhos for melhor que a rainha, a rainha é substituída por este filho.

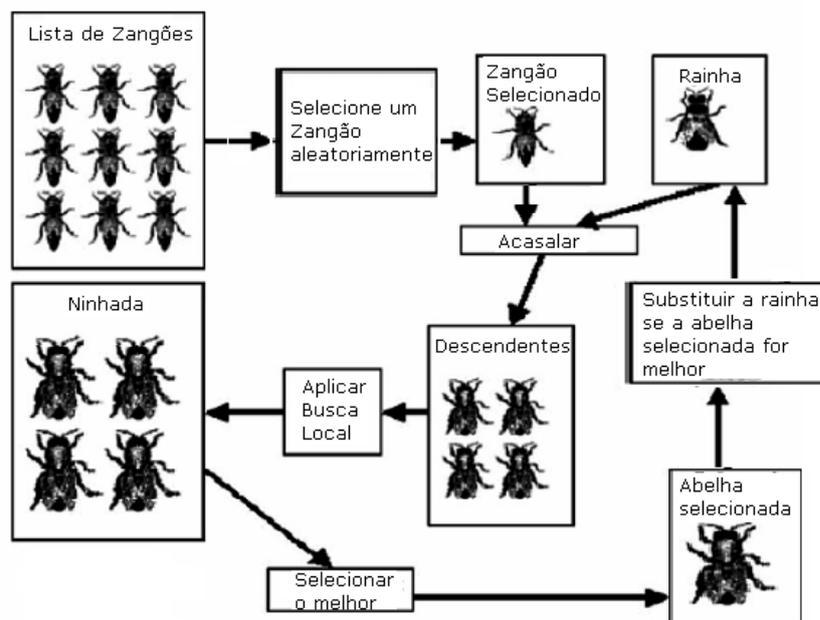


Figura 8: Algoritmo HBMO

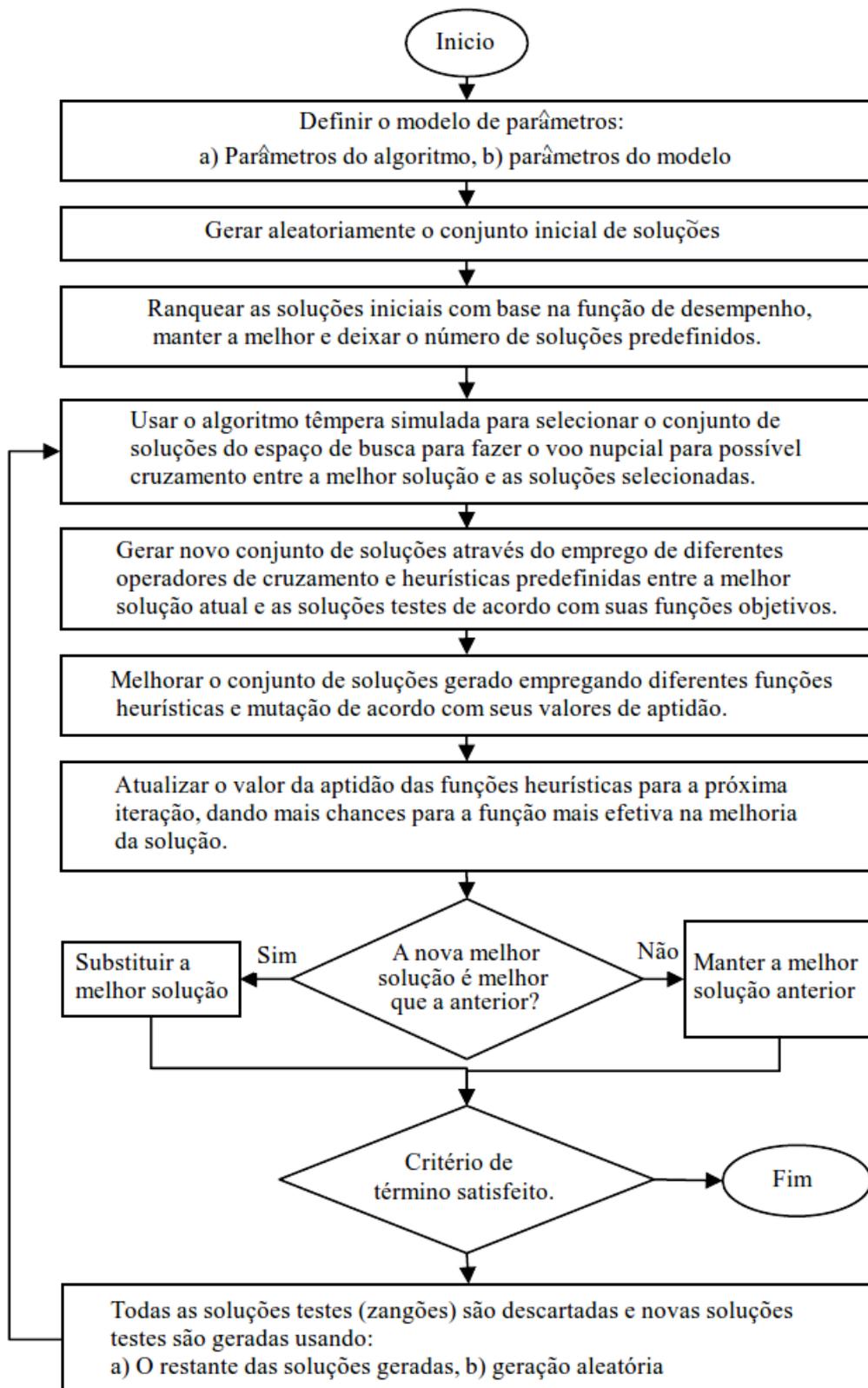


Figura 9: Representação do algoritmo HBMO

3.1.2 Bee Algorithm

O algoritmo Bee Algorithm (PHAM et al., 2007b) é baseado no comportamento de busca por fontes de alimentos feito pelas abelhas. O processo de busca de alimento em uma colônia começa quando as abelhas escoteiras são enviadas para a procura de trechos de flores promissores. Os trechos de flores com maior quantidade de néctar ou pólen, que podem ser coletados com menos esforço, tendem a ser mais visitados pelas abelhas, enquanto os trechos de flores com menos néctar ou pólen, e com mais dificuldade na coleta, tendem a ser menos visitados. Durante a época da colheita, uma colônia continua sua exploração, mantendo uma percentagem da população como abelhas escoteiras. Quando retornam à colmeia, as abelhas que encontrarem um trecho de flores com quantidade de néctar ou pólen acima de um certo nível de qualidade vão para a 'pista de dança' para executar uma dança.

Esta dança misteriosa é essencial para a comunicação da colônia, e contém três partes de informações sobre um trecho de flor: a direção em que será encontrado, sua distância para a colmeia e sua avaliação de qualidade (*fitness*). Essas informações ajudam a colônia a enviar suas abelhas ao trecho de flores precisamente. Depois de dançar, a dançarina volta para o trecho de flores com as abelhas seguidoras que estavam esperando dentro da colmeia. Os trechos de flores com mais qualidades recebem mais abelhas seguidoras. Isso permite que a colônia recolha alimentos de forma rápida e eficiente.

O algoritmo se baseia nesse comportamento das abelhas para resolver o problema de ótimo local do algoritmo *k - means*. Mais especificadamente, a tarefa é buscar por centros de clusters apropriados c_1, c_2, \dots, c_k , de forma que a métrica E (Equação 3.1) seja minimizada.

$$E = \sum_{j=1}^k \sum_{i=1}^{n_j} \left\| x_i^{(j)} - c_j \right\|^2 \quad (3.1)$$

onde $x_i^{(j)}$ é o i ésimo ponto pertencente ao j ésimo cluster, c_j é o centro do j ésimo cluster, k é o número de clusters e n_j é o número de pontos de dados no cluster j .

A seguir são mostradas as etapas básicas do algoritmo utilizado para clusterização:

1. Iniciar a população de n soluções
2. Avaliar a função de aptidão (*fitness*) da população
3. Enquanto (Critério de parada não é satisfeito)
4. Selecionar m locais para busca da vizinhança
5. Recrutar e abelhas para os locais selecionados (mais abelhas para os nep melhores locais) e avaliar suas *fitness*

6. Selecionar a melhor abelha de cada local
7. Atribuir as abelhas restantes para busca aleatória e avaliar suas fitness
8. Fim enquanto

O algoritmo requer um certo número de parâmetros a serem definidos, a saber: número de abelhas escoteiras (n), número de locais selecionados para a busca da vizinhança (de n locais visitados) (m), número dos melhores locais entre os m selecionados (e), número de abelhas recrutadas para os e melhores locais (nep), número de abelhas recrutadas para os outros ($m - e$) locais selecionados (nsp), e o critério de parada.

O algoritmo inicia com uma população inicial n de abelhas escoteiras. Cada abelha representa uma solução, ou seja, um conjunto k de centros dos clusters. As posições iniciais dos centros são atribuídas aleatoriamente.

As distâncias euclidianas entre cada objeto de dados e todos os centros são calculadas para determinar o cluster ao qual pertence o objeto de dados, ou seja, o cluster com o centro mais próximo do objeto. Desta forma, os clusters iniciais podem ser construídos.

Após a formação dos clusters, os centros dos clusters anteriores são substituídos pelos centróides atuais dos clusters para definir uma solução (ou seja, uma abelha).

Na etapa 2 do algoritmo, o processo de cálculo da função de aptidão (*fitness*) é realizada para cada local visitado por uma abelha a métrica de clusterização E (Equação 3.1) que é inversamente proporcional à *fitness*.

Na etapa 4, os m locais com melhores *fitness* são designados como os locais selecionados e escolhidos para uma busca pela vizinhança. Nas etapas 5 e 6, o algoritmo realiza buscas em torno dos locais selecionados, atribuindo mais abelhas para realizarem busca nas proximidades dos melhores locais e . A seleção dos melhores locais pode ser feita diretamente de acordo com as *fitness*. Alternativamente, os valores de fitness são utilizados para determinar a probabilidade dos locais serem selecionados. As buscas na vizinhança dos melhores e locais são mais detalhadas.

Na etapa 6, para cada local, somente a abelha com a melhor solução será selecionada para fazer parte da próxima população. Na etapa 7, as abelhas restantes da população são distribuídas aleatoriamente em torno do espaço de busca para explorar novas soluções.

No final de cada iteração, a colônia terá duas partes para sua nova população: os representantes dos locais selecionados e as abelhas escoteiras que realizam busca aleatória. Esses passos são repetidos até que um critério de parada seja satisfeito.

3.2 Algoritmos para Clusterização em Redes de Sensores

Esta seção descreve alguns algoritmos de roteamento baseado em clusterização para redes de sensores. Entre os quais, destacam-se o LEACH e o LEACH-C, pela grande contribuição que têm para com este trabalho. Os demais protocolos descritos nesta seção são baseados no LEACH e LEACH-C.

3.2.1 LEACH

O protocolo LEACH (Low-Energy Adaptive Clustering Hierarchy) (HEINZELMAN et al., 2000) é um protocolo hierárquico que visa diminuir o consumo de energia para aumentar o tempo de vida da rede.

No LEACH, os nós se organizam em clusters, com um nó agindo como líder (cluster-head). Todos os nós não-líderes devem transmitir seus dados ao cluster-head, enquanto o cluster-head deve receber dados de todos os membros do cluster, executar funções de processamentos nos dados (por exemplo, agregação de dados), e transmitir os dados para a estação base. Consequentemente, o consumo de energia de um nó cluster-head é mais intenso que o consumo de energia de um não-cluster-head.

O LEACH funciona por rodadas. A cada rodada os nós líderes são trocados com o objetivo de distribuir o consumo de energia da rede. Duas fases compõem as rodadas: fase de formação de clusters e fase de comunicação. Na fase de formação de clusters, é realizada a escolha dos líderes e a formação dos clusters e na fase de comunicação é feita a transferência dos dados até a estação base, incluindo agregação/fusão de dados pelos líderes.

O LEACH forma os clusters através de um algoritmo distribuído, onde os nós fazem decisões autônomas sem nenhum controle centralizado. As vantagens desta abordagem são que a comunicação de longa distância com a estação base é requerida e a formação de cluster distribuída pode ser feita sem conhecimento exato da localização dos nós na rede. Além disso, comunicação global é necessária para configurar os clusters, e nada é assumido sobre o estado corrente de algum outro nó durante a formação do cluster. Através do algoritmo distribuído cada nó decide se será ou não líder de acordo com uma probabilidade $P_i(t)$ no tempo t . Isso é feito levando em consideração a taxa k de clusters que a rede poderá ter, e que um nó pode se tornar líder uma vez em N/k rodadas. Os nós sensores selecionam um número aleatório r entre 0 e 1. Caso r seja menor que $P_i(t)$ (equação 3.2), então o nó torna-se cluster-head. A probabilidade $P_i(t)$ de um nó se tornar cluster-head é:

$$P_i(t) = \begin{cases} \frac{k}{N - k * (r \bmod \frac{N}{k})} & : C_i(t) = 1 \\ 0 & : C_i(t) = 0 \end{cases} \quad (3.2)$$

onde N = Número de nós da rede e k = Número de líderes da rodada. O valor de $C_i(t)$ é 1 (um) caso o nó não tenha sido selecionado nas últimas N/k rodadas e 0 (zero) caso contrário. Isso garante uma rotação entre os líderes, e portanto haverá uma boa distribuição no consumo de energia da rede.

Os cluster-heads escolhidos enviam a mensagem de anúncio, ADV-CH, em broadcast, informando que ele é líder. Os outros nós selecionam o cluster-head com sinal mais forte. Isso significa que os nós fontes estarão mais perto do cluster-head de seu grupo e, assim, gastarão menos energia para transmitir suas mensagens.

Os nós fontes enviam aos seus líderes uma solicitação para ingressar no grupo através da mensagem JOIN-REQ. O cluster-head armazena os nós que compõem o grupo. Os nós que não recebem a mensagem ADV-CH não fazem parte de nenhum grupo e enviam seus dados diretamente para a estação base.

Para uma organização melhor, os cluster-heads criam um escalonamento TDMA (Time Division Multiple Access) com os elementos do grupo, dando a cada nó um slot de tempo e enviam o escalonamento em broadcast na mensagem ADV-SCH, informando em qual período o nó pode transmitir os dados.

A figura 10 mostra um fluxograma do algoritmo de formação de cluster utilizado pelo LEACH.

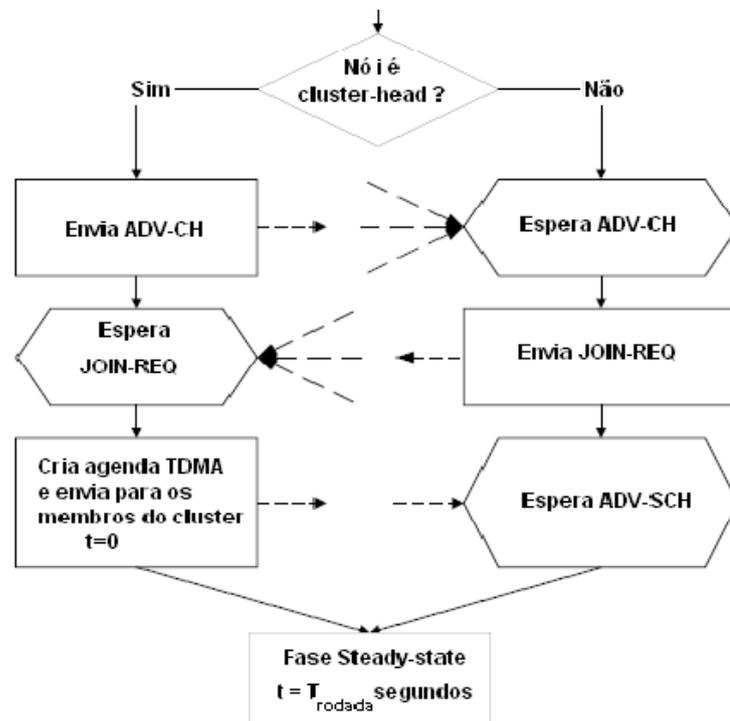


Figura 10: Formação dos grupos no protocolo LEACH

Depois da fase de formação de grupos, os nós fontes passam a monitorar e a enviar seus dados aos cluster-heads. Quando o cluster-head recebe as informações de todos os nós do grupo, ele faz a fusão/agregação de dados e encaminha para a estação base. As colisões entre as transmissões dos grupos são evitadas pelo uso de um código CDMA (Code Division Multiple Access). Esse código é escolhido aleatoriamente pelo líder e enviado aos nós fontes.

A figura 11 mostra o fluxograma da fase de comunicação do LEACH.

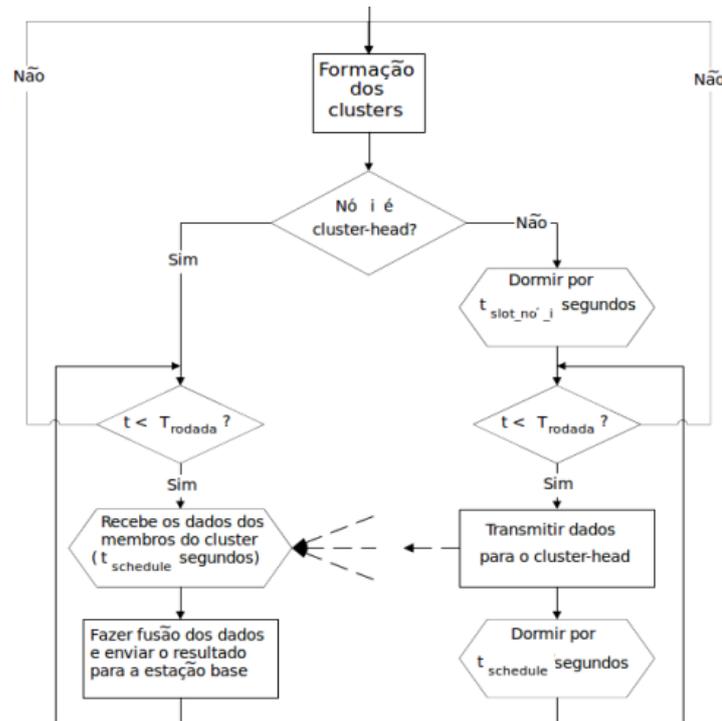


Figura 11: Fase de comunicação do protocolo LEACH

Esse processo é repetido ao fim de cada rodada, com alteração dos líderes e dos grupos.

3.2.2 LEACH-C

Anteriormente foi descrito o funcionamento do LEACH. Embora haja vantagem no uso do protocolo de formação distribuída de cluster LEACH, onde cada nó faz decisões autônomas e são colocados em um cluster, este protocolo não oferece nenhuma garantia de que seja escolhido o número necessário de cluster-heads, e de que eles estejam bem localizados em relação aos outros nós da rede. Uma vez que os clusters são adaptáveis, a obtenção de uma fraca criação de cluster durante uma rodada não irá afetar grandemente o desempenho global do LEACH. No entanto, o uso de um algoritmo de controle centralizado para formar clusters permite que melhor geração de clusters pela dispersão dos cluster-heads pela rede. Isto é base para o LEACH-C (LEACH-Centralizado) (HEINZELMAN et al., 2002), um protocolo que usa um algoritmo de clusterização centralizado e a mesma fase de comunicação do LEACH (os nós enviam seus

dados para o cluster-head, e o cluster-head agrega os dados e envia para a estação base). Um algoritmo formação centralizada produz clusters melhores ao selecionar o número desejado de cluster-heads e com boa distribuição na rede.

Durante a fase de formação de grupos do LEACH-C, cada nó envia informação sobre sua localização e nível de energia à estação base. A estação base executa um algoritmo de otimização para determinar os clusters para aquela rodada. Os clusters formados pela estação base em geral são melhores que aqueles formados por um algoritmo distribuído. Entretanto, o LEACH-C requer que cada nó transmita informação sobre sua localização para a estação base ao início de cada rodada. Esta informação pode ser obtida pelo uso de um sistema de posicionamento global (GPS) que é ativado no início de cada rodada para retornar a localização corrente dos nós [MENG 1998].

Com o objetivo de garantir a distribuição de energia entre todos os nós da rede, a estação base calcula a média de energia dos nós em cada rodada. Só podem ser cluster-head os nós que estão com o nível de energia acima da média, e com base neles a estação base executará o algoritmo Têmpera Simulada para determinar os melhores k cluster-heads. O algoritmo minimiza a quantidade de energia que os nós não-cluster-heads terão que usar para transmitir seus dados para o cluster-head, minimizando soma do quadrado das distâncias de todos os nós não-cluster-heads ao cluster-head mais próximo. A cada iteração, o próximo estado, que consiste de um conjunto de nós C' , é determinado pelo estado atual, o conjunto de nós C , pela perturbação aleatória das coordenadas x e y dos nós c em C , para gerar novas coordenadas x' e y' . Os nós que têm a localização mais próxima a (x', y') tornam-se o novo conjunto de cluster-heads c' quem formam o conjunto C' . Dado o estado atual na iteração k , representado pelo conjunto de cluster-heads C com função de custo $f(C)$, o novo estado, representado pelo conjunto de cluster-heads C' com custo $f(C')$ se tornará o estado corrente com a probabilidade:

$$P_k = \begin{cases} e^{-(f(C')-f(C))/\alpha_k} & : f(C') \geq f(C) \\ 1 & : f(C') < f(C) \end{cases} \quad (3.3)$$

onde α_k é o parâmetro de controle. A função de custo é representada por:

$$f(C) = \sum_{i=1}^N \min_{c \in C} d^2(i, c) \quad (3.4)$$

onde $d(i, c)$ é a distância entre o nó i e o nó c .

Após encontrar os clusters e os líderes correspondentes, a estação base transmite essa informação aos nós da rede. Os nós determinam o slot TDMA e esperam o tempo correto para transmitir.

3.2.3 ICA

O ICA (Inter Cluster Routing Algorithm) (MAIA; CÂMARA; LOUREIRO, 2004) é baseado no LEACH. No ICA, quando o processo inicia, a estação base envia um broadcast para todos os nós informando sua posição geográfica. Após isto os nós sabem a posição geográfica da estação base e as suas próprias posições. Para a formação dos clusters as regras são semelhantes às do LEACH, com exceção da decisão de qual cluster os nós vão participar, que é dada pela proximidade do nó ao cluster-head. Outra diferença é que ao se eleger líder, o nó envia sua posição geográfica a todos os vizinhos. Dessa forma, os líderes sabem onde estão os outros líderes mais próximos. Isso é importante porque no ICA os cluster-heads não enviam as mensagens diretamente para a estação base, eles enviam para o cluster-head mais próximo, na direção da estação base, fazendo assim uma economia de energia e aumentando o tempo de vida da rede.

No LEACH, os cluster-heads transmitem suas mensagens diretamente para a estação base. Embora haja uma mudança periódica de cluster-heads, o consumo de energia tem relação quadrática com a distância. O ICA resolve este problema utilizando comunicação multi-hop entre os líderes, no entanto, os líderes mais próximos da estação base terão um gasto maior de energia, pois receberão os dados de toda rede e retransmitirão à estação base, que fica distante. Para evitar a morte prematura dos nós perto da estação base, os cluster-heads podem recusar retransmitir mensagens de outros clusters para a estação base. Quando o cluster-head percebe que está ficando sem energia, ele para de enviar mensagens para a estação base. Isso acontece quando sua energia atinge o limite determinado pela equação:

$$E = d * NB, \quad (3.5)$$

onde E é a quantidade mínima de energia que o nó deve possuir para agir como ponte, d é a distância entre o nó e a estação base e NB é a média entre o número de bytes transmitidos em mensagens passadas e o tamanho da mensagem atual. E quando ocorre uma recusa em transmitir dados, o cluster-head que requisitou o serviço envia a mensagem diretamente para a estação base. A figura 12 ilustra este processo.

Portanto, no ICA, os nós próximos da estação base, que gastam mais energia, garantem a transmissão dos dados de seus próprios clusters à estação base. Entretanto, os nós mais distantes, que utilizam menos energia durante o roteamento, passam a gastar a energia economizada para enviar suas mensagens diretamente à estação base. Assim, o ICA pretende aumentar o tempo de vida da rede.

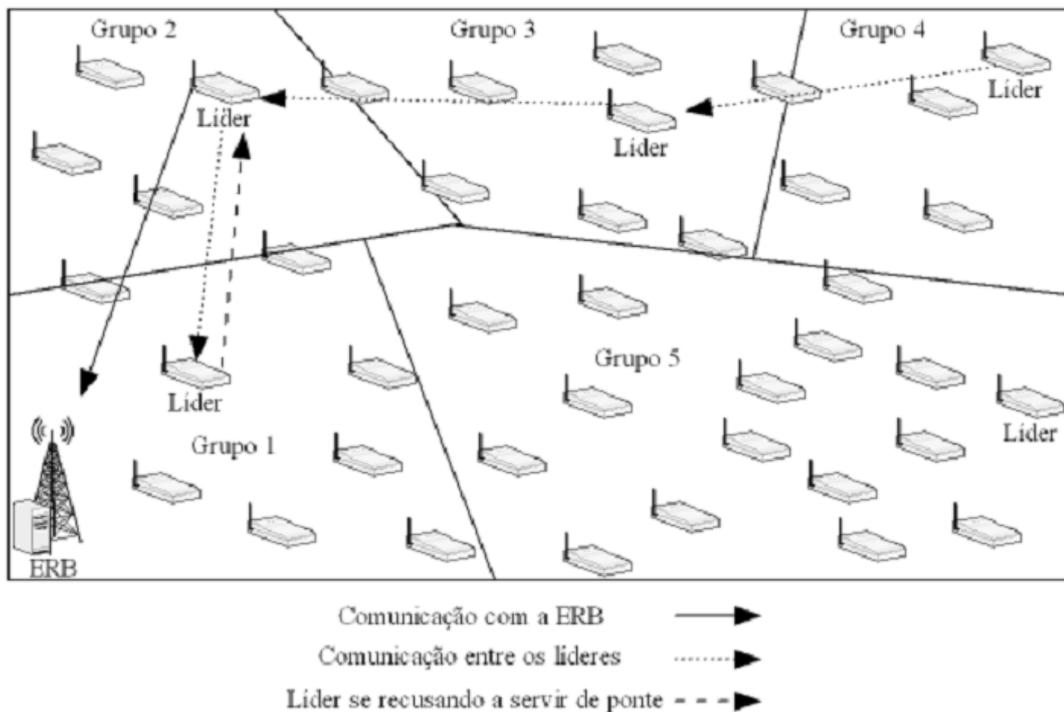


Figura 12: Cenário em que o líder do cluster 1 se recusa a servir de ponte para o cluster 2.

3.2.4 BCDCP

O BCDCP (Base Station Controlled Dynamic Protocol) é um protocolo de roteamento de redes de sensores que usa a estação base como o componente responsável pela organização dos nós. Este protocolo opera em duas fases: configuração e comunicação de dados.

Na fase de configuração o algoritmo realiza as atividades de formação de cluster, seleção de cluster-head (CH), formação de rota multi-hop entre os cluster-heads (CH-to-CH) para roteamento, e criação de agenda para cada cluster. Assim como no LEACH-C, durante a fase de configuração a estação base recebe informação do status de energia de todos os nós da rede, calcula a média e cria um conjunto de nós elegíveis S , que são os nós que têm seus níveis de energia maiores que a média. Os cluster-heads para a rodada são escolhidos do conjunto S , com a finalidade de prolongar o tempo de vida da rede. As maiores tarefas da estação base são:

- Identificar os nós cluster-heads N_{ch} escolhidos no conjunto S .
- Agrupar os outros nós em clusters de modo que o consumo global de energia seja minimizado durante a fase de comunicação de dados.

No BCDCP, essas tarefas são feitas por meio de um algoritmo iterativo de divisão de cluter. Este algoritmo, primeiro divide a rede em dois subclusters, depois vai dividindo em subclusters ainda menores. A estação base repete o processo de divisão de cluster até encontrar

o número desejado de cluster-heads. Este algoritmo garante que o número de cluster-heads selecionados sejam colocados uniformemente na rede, maximizando a distância entre os cluster-heads em cada etapa de divisão. Além disso, o algoritmo utiliza a técnica de balanceamento de cluster, onde na divisão dos clusters, os subclusters terão aproximadamente o mesmo número de nós sensores.

Depois de formar os clusters, o protocolo BCDCP usa um esquema de roteamento multihop CH-to-CH para transmitir os dados para a estação base. As rotas são selecionadas conectando todos os cluster-heads usando a abordagem árvore mínima de spanning, que minimiza o consumo de energia em cada cluster-head. Que minimiza o consumo de energia para cada nó, e então escolhe aleatoriamente um cluster-head para transmitir os dados para a estação base, para distribuir a carga de trabalho entre os cluster-heads.

Por fim, na fase de configuração, o protocolo utiliza um esquema de agendamento TDMA para minimizar as colisões entre os nós sensores, quando estes transmitem os dados para o cluster-head.

A fase de comunicação de dados consiste de três principais tarefas: coleta de dados, fusão de dados e roteamento de dados. Como os nós sensores estão bem agrupados na rede, suas transmissões para cluster-head consomem pouca energia. Uma vez que os dados coletados de todos os sensores são recebidos pelos cluster-heads, estes realizam a fusão dos dados, reduzindo a quantidade de dados a ser enviados para a estação base. Estes dados são enviados através de comunicação multi-hop entre os cluster-heads.

3.2.5 PSO-C

O PSO-C (Particle Swarm Optimization - Centralizado) (LATIFF; TSIMENIDIS; SHARIF, 2007) é um protocolo de clusterização em RSSF eficiente em energia, e que utiliza o algoritmo PSO para encontrar os clusters. A ideia principal deste protocolo é minimizar a distância intra-cluster entre os nós dos clusters e seus cluster-heads, e ao mesmo tempo otimizar o gerenciamento de energia na rede.

O protocolo consiste de duas fases: formação de grupo e comunicação. A formação dos clusters no PSO-C é realizada na estação base. Ele opera em rodadas, onde cada rodada é iniciada com a formação de clusters. A fase de comunicação é semelhante à do protocolo LEACH. Assim como no LEACH-C, no início da formação dos clusters, todos os nós da rede enviam seus atuais níveis de energia e localização geográfica para a estação base. Para garantir que somente os nós com boa quantidade de energia serão escolhidos como cluster-heads, apenas os nós com quantidade de energia acima da média são os candidatos a cluster-head. Depois a estação base executa o algoritmo PSO para encontrar os k melhores cluster-heads que podem minimizar a função de custo definida por:

$$cost = \beta \times f_1 + (1 - \beta) \times f_2 \quad (3.6)$$

$$f_1 = \max_{k=1,2,\dots,K} \left\{ \sum_{\forall n_i \in C_{p,k}} \frac{d(n_i, CH_{p,k})^2}{|C_{p,k}|} \right\} \quad (3.7)$$

$$f_2 = \frac{\sum_{i=1}^N E(n_i)}{\sum_{k=1}^K E(CH_{p,k})} \quad (3.8)$$

onde f_1 é a média máxima da distância euclidiana dos nós sensores aos cluster-heads associados e $|C_{p,k}|$ é o número de nós que participam do cluster C_k da partícula p . A função f_2 é a razão entre o total da energia inicial de todos os nós e n_i , $i = 1, 2, \dots, N$ na rede e o total da energia atual dos candidatos a cluster-heads na rodada corrente. A constante β é uma constante definida pelo usuário para ponderar a contribuição de cada uma das funções sub-objetivas (f_1 e f_2). A função de custo tem o objetivo de minimizar a distância intra-cluster dos nós aos seus cluster-heads, através da função f_1 ; e também otimizar a eficiência de energia da rede, através da função f_2 . Então, conforme a função de custo do protocolo, um pequeno valor de f_1 e f_2 sugere clusters compactos com um ótimo conjunto de nós que têm energia suficiente para executar as tarefas atribuídas aos cluster-heads.

A tabela 1 mostra as características dos protocolos descritos neste capítulo.

Protocolo	Topologia	Clusterização	Comunicação multihop entre cluster-heads	Utiliza posição dos nós
LEACH	Hierárquica	Distribuída	Não	Não
LEACH-C	Hierárquica	Centralizada	Não	Sim
ICA	Hierárquica	Distribuída	Sim	Sim
BCDCP	Hierárquica	Centralizada	Sim	Sim
PSO-C	Hierárquica	Centralizada	Não	Sim

Tabela 1: Características dos protocolos

3.3 Síntese do Capítulo

Este capítulo apresentou os trabalhos relacionados com a presente proposta. Inicialmente, foram mostradas duas abordagens baseadas no comportamento de abelhas (Bee Algorithm e HBMO) para clusterização de dados. Logo em seguida, foram apresentados os protocolos de clusterização em Redes de Sensores Sem Fio que têm fundamental importância para este trabalho.

O próximo capítulo descreverá o BEE-C, que é algoritmo de roteamento proposto neste trabalho.

4 BEE-C: ALGORITMO BIO-INSPIRADO EM ABELHAS PARA CLUSTERIZAÇÃO EM REDES DE SENSORES

Neste capítulo é apresentada a proposta para este trabalho, que é um algoritmo de roteamento baseado no comportamento de abelhas para redes de sensores sem fio.

4.1 Pressuposições

- A estação base é fixa e fica longe dos nós.
- Os nós são igualmente equipados com recursos de controle que podem variar sua potência de transmissão.
- Todos os nós sensores são imóveis.
- Os nós sensores tem conhecimento de sua localização.

4.2 Funcionamento do Protocolo BEE-C

O protocolo proposto funciona por rodadas, o tempo de cada rodada é definido como parâmetro do protocolo. Cada rodada é composta por duas fases: formação dos clusters e comunicação dos dados. Na primeira fase, é realizada a formação dos grupos. O processo de formação dos clusters é realizada de maneira centralizada. O início deste processo é realizado de forma semelhante ao LEACH-C, onde cada nó i envia para a estação base a informação de sua posição geográfica e quantidade de energia E_i , com estes dados, a estação base calcula a média geral de energia $Media_E$, através da equação seguinte.

$$Media_E = \frac{\sum_{i=1}^n E_i}{n} \quad (4.1)$$

onde n é o número de nós da rede.

Somente os nós que tiverem seu nível de energia acima da média são elegíveis. A probabilidade P para um nó i ser elegível é dada por:

$$P(i) = \begin{cases} 1, & E_i \geq Media_E \\ 0, & E_i < Media_E \end{cases} \quad (4.2)$$

Para encontrar os melhores clusters, a estação base executa um algoritmo bio-inspirado em abelhas baseado no HBMO, descrito no capítulo 3, para minimizar ou maximizar (dependendo da função) a função objetivo do algoritmo. Para esta proposta cada nó da rede é considerado um ponto cartesiano, os nós são agrupados pela distância mínima e os centróides são os líderes dos grupos. Sendo que cada abelha do algoritmo é representada pelo conjunto de líderes da rede.

A figura 13 mostra as etapas da fase de formação de grupos do BEE-C.

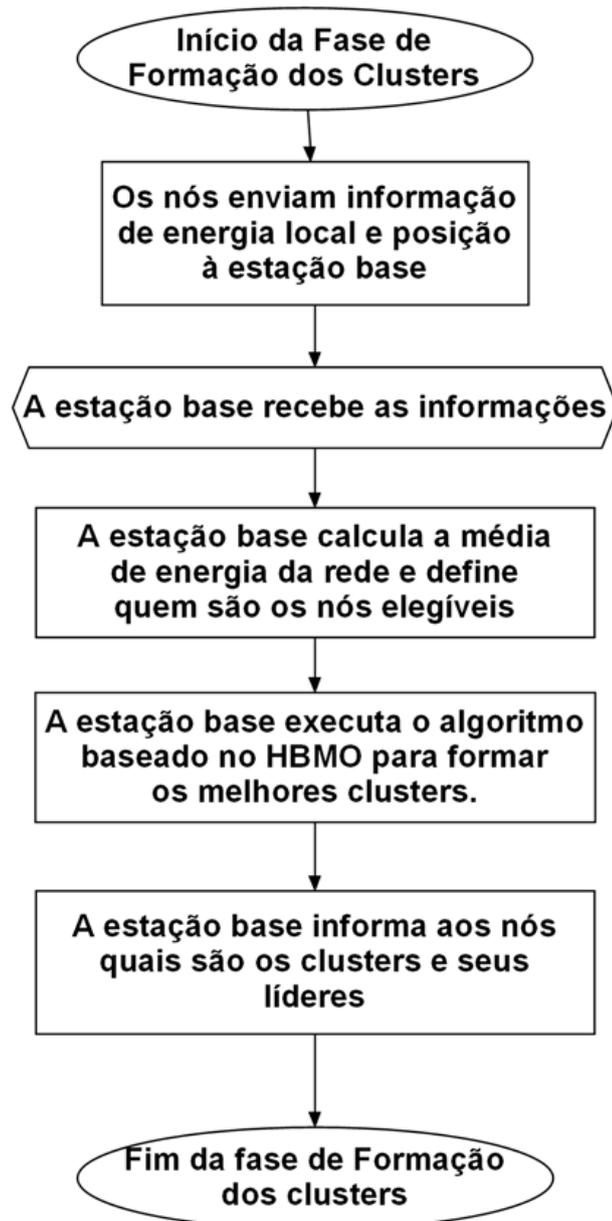


Figura 13: Fase de formação de grupos do BEE-C

A figura 14 apresenta o algoritmo de formação de grupos do BEE-C.

As subseções seguintes explicam detalhes do algoritmo.

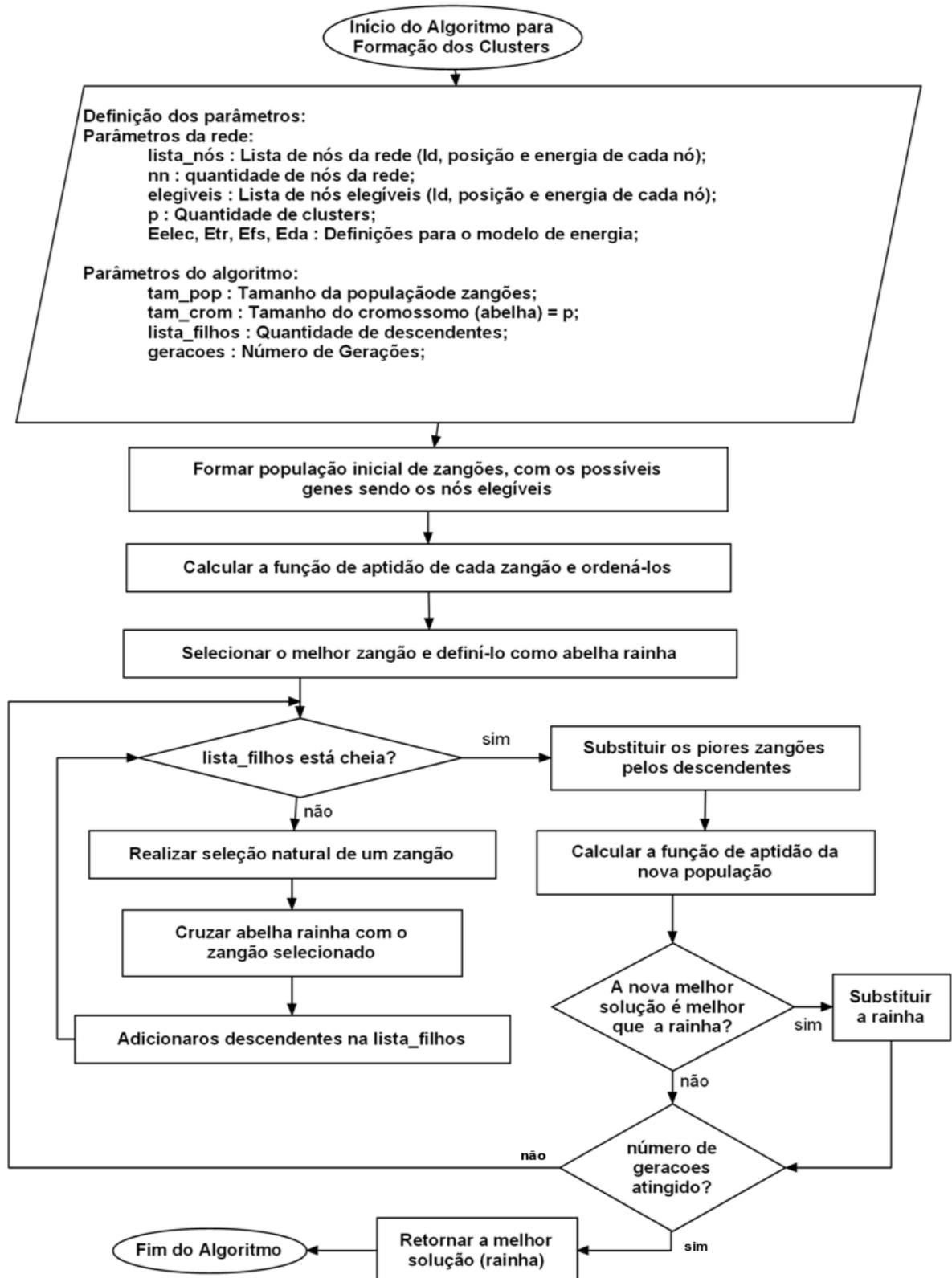


Figura 14: Algoritmo de formação de grupos do BEE-C

4.2.1 Representação da solução do BEE-C

A solução do algoritmo é representada por um cromossomo, sendo este um zangão, uma rainha ou um descendente. Cada cromossomo contém um conjunto de genes, sendo que cada gene representa o ponto cartesiano de um nó líder. A figura 15 ilustra uma rede clusterizada.

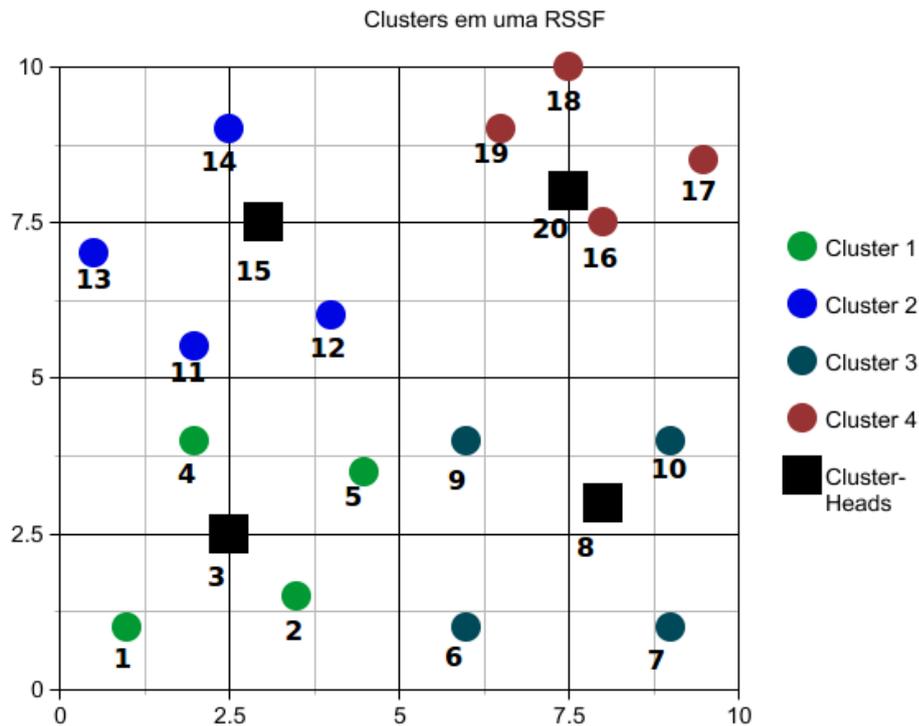


Figura 15: Exemplo de uma rede de sensores clusterizada

A solução do algoritmo (cromossomo) é formada pelo conjunto de cluster-heads. A figura 16 mostra como fica o cromossomo para a rede mostrada na figura 15.

Para não confundir é importante notar que uma abelha (zangão ou rainha) é um cromossomo, que é um conjunto de cluster-heads, os quais formam uma possível solução para o algoritmo.

3	8	15	20
----------	----------	-----------	-----------

Figura 16: Solução para o problema de clusterização (conjunto de nós cluster-heads)

Cada cromossomo (solução) contém um conjunto de genes (cluster-heads), sendo que cada gene é um objeto que contém as informações de id e localização, como ilustra a figura 17.

ID: 3
X: 2,5
Y: 2,5

Figura 17: Representação de um gene

4.2.2 Função Objetivo

A função objetivo, também chamada de fitness ou função de aptidão, é o critério utilizado para saber o quão bom é um indivíduo.

Para calcular a função objetivo do algoritmo é necessário que seja definido todos os clusters para cada solução, atribuindo cada nó ao cluster cujo cluster-head é mais próximo.

Este trabalho apresenta duas opções para função de aptidão.

4.2.2.1 Primeira Função Objetivo

A primeira opção de função objetivo para este trabalho é baseada no modelo de consumo de energia da rede. Esta função tenta prever a soma da quantidade média de energia dos nós de cada cluster que a rede terá depois da rodada. Assim, o algoritmo deve maximizar esta função.

Inicialmente é calculada a quantidade de energia E_1 que o cluster-head do cluster k irá gastar para enviar os pacotes à estação base através da equação 4.3:

$$E_1 = N_k \cdot (E_{elec} + \epsilon_{tr} \cdot d_{CH-to-BS}^4) \quad (4.3)$$

onde N_k é o número de nós do cluster k , E_{elec} é energia gasta por bit transmitido ou recebido, ϵ_{tr} é o parâmetro para o modelo de consumo de energia de dois raios, utilizado para grandes distâncias entre a origem e o destino e $d_{CH-to-BS}$ é a distância do cluster-head à estação base.

Depois é feito o cálculo da quantidade de energia E_2 que os nós fontes irão gastar para enviar os dados aos cluster-heads utilizando a equação 4.4:

$$E_2 = \left(\sum_{j=1}^{N_k} E_{elec} + \epsilon_{fs} \cdot d_{no_j-to-CH}^2 \right) \quad (4.4)$$

onde ϵ_{fs} é o parâmetro para o modelo de consumo em espaço livre, utilizado quando a distância entre a origem e o destino é pequena, $d_{no_j-to-CH}$ é a distância do nó j ao cluster-head.

Em seguida, na equação 4.5, calcula-se a energia E_3 consumida pelos cluster-heads para fusão de dados:

$$E_3 = N_k \cdot E_{BF} \quad (4.5)$$

onde E_{BF} é a energia consumida por bit para fusão dos dados.

E por último, na equação 4.6, é calculada a energia consumida pelos cluster-heads E_4 para receber os dados dos nós fontes:

$$E_4 = (N_k - 1) \cdot E_{elec} \quad (4.6)$$

A energia total consumida E_{C_k} pelo cluster k é:

$$E_{C_k} = E_1 + E_2 + E_3 + E_4 \quad (4.7)$$

que é equivalente a:

$$E_{C_k} = N_k \cdot (E_{elec} + E_{DA} + \epsilon_{tr} \cdot d_{CH-to-BS}^4) + \left(\sum_{j=1}^{N_k} E_{elec} + \epsilon_{fs} \cdot d_{no-to-CH}^2 \right) + (N_k - 1) \cdot E_{elec} \quad (4.8)$$

A energia total E_{T_k} do cluster k é:

$$E_{T_k} = \sum_{j=1}^{N_k} E_{no_j} \quad (4.9)$$

Por último a função F é calculada obtendo-se a soma das médias da energia restante dos nós de cada cluster:

$$F = \sum_{k=1}^p \frac{E_{T_k} - E_{C_k}}{N_k} \quad (4.10)$$

onde p é número de clusters da rede.

4.2.2.2 Segunda Função Objetivo

Para esta segunda opção de função objetivo do algoritmo foi utilizada a função objetivo do LEACH-C e acrescentadas outras duas funções.

A primeira função é:

$$f_1 = \sum_{i=1}^N \min_{k \in K} d^2(i, k) \quad (4.11)$$

onde N é o conjunto de nós da rede, K é o conjunto de cluster-heads e d é a distância. Esta função busca o menor custo para os nós fontes da rede, diminuindo suas distâncias aos cluster-heads.

A segunda função é:

$$f_2 = \frac{\sum_{i=1}^N E(n_i)}{\sum_{k=1}^K E(CH_k)} \quad (4.12)$$

onde $E(n_i)$ é a energia do nó i , e $E(CH_k)$ é a energia do cluster-head k . Minimizando esta função são encontrados o conjunto de cluster-heads com mais energia.

A terceira função é:

$$f_3 = \sum_{k=1}^K d(CH_k - to - BS) \quad (4.13)$$

onde $d(CH_k - to - BS)$ é a distância do cluster-head k à estação base. A minimização desta função encontra os cluster-heads com menores distância à estação base, que objetiva o menor consumo ao transmitir os dados.

O equilíbrio entre as três funções visa encontrar os clusters onde os nós fontes estejam próximos aos cluster-heads (f_1), os cluster-heads tenha muita energia (f_2) e gastem pouca energia ao transmitir f_3 .

A função objetivo é calculada multiplicando as três funções:

$$F = f_1 \times f_2 \times f_3. \quad (4.14)$$

4.2.3 População Inicial e Escolha da Abelha Rainha

A população inicial de abelhas é formada por um conjunto aleatório de soluções possíveis. Depois de formada a população inicial de zangões, é feito o processo de escolha da abelha rainha, que é a melhor abelha da população. O pseudocódigo para este processo é apresentado na figura 18.

4.2.4 Seleção dos Zangões

Após a abelha rainha ter sido escolhida, é feita a seleção do zangão, de acordo com seu nível de aptidão. Para este protocolo são elaborados dois métodos de seleção. O primeiro é o método baseado em Têmpera Simulada, que é próprio do HBMO. O segundo é um método mais simples e mais usado na literatura, que é o roleta.

4.2.4.1 Seleção por Têmpera Simulada

A seleção por Têmpera Simulada faz uma alusão bem interessante ao comportamento de seleção natural que a abelha rainha realiza na seleção dos zangões para o acasalamento.

Para entender esse algoritmo de seleção é interessante saber como esse processo ocorre biologicamente. Antes do acasalamento, a rainha executa uma dança e logo depois começa a voar. Os zangões começam a voar também para acompanhar a rainha e acasalar com ela no ar. Em cada acasalamento, o esperma do atinge a espermateca e acumula, para formar o pool genético da colônia. No início do voo, a rainha começa com uma certa quantidade de energia, e com uma velocidade maior, e retorna ao seu ninho quando está com pouca energia e sua espermateca está cheia.

No algoritmo, um zangão Z cruza com a rainha R probabilisticamente usando a seguinte

```

Entrada:
  m                //Número de zangões
  p                //Tamanho do cromossomo
  n                //Quantidade de nós da rede
  nos              //Lista de nós da rede
  mediaEnergia    //Calcular a média de energia da rede
  pop              //Lista de cromossomos
Saída: população inicial e abelha rainha

  Inicializar;
  Calcular a média de energia da rede mediaEnergia;

  para i de 1 até n faça
    | se energiaNo[i] > mediaEnergia então
    | | no[i] é elegível ;
    | fim
  fim

  para i de 1 até tampop faça
    | Criar um cromossomo cTemp ;
    | enquanto número de genes de cTemp < p faça
    | | Selecionar um nó randno aleatoriamente ;
    | | se randno é elegível então
    | | | cTemp adiciona randno como gene ;
    | | fim
    | fim
    | pop adiciona cTemp ;
  fim

  A rainha r é o melhor cromossomo de pop;
  Retorna a população inicial e a abelha rainha;

```

Figura 18: Algoritmo para população inicial e escolha da abelha rainha

função:

$$P(Z,R) = \exp[-\Delta(f)/V(t)] \quad (4.15)$$

onde $P(Z,R)$ é a probabilidade do zangão Z adicionar esperma na espermateca da rainha R (ou seja, a probabilidade de cruzamento); $\Delta(f)$ é a diferença absoluta entre a aptidão (*fitness*) do zangão Z e a aptidão da rainha R ; e $V(t)$ é a velocidade da rainha no tempo t . Assim a probabilidade de seleção do zangão é alta no início do voo ou quanto a aptidão do zangão tão boa quanto a da rainha. Com o passar das iterações a velocidade e energia da rainha vai diminuindo de acordo com a seguinte equação:

$$S(t+1) = \alpha \times V(t) \quad (4.16)$$

onde α é um fator (entre 0 e 1), que indica a redução da velocidade depois de cada transição.

A figura 19 apresenta o pseudocódigo do método de seleção por Têmpera simulada.

```

Entrada:
Z           //Vetor de zangões
R           //Rainha das abelhas
Vmax      //Velocidade da Abelha ao iniciar o voo
Vmin      //Velocidade da Abelha no fim do voo
v           //Esquema de redução de velocidade
G           //Número de iterações
V = Vmax   //Setar V como velocidade inicial
Saída: Zangão selecionado

repita
  Selecionar um Zi de Z aleatoriamente;
  Calcular  $\Delta(f) = |f(R) - f(Z_i)|$ ;
  Gerar r aleatoriamente;
  se  $\exp(-\Delta(f)/V) > r$  então
    | Selecionar Zi;
  fim
  senão
    | V = v * V
  fim
até Até selecionar um zangão ;
Retornar o zangão Zi selecionado;

```

Figura 19: Algoritmo de Seleção por Têmpera Simulada

4.2.4.2 Seleção por Roleta

Este método foi sugerido por (BAKER, 1987), e é semelhante a uma roleta, onde cada indivíduo tem sua parte da roleta proporcional à sua função de aptidão, ou seja, quanto melhor o indivíduo, maior sua chance de ser selecionado na roleta.

O método da roleta é o mais simples e atualmente o mais utilizado.

É importante observar que todos os indivíduos têm chance de ser selecionados, entretanto quanto pior a aptidão, menor a fatia na roleta.

A figura 20 apresenta o pseudocódigo do método de seleção por roleta.

```

Entrada:
Z //Vetor de zangões
fitnessZangao //Aptidão da cada zangão
somaFitTotal = 0 //Soma das aptidões dos zangões
somaFitness = 0 //soma atualizada até selecionar o zangão
iSelecionado = 0 //índice selecionado
Saída: Zangão selecionado

Ordenar os Zangões Z de acordo com o valor de fitnessZangao;
Calcular a somaFitTotal;
Gerar r (entre 0 e somaFitTotal) aleatoriamente;
repita
| somaAptidao += fitnessZangao[iSelecionado];
| iSelecionado += 1;
até Até r > somaFitness ;
Retornar o zangão ZiSelecionado;

```

Figura 20: Algoritmo de Seleção por Roleta

4.2.5 Acasalamento ou Cruzamento

Este processo é o mais importante para a evolução do algoritmo. O método de cruzamento utilizado para esta proposta é o de máscara binária. No cruzamento por máscara binária é gerada de uma máscara de números binários, com o mesmo tamanho de um cromossomo, que define os genes de cada cromossomo que cada filho herdará. Na figura 21, pode ser observado um exemplo de cruzamento causado por este operador. No caso, o valor 1 (um) na máscara indica que o filho 1 herdará o gene correspondente do abelha rainha, e o filho 2 herdará o gene correspondente do zangão, o valor 0 (zero) indica o contrário.

A figura 22 mostra o pseudocódigo do cruzamento.

4.2.6 Busca Local ou Mutação

Logo após o processo de acasalamento, é realizada uma busca local nos descendentes. Este processo é realizado substituindo um gene do cromossomo por um novo gene. Então, caso o resultado dessa alteração seja bom, o cromossomo da abelha descendente é substituído. A figura 23 ilustra esse processo.

O pseudocódigo para esta etapa é mostrado na figura

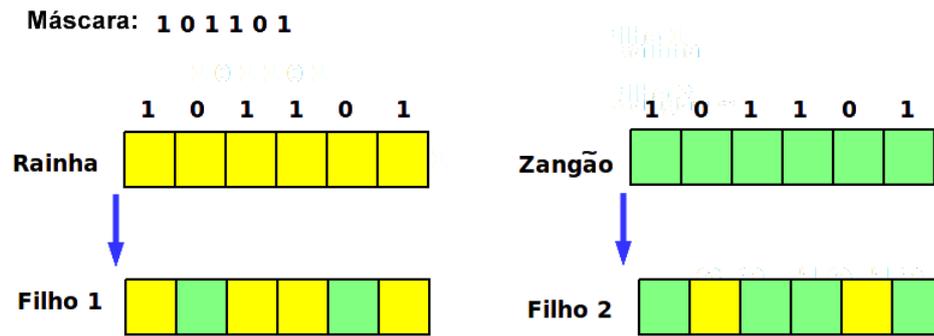


Figura 21: Cruzamento entre zangão e rainha por máscara binária

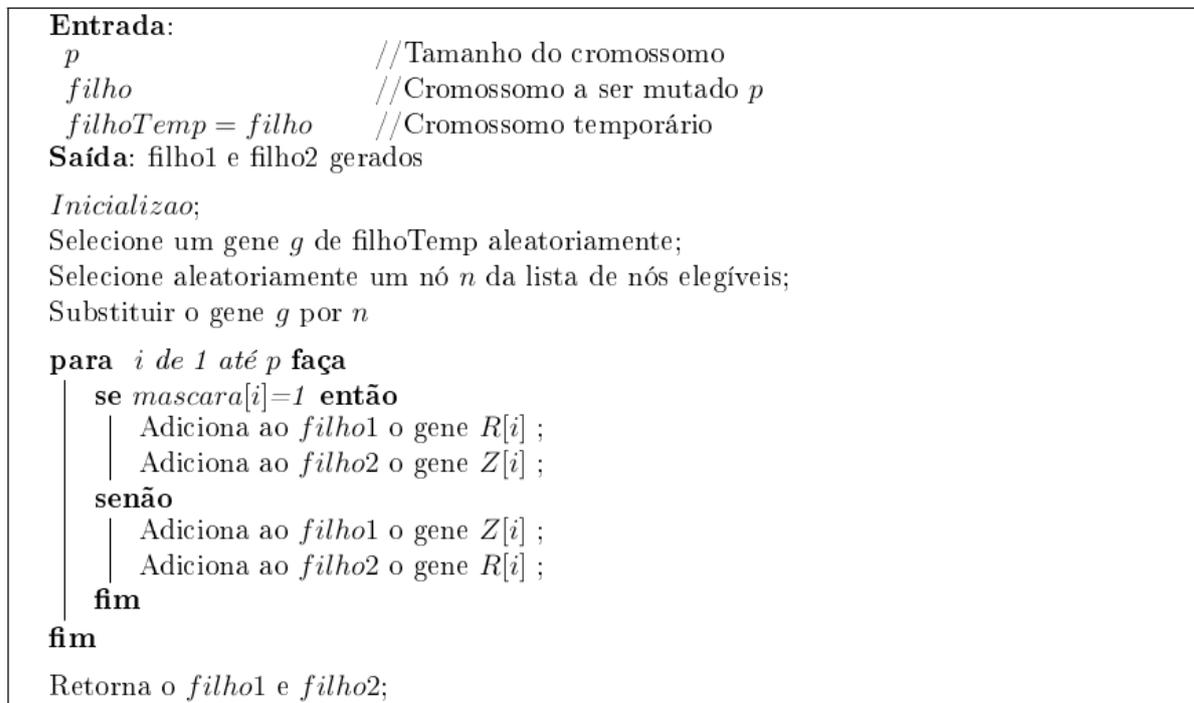


Figura 22: Algoritmo de cruzamento por máscara binária

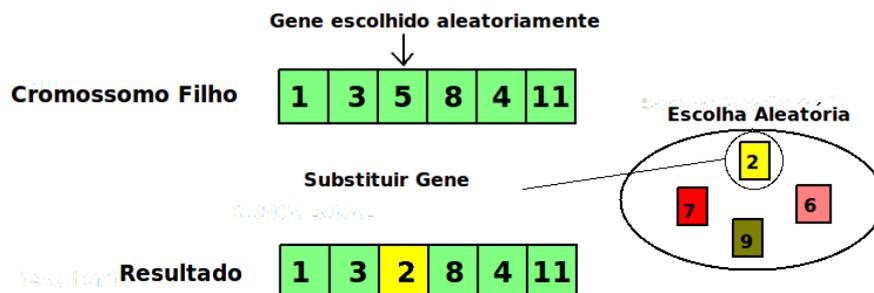


Figura 23: Exemplo de mutação ou busca local em um cromossomo

```

Entrada:
  p                //Tamanho do cromossomo
  filho            //Cromossomo a ser mutado p
  filhoTemp = filho //Cromossomo temporário
Saída: Filho mutado

  Inicializao;
  Selecione um gene g de filhoTemp aleatoriamente;
  Selecione aleatoriamente um nó n da lista de nós elegíveis;
  Substituir o gene g do filhoTemp por n;

  se filhoTemp é melhor que filho então
  | filho=filhoTemp;
  fim

  Retorna o filho;

```

Figura 24: Algoritmo de mutacao

4.2.7 Substituição da Abelha Rainha e Novas Gerações

Logo após a formação dos descendentes, caso algum destes seja melhor que a rainha, é realizada a substituição da rainha.

Após isso, os descendentes substituem os piores zangões e é iniciada uma nova iteração com todos os passos: seleção, cruzamento, busca local e substituição da abelha rainha.

4.3 Síntese do Capítulo

Este capítulo apresentou basicamente o funcionamento do BEE-C, focando, principalmente no algoritmo para clusterizar os nós da rede. Para facilitar o entendimento do algoritmo, foram apresentadas as etapas do algoritmo em subseções distintas, com algumas ilustrações e pseudocódigos.

O próximo capítulo apresenta as simulações e resultados do BEE-C, comparados com o LEACH e LEACH-C.

5 SIMULAÇÕES E RESULTADOS

Este capítulo apresenta as simulações do algoritmo BEE-C, e os resultados comparados com os dos protocolos LEACH e LEACH-C.

Para simular o BEE-C, foi estendido o módulo do simulador NS (Network Simulator) desenvolvido pelo projeto uAMPS, no qual foram desenvolvidos os protocolos LEACH e o LEACH-C. Portanto, foram utilizados os mesmos parâmetros de simulação da implementação original do LEACH-C (modelo de tráfego, energia, rádio e propagação de sinal).

5.1 Ambiente de Utilizado

O ambiente utilizado foi o NS-2.1b5, o mesmo no qual foi o desenvolvido o módulo original do projeto uAMPS.

5.2 Cenário

A tabela 2 mostra os cenários utilizados para as simulações:

Cenário	Quantidade de nós	Área (m x m)	Posição da Estação Base
1	100	100 X 100	(50, 175)
2	200	100 X 100	(50, 175)
3	100	200 X 200	(100, 275)
4	200	200 X 200	(100, 275)

Tabela 2: Cenários utilizados nas simulações

Nos experimentos os nós foram depositados aleatoriamente na rede, variando 10 vezes em cada cenário, e em cada variação foram realizadas 5 simulações, resultando um total de 50 simulações por cenário para cada protocolo.

A tabela 3 mostra as densidades da rede em cada cenário.

Cenário	Densidade
1	0,0100
2	0,0200
3	0,0025
4	0,0050

Tabela 3: Densidade da rede em cada cenário

5.2.1 Parâmetros da rede para as simulações

Nas simulações, os rádios podem controlar suas potências de transmissão de modo a consumir o mínimo de energia para atingir seu destino. Os nós podem ser desligados caso não estejam em uso, a fim de economizar energia.

Cada nó na rede possui 1 Joule de energia e o tamanho dos pacotes enviados é de 500 bytes. A simulação acaba quando menos de 5% dos nós da rede se encontram ativos.

A tabela 5.3 mostra as características do rádio para a rede.

<i>Descrição</i>	<i>Parâmetro</i>	<i>Valor</i>
Distância de cruzamento entre os modelos de radio propagação em espaço livre (Friis) e o modelo de dois raios (two ray)	$d_{crossover}$	$\frac{4\pi h_t h_r}{\lambda}$
Energia gasta na transmissão dos dados	P_t	$\epsilon_{friss-amp} R_b d^2 : d < d_{crossover}$ $\epsilon_{two-ray-amp} R_b d^4 : d \geq d_{crossover}$
Energia gasta na recepção dos dados	P_r	$\frac{\epsilon_{friss-amp} R_b G_t G_r \lambda^2}{(4\pi)^2} : d < d_{crossover}$ $\epsilon_{two-ray-amp} R_b G_t G_r h_t^2 h_r^2 : d \geq d_{crossover}$
Mínimo de energia do receptor necessário para recepção com sucesso	$P_{r-thresh}$	$6.3nW$
Energia do amplificador de radio	$\epsilon_{friss-amp}$ $\epsilon_{two-ray-amp}$	$\frac{P_{r-thresh}(4\pi)^2}{R_b G_t G_r \lambda^2}$ $\frac{P_{r-thresh}}{R_b G_t G_r h_t^2 h_r^2}$
Energia eletrônica do radio	E_{elec}	$50nJ/bit$
Cálculo de energia para conformação de feixe	E_{BF}	$5nJ/bit$
Taxa de bits	R_b	$1Mbps$
Fator de ganho da antena	G_t, G_r	1
Altura da antena acima do solo	h_t, h_r	1.5 m
Comprimento de onda do sinal	λ	0.325 m
Distância de cruzamento entre os modelos de radio propagação em espaço livre (Friis) e o modelo de dois raios (two ray)	$d_{crossover}$	87 m
Energia do amplificador de radio	$\epsilon_{friss-amp}$ $\epsilon_{two-ray-amp}$	$10pJ/bit/m^2$ $0.0013 pJ/bit/m^4$

Tabela 4: Características e parâmetros do rádio

Dois fatores são muito importantes para a realização das simulações: quantidade de clusters que a rede deve ter e o tempo de duração de cada rodada.

5.2.1.1 Quantidade de clusters da rede

Em (HEINZELMAN, 2000) foi elaborado uma equação matemática que calcula o número k ótimo de clusters que uma rede deve ter. Esta equação é dada por:

$$k = \frac{\sqrt{N}}{\sqrt{2\pi}} \sqrt{\frac{\epsilon_{friss-amp}}{\epsilon_{two-ray-amp}} \frac{M}{d_{toBS}^2}} \quad (5.1)$$

onde:

N = número de nós da rede;

$\epsilon_{friss-amp}$ = energia de amplificação do radio transmissor para $d < d_0$;

$\epsilon_{two-ray-amp}$ = energia de amplificação do radio transmissor para $d > d_0$;

d_0 = distância de cruzamento para os modelos *Friss* e *two-ray*;

M = considerando uma rede de área $100m \times 100m$, $M = 100$;

d_{toBS} = distância dos nós para a estação base;

Com base na equação 5.1, considerando os valores $\epsilon_{friss-amp} = 10$ pJ e $\epsilon_{two-ray-amp} = 0.0013$ pJ. a tabela 5 mostra a quantidade de cluster heads para cada cenário.

Cenário	Distância para Estação base (d)	Número de nós (n)
1	$75 < d < 185$	$1 < n < 6$
2	$75 < d < 185$	$1 < n < 8$
3	$75 < d < 285$	$1 < n < 12$
4	$75 < d < 285$	$1 < n < 17$

Tabela 5: Quantidade de clusters possíveis para cada cenário

Em (HEINZELMAN, 2000) foram feitos vários testes para o cenário 1 da tabela 2, variando o número de clusters. A rede obteve os melhores resultados quando a quantidade de clusters estava definida em 5% do total de nós. Para os demais cenários, realizamos testes que também obtiveram melhores resultados com a quantidade de clusters igual a 5%. Portanto, para todos os cenários, a quantidade de clusters utilizada nas simulações é 5% do total de nós da rede. Este é um valor muito utilizado na literatura para diversos tipos de cenários ().

5.2.1.2 Tempo para cada rodada

Em (HEINZELMAN, 2000), o tempo ótimo para cada rodada é dado por:

$$T_{rodada} = 0.08 \text{segundos} * \frac{E_{inicial}}{9mJ} \quad (5.2)$$

onde T_{rodada} é o tempo de cada rodada e $E_{inicial}$ é a quantidade de energia inicial dos nós. Para nossas simulações, como foi descrito anteriormente, $E_{inicial} = 1$ J. Assim o tempo de cada rodada $T_{rodada} \approx 10$ segundos.

5.2.2 Parâmetros para o algoritmo baseado no HBMO

Para o algoritmo bio-inspirado utilizado para clusterizar a rede, são considerados os seguintes parâmetros:

- Tamanho da população = 50;
- Tamanho do cromossomo = 5% do total de nós da rede (Quantidade de clusters);
- Tamanho da espermateca = 20;
- Método de seleção = roleta;
- Número de gerações = 100.

5.3 Métricas para Avaliação

Para a avaliação dos resultados, são consideradas as seguintes métricas:

- Quantidade de dados enviados à estação base;
- Tempo de vida da rede;
- Tempo de cobertura total da rede (Tempo até a morte do primeiro nó)
- Consumo de energia.

5.4 Resultados

Esta seção analisa os resultados das simulações e compara os resultados obtidos pelo BEE-C com os resultados obtidos pelo LEACH e LEACH-C. Foram considerados duas versões do BEE-C, as quais chamamos BEE-C1 e BEE-C2, pois foram desenvolvidas duas funções objetivos, o BEE-C1 utiliza a primeira função objetivo descrita na seção 4.2.2 do capítulo 4, já o BEE-C2 utiliza a segunda função objetivo descrita na mesma seção.

Os parâmetros e métricas utilizados nas simulações foram apresentados nas seções anteriores. As subseções seguintes mostram os resultados para cada métrica utilizada. Teremos como cenário padrão, o cenário 1 da tabela 2, que foi o mesmo cenário utilizado para mostrar os resultados dos protocolos LEACH e LEACH-C em (HEINZELMAN, 2000).

5.4.1 Quantidade de dados enviados à estação base

A figura 25 mostra o número médio de pacotes enviados pelos protocolos em cada cenário. Pode-se observar que em todos os cenários o número de pacotes enviados pelo BEE-C1 foi superior aos demais, sendo que o BEE-C2 também conseguiu enviar mais pacotes que o LEACH-C e o LEACH. A média do ganho no envio de pacotes do BEE-C1 em relação ao LEACH-C foi de aproximadamente 4%, já o BEE-C2 conseguiu um ganho médio de 1% em relação ao LEACH-C. No cenário 1 o BEE-C1 conseguiu o maior ganho em relação ao LEACH-C, que foi de aproximadamente 6%, já o BEE-C2 teve um ganho de pouco mais de 1%. No entanto, pode-se observar que quando a densidade da rede aumenta (Cenário 2), o ganho do BEE-C1 diminui para 1,5% e o ganho do BEE-C2 foi de apenas 0,5% em relação ao LEACH-C. Já o protocolo LEACH nos cenários 2 e 4, em que a rede possui 200 nós, teve uma grande queda no envio dos dados, que acontece porque diferente do BEE-C e do LEACH-C, o LEACH forma os clusters de forma distribuída, através da troca de mensagens entre os nós, e aumentando o número de nós, aumenta o número de colisões e diminui a eficiência na formação dos grupos.

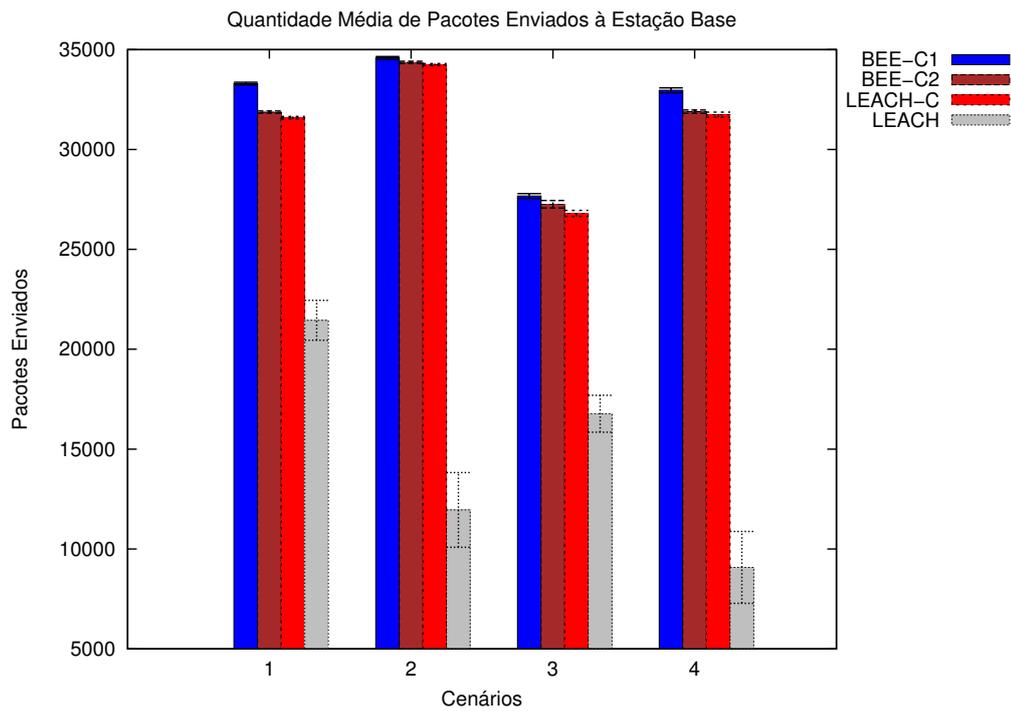


Figura 25: Quantidade média de dados enviados à estação base em cada cenário

A tabela 6 mostra a quantidade média de dados, desvio padrão (σ), coeficiente de variação (CV) e limite inferior (LI) e superior (LS) de confiança (considerando o intervalo de confiança 95%) dos pacotes enviados pelo BEE-C1, BEE-C2, LEACH-C e LEACH nos diferentes cenários.

Podemos observar que as abordagens centralizadas possuem uma baixa variação nos resultados, e os valores do coeficiente de variação são próximos, com isso chegamos à con-

CENÁRIO 1		MÉDIA	σ	C.V.	L.I.	L.S.
	BEE-C1	33299,2	160,8	0,5%	33232,9	33365,6
	BEE-C2	31874,1	135,9	0,4%	31818,0	31930,2
	LEACH-C	31588,0	173,7	0,6%	31516,3	31659,7
	LEACH	21447,4	2421,4	11,3%	20447,9	22446,8
CENÁRIO 2		MÉDIA	σ	C.V.	L.I.	L.S.
	BEE-C1	34584,5	170,5	0,5%	34514,1	34654,9
	BEE-C2	34359,1	138,9	0,4%	34301,8	34416,5
	LEACH-C	34252,3	125,5	0,4%	34200,5	34304,1
	LEACH	11956,2	4536,8	37,9%	10083,4	13828,9
CENÁRIO 3		MÉDIA	σ	C.V.	L.I.	L.S.
	BEE-C1	27672,3	289,0	1,0%	27550,2	27794,3
	BEE-C2	27257,0	452,9	1,7%	27065,7	27448,2
	LEACH-C	26803,4	354,5	1,3%	26653,7	26953,1
	LEACH	16771,7	2185,9	13,0%	15848,7	17694,7
CENÁRIO 4		MÉDIA	σ	C.V.	L.I.	L.S.
	BEE-C1	32966,5	291,3	0,9%	32843,5	33089,6
	BEE-C2	31897,3	188,5	0,6%	31817,7	31976,9
	LEACH-C	31743,2	296,8	0,9%	31614,8	31871,5
	LEACH	9076,9	4254,7	46,9%	7280,3	10873,5

Tabela 6: Informações estatísticas para os dados enviados para a estação base

clusão de que para um mesmo cenário os protocolos, executados várias vezes, terão resultados parecidos, pois a formação dos clusters considera as posições e energia dos nós, e com isso a escolha dos clusters poderão ser semelhantes. O que nos leva a ter maior confiança nos resultados das simulações. Já no LEACH, o coeficiente de variação do envio dos dados é alto, pois a formação dos grupos é praticamente aleatória, podendo ser eficiente em alguns casos e em outros não.

A figura 26 mostra a quantidade de dados enviados por unidade de tempo, para os diferentes protocolos, considerando o cenário 1. Pode-se observar que o protocolo BEE-C1 consegue enviar mais dados que os demais no final da simulação. O BEE-C2 é um pouco mais eficiente que o LEACH-C na entrega dos dados. É importante observar que apesar de enviar uma maior quantidade de dados, a taxa de envio de dados do BEE-C1 é menor que a dos outros protocolos. Isso pode ser observado na figura 26 entre o tempo 50s e 270s, onde o número de pacotes enviados pelo BEE-C1 é menor que o número de pacotes enviados pelos outros. Isto acontece porque o BEE-C1 leva um maior período de tempo para formar os clusters, pois sua função objetivo é muito complexa e demanda um bom tempo para ser calculada.

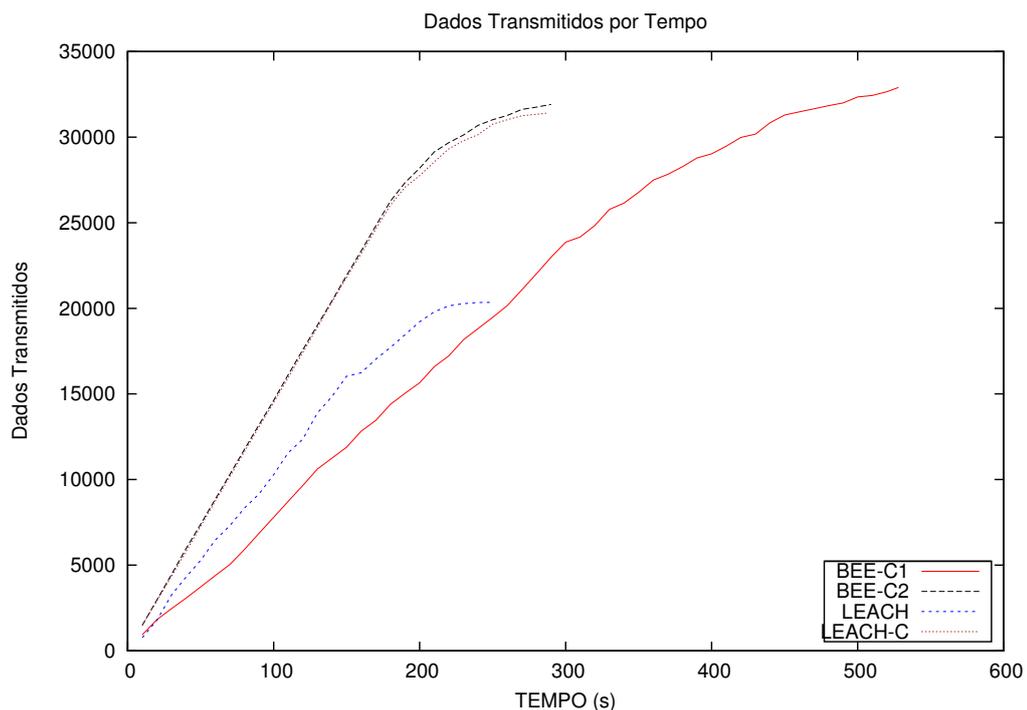


Figura 26: Quantidade de dados enviados à estação base no decorrer do tempo

5.4.2 Tempo de vida total da rede

A figura 27 mostra o tempo médio de vida da rede. Esse tempo é importante, uma vez que quanto maior o tempo de vida da rede, maior é o tempo em que ela estará monitorando o ambiente. O ganho médio do tempo de vida do BEE-C1 em relação ao LEACH-C foi de

55%, já em relação ao LEACH foi superior a 100%. O tempo médio de vida do BEE-C2 foi superior ao do LEACH-C em cerca de 1%. Pode ser observado que enquanto que no cenário 1, o ganho de tempo de vida do BEE-C1 é de cerca de 80% em relação ao LEACH-C, no cenário 3 esse ganho é de apenas 15%. O cenário 3, apresenta a rede com menor densidade, ou seja, os nós são jogados em uma área muito grande e longe da estação base, isso faz com que o consumo de energia da rede aumente, e embora o algoritmo forme clusters eficientemente, o tempo de vida diminui. Em todos os cenários, os resultados do BEE-C2 foram parecidos com os resultados do LEACH-C. O protocolo LEACH obteve o pior tempo de vida em todos os cenários, principalmente nos cenários 2 e 4, onde a rede apresenta o maior número de nós.

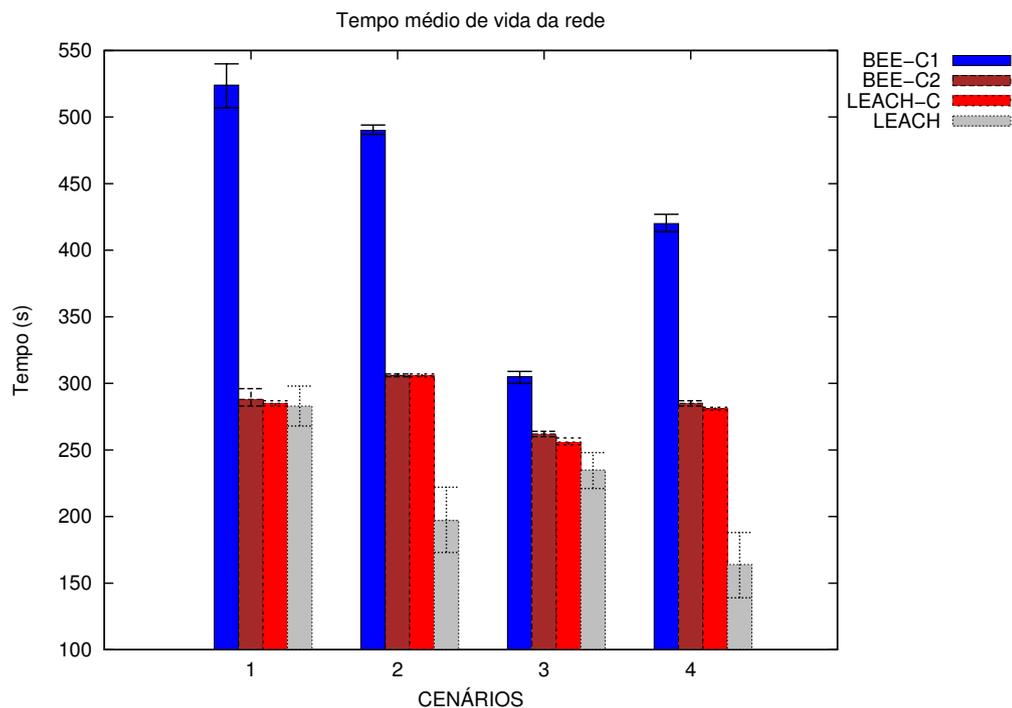


Figura 27: Tempo médio de vida da rede nos diferentes cenários

A tabela 7 mostra a média, desvio padrão (σ), coeficiente de variação (CV) e limite inferior (LI) e superior (LS) de confiança - considerando o intervalo de confiança 95% - do tempo de vida do BEE-C1, BEE-C2, LEACH-C e LEACH nos diferentes cenários.

A figura 28 mostra a quantidade de nós vivos por tempo, considerando o cenário 1. Podemos observar que os nós sensores do BEE-C1 ficam ativos por um maior período de tempo, seguido pelo BEE-C2, que tem o segundo melhor desempenho neste quesito. Entre os protocolos comparados, o LEACH é o que envia menos dados à estação base, mas apesar disso, consegue ter um bom tempo de vida, que quase se equipara com o tempo do LEACH-C e BEE-C2.

CENÁRIO 1		MÉDIA	σ	C.V.	L.I.	L.S.
	BEE-C1	524	39,8	7,6%	287,2	289,9
	BEE-C2	289	3,3	1,1%	287,2	289,9
	LEACH-C	286	4,5	1,6%	283,8	287,5
	LEACH	283	36,0	12,7%	268,5	298,2
CENÁRIO 2		MÉDIA	σ	C.V.	L.I.	L.S.
	BEE-C1	491	8,7	1,8%	487,0	494,3
	BEE-C2	306	2,5	0,8%	305,5	307,5
	LEACH-C	306	2,8	0,9%	305,3	307,6
	LEACH	198	59,5	30,1%	173,4	222,5
CENÁRIO 3		MÉDIA	σ	C.V.	L.I.	L.S.
	BEE-C1	305	9,8	3,2%	300,9	309,2
	BEE-C2	262	4,5	1,7%	260,3	264,1
	LEACH-C	257	6,4	2,5%	254,2	259,6
	LEACH	235	31,8	13,5%	221,7	248,5
CENÁRIO 4		MÉDIA	σ	C.V.	L.I.	L.S.
	BEE-C1	421	15,3	3,6%	414,2	427,2
	BEE-C2	285	3,8	1,3%	283,8	287,0
	LEACH-C	282	2,1	0,7%	280,7	282,5
	LEACH	164	59,1	36,0%	139,0	189,0

Tabela 7: Informações estatísticas para o tempo de vida total da rede

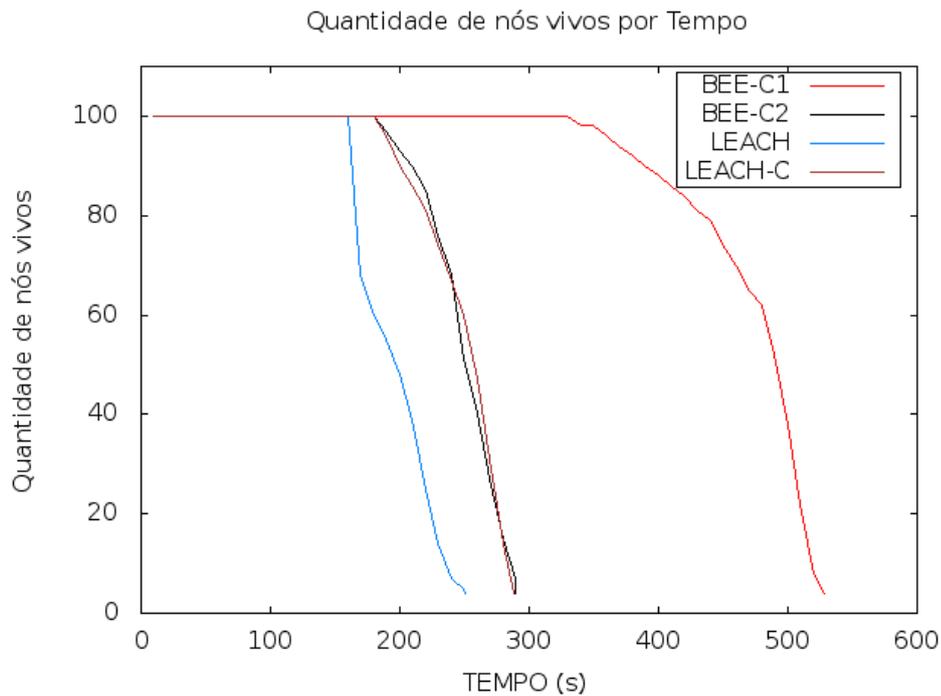


Figura 28: Quantidade de nós vivos no decorrer do tempo

5.4.3 Tempo de cobertura total da rede

A figura 29 mostra o tempo médio de cobertura total da rede, esse valor é calculado obtendo o tempo de vida da rede até a morte do primeiro nó, ou seja, quando partes da área da rede passam a não ser cobertas. Em relação a este quesito, o BEE-C1 foi superior ao LEACH-C em aproximadamente 35% e mais de 100% ao LEACH, enquanto que o BEE-C2 teve um tempo médio de cobertura total da rede em torno de 10% a mais que o LEACH-C e cerca de 50% a mais que o LEACH. É importante observar que nos cenários 3 e 4, como os nós são dispersos em áreas maiores, a morte do primeiro nó ocorre logo nas primeiras rodadas para todos os protocolos comparados.

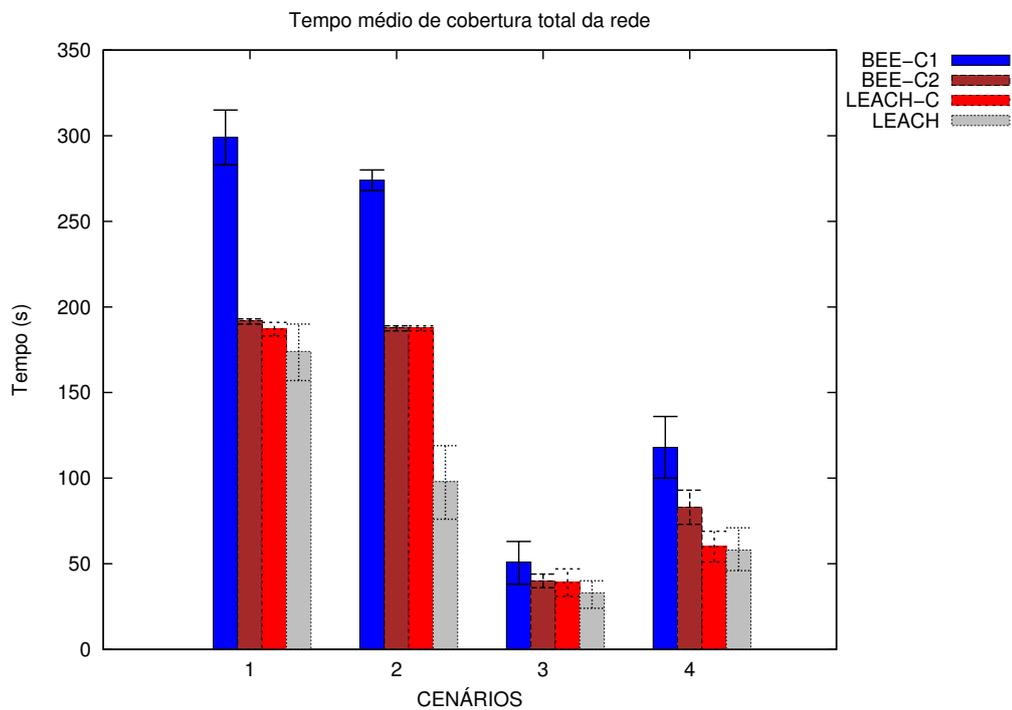


Figura 29: Tempo médio de cobertura total da rede

A tabela 7 mostra a média, desvio padrão (σ), coeficiente de variação (CV) e limite inferior (LI) e superior (LS) de confiança - considerando o intervalo de confiança 95% - do tempo de cobertura total da rede nos diferentes cenários.

Podemos observar na tabela que o coeficiente de variação do LEACH é alto em todos os cenários, ou seja, a confiabilidade da média é menor no LEACH. No entanto, apesar de terem coeficiente de variação pequeno nos cenários 1 e 2, as abordagens centralizadas têm altos valores do coeficiente de variação nos cenários 3 e 4, justamente os cenários que possuem maior área. Isto se justifica pelo fato de que os cluster-heads que se encontram muito longe da estação base tendem a morrer logo, independentemente do algoritmo escolhido. No entanto, apesar dos valores médios baixos e coeficientes de variação altos, os protocolos BEE-C1 e BEE-C2 apresentam os melhores resultados, sendo que o BEE-C2 é o que apresenta os resultados mais

CENÁRIO 1		MÉDIA	σ	C.V.	L.I.	L.S.
	BEE-C1	299	38,3	12,8%	283,4	315,0
	BEE-C2	192	4,1	2,1%	190,3	193,7
	LEACH-C	187	9,8	5,2%	183,2	191,2
	LEACH	174	39,1	22,4%	157,9	190,1
CENÁRIO 2		MÉDIA	σ	C.V.	L.I.	L.S.
	BEE-C1	274	15,3	5,6%	268,1	280,7
	BEE-C2	188	3,7	2,0%	186,9	189,9
	LEACH-C	188	4,1	2,2%	186,3	189,7
	LEACH	98	52,6	53,7%	76,3	119,7
CENÁRIO 3		MÉDIA	σ	C.V.	L.I.	L.S.
	BEE-C1	51	29,2	57,4%	38,5	63,2
	BEE-C2	40	8,6	21,2%	36,8	44,0
	LEACH-C	39	19,3	49,3%	31,0	47,3
	LEACH	33	18,9	58,3%	24,5	40,5
CENÁRIO 4		MÉDIA	σ	C.V.	L.I.	L.S.
	BEE-C1	118	42,7	36,1%	100,3	136,4
	BEE-C2	83	24,1	28,9%	73,2	93,5
	LEACH-C	61	21,1	34,6%	51,8	70,0
	LEACH	59	29,8	50,8%	46,2	71,3

Tabela 8: Informações estatísticas para o tempo de cobertura total da rede

confiáveis, pois apresenta o coeficiente de variação menor em todos os cenários. Este resultado se deve a função objetivo do BEE-C2, que considera não somente a quantidade de energia dos nós, mas também as distâncias para a estação base, no processo de escolha dos cluster-heads.

Na figura 28 podemos notar que o tempo de cobertura do BEE-C1 é maior que o tempo de vida dos outros algoritmos comparados. Além disso, podemos notar que o tempo de cobertura total da rede do BEE-C2 é praticamente igual ao do LEACH-C, que são superiores ao tempo de cobertura do LEACH.

5.4.4 Consumo de energia

Em todas as simulações, é consumida praticamente toda a energia da rede. A figura 30 apresenta o gráfico do consumo de energia dos algoritmos pelo tempo, considerando o cenário 1. Podemos observar na figura 30 que os algoritmos LEACH-C e BEE-C2 apresentam comportamentos semelhante em relação ao consumo de energia, embora o BEE-C2 consiga manter a rede ativa por um período um pouco maior. O BEE-C1 consome menos energia por tempo que os demais. O LEACH é instável, em momentos consome pouca energia e outros acelera no consumo de energia, chegando a ter um tempo de vida médio inferior que os demais.

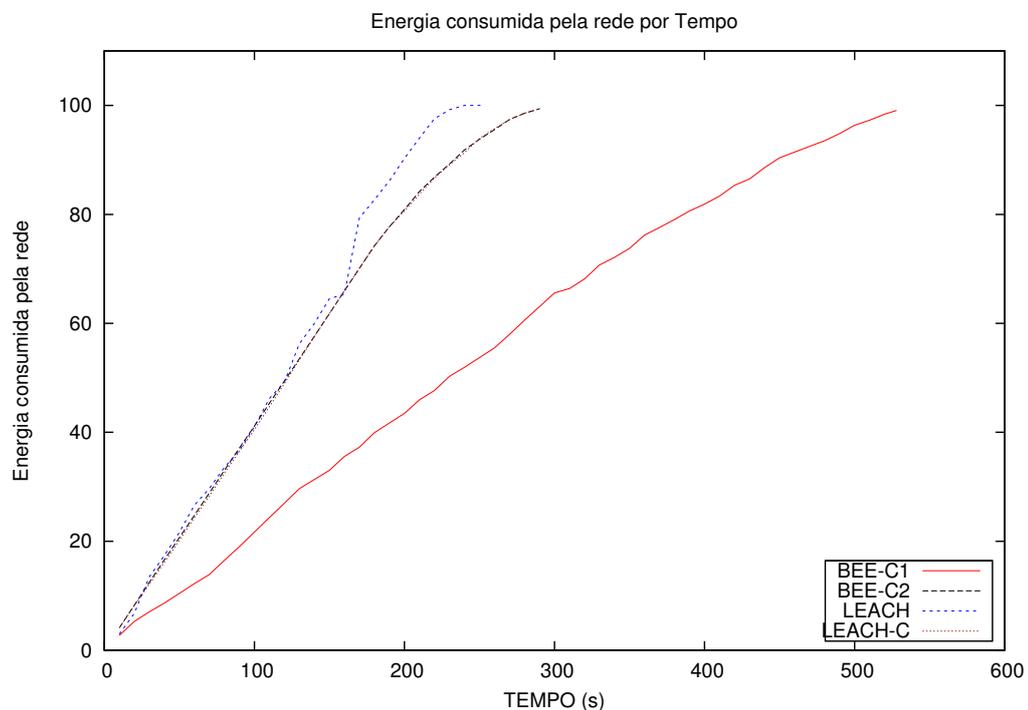


Figura 30: Quantidade de energia consumida pela rede no decorrer do tempo

A figura 31 apresenta o gráfico da entrega de dados de acordo com o consumo de energia, podemos observar que o BEE-C1 consegue enviar uma maior quantidade de dados por energia consumida, sendo seguido pelo BEE-C2, que apresenta um pequeno ganho em relação

ao LEACH-C. Por último, o protocolo LEACH apresenta a menor taxa de envio por energia consumida.

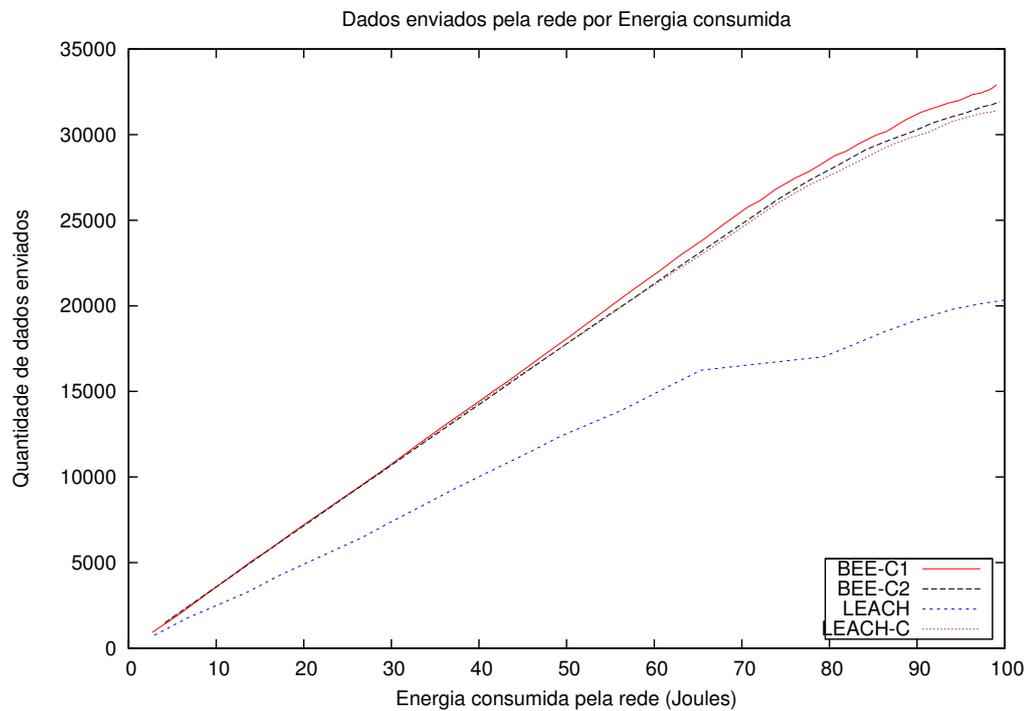


Figura 31: Quantidade de pacotes enviados por energia consumida

5.5 Síntese do Capítulo

Este capítulo apresentou os resultados dos algoritmos BEE-C e comparou-os com os do LEACH-C. Foram considerados dois tipos do BEE-C, o BEE-C1 e o BEE-C2, pois foram desenvolvidas duas funções objetivos. As métricas utilizadas foram o tempo de vida da rede, a quantidade de dados enviados e o consumo de energia total da rede. Em todos os casos, o BEE-C obteve ganhos.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho teve como objetivo principal a elaboração de um algoritmo de roteamento para economia de energia em redes de sensores sem fio, o BEE-C. Para desenvolvê-lo foram estudadas técnicas e protocolos que têm como principal objetivo o prolongamento do tempo de vida da rede através de um bom gerenciamento de energia. Uma técnica muito utilizada pelos protocolos para economia de energia da rede é a técnica de clusterização, que consiste em dividir os nós em grupos, sendo que cada grupo têm um líder, responsável por receber os pacotes dos nós pertencentes ao grupo e enviá-los à estação base. Além de melhorar a rede em consumo de energia, a clusterização também garante uma melhor escalabilidade.

Entre os protocolos estudados que utilizam clusterização destacam-se o LEACH e o LEACH-C, sendo que este último foi utilizado como base para esta proposta. Para clusterizar a rede, o LEACH-C utiliza o algoritmo Têmpera Simulada e utiliza uma função que encontra os cluster-heads minimizando o quadrado de suas distâncias aos nós fontes. Diferente do LEACH-C, este trabalho utilizou um método de clusterização bio-inspirado no comportamento de reprodução de abelhas para agrupar os nós sensores, a fim de otimizar o consumo de energia. A utilização de algoritmos baseados em comportamentos de abelhas vêm sendo utilizada para resolver muitos problemas computacionais, incluindo clusterização de dados, obtendo bons resultados. Além disso, foram desenvolvidas duas funções objetivos que visam aumentar o tempo de vida da rede através do consumo reduzido de energia. A primeira função objetivo foi elaborada com base no modelo de dissipação de energia e a segunda função foi elaborada utilizando a função do LEACH-C, e acrescentando melhorias em relação ao consumo de energia pelos cluster-heads.

Os experimentos mostraram que o algoritmo teve ganhos com as duas opções de funções, na quantidade de pacotes enviados à estação base, no tempo vida da rede, no tempo de cobertura total da rede, em consequência da utilização eficiente de energia pela rede. A primeira função objetivo obteve um ganho superior, no entanto com maior atraso no envio dos pacotes. Enquanto que, embora o algoritmo utilizando a segunda opção de função objetivo não tenha superado, em transmissão de pacotes e tempo de vida, a primeira opção, ele apresenta menos atraso. Portanto, caso a diminuição do atraso seja crucial para a aplicação, é melhor utilizar o algoritmo com a segunda função objetivo.

Para trabalhos futuros são feitas duas sugestões que podem trazer melhoras significativas. A primeira se trata da otimização das funções objetivo. Seria interessante simplificar a primeira função objetivo ou até mesmo juntá-la com a segunda, isto pode trazer bons ganhos para o BEE-C. A segunda sugestão é a utilização de comunicação em múltiplos saltos (multihop) entre os cluster-heads, fazendo isto o consumo de energia pode se tornar mais eficiente ainda, apesar de poder trazer um maior atraso no envio dos pacotes.

REFERÊNCIAS BIBLIOGRÁFICAS

- AGARWAL, P.K.; PROCOPIUC, C.M. Exact and approximation algorithms for clustering. *Algorithmica*, Springer, v. 33, n. 2, p. 201–226, 2002.
- AKKAYA, K.; YOUNIS, M. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, Elsevier, v. 3, n. 3, p. 325–349, 2005.
- AKYILDIZ, IF et al. Wireless sensor networks: a survey. *Computer networks*, Elsevier, v. 38, n. 4, p. 393–422, 2002.
- AL-KARAKI, JN; KAMAL, AE. Routing techniques in wireless sensor networks: a survey. *IEEE wireless communications*, v. 11, n. 6, p. 6–28, 2004.
- BAKER, J.E. Reducing bias and inefficiency in the selection algorithm. In: L. ERLBAUM ASSOCIATES INC. *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*. [S.l.], 1987. p. 14–21.
- BERKHIN, P. A survey of clustering data mining techniques. *Grouping Multidimensional Data*, Springer, p. 25–71, 2006.
- BRAGINSKY, D.; ESTRIN, D. Rumor routing algorithm for sensor networks. In: ACM. *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. [S.l.], 2002. p. 31.
- BROADWELL, P.; POLASTRE, J.; RUBIN, R. GeoMote: Geographic Multicast for Networked Sensors. *Class Project Paper*, Citeseer, v. 21, 2001.
- CULLER, D.; ESTRIN, D.; SRIVASTAVA, M. Guest editors' introduction: Overview of sensor networks. *Computer*, v. 37, n. 8, p. 41–49, 2004.
- DRESSLER, F. *Self-organization in sensor and actor networks*. [S.l.]: Wiley, 2007.
- FATHIAN, M.; AMIRI, B.; MAROOSI, A. Application of honey-bee mating optimization algorithm on clustering. *Applied Mathematics and Computation*, Elsevier, v. 190, n. 2, p. 1502–1513, 2007.
- GARCÍA-HERNANDO, A.B. et al. *Problem Solving for Wireless Sensor Networks*. [S.l.]: Springer, 2008.
- HEINZELMAN, W.B. *Application-specific protocol architectures for wireless networks*. Tese (Doutorado) — Citeseer, 2000.
- HEINZELMAN, WR et al. Energy-efficient communication protocol for wireless microsensor networks. In: *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*. [S.l.: s.n.], 2000. p. 10.
- HEINZELMAN, WB et al. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on wireless communications*, v. 1, n. 4, p. 660–670, 2002.

- HEINZELMAN, W.R.; KULIK, J.; BALAKRISHNAN, H. Adaptive protocols for information dissemination in wireless sensor networks. In: ACM. *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. [S.l.], 1999. p. 174–185.
- INTANAGONWIWAT, C.; GOVINDAN, R.; ESTRIN, D. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In: ACM. *Proceedings of the 6th annual international conference on Mobile computing and networking*. [S.l.], 2000. p. 56–67.
- JAIN, A.K.; DUBES, R.C. *Algorithms for clustering data*. [S.l.: s.n.], 1988.
- JIANG, Q.; MANIVANNAN, D. Routing protocols for sensor networks. In: *First IEEE Consumer Communications and Networking Conference, 2004. CCNC 2004*. [S.l.: s.n.], 2004. p. 93–98.
- KAO, Y.T. et al. A hybridized approach to data clustering. *Expert Systems with Applications*, Elsevier, v. 34, n. 3, p. 1754–1762, 2008.
- KARL, H.; WILLIG, A. *Protocols and architectures for wireless sensor networks*. [S.l.]: Wiley-Interscience, 2007.
- KUHN, F.; WATTENHOFER, R.; ZOLLINGER, A. Worst-case optimal and average-case efficient geometric ad-hoc routing. In: ACM. *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. [S.l.], 2003. p. 267–278.
- LABRADOR, M.A.; WIGHTMAN, P.M. *Topology control in wireless sensor networks*. [S.l.]: Springer, 2009.
- LATIFF, NMA; TSIMENIDIS, CC; SHARIF, BS. Energy-aware clustering for wireless sensor networks using particle swarm optimization. In: *IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007*. [S.l.: s.n.], 2007. p. 1–5.
- LI, Q.; ASLAM, J.; RUS, D. Hierarchical power-aware routing in sensor networks. In: CITESEER. *Proceedings of the DIMACS Workshop on Pervasive Networking*. [S.l.], 2001.
- LINDSEY, S.; RAGHAVENDRA, CS. PEGASIS: Power-efficient gathering in sensor information systems. In: *IEEE Aerospace Conference Proceedings, 2002*. [S.l.: s.n.], 2002. v. 3.
- MAIA, E.H.B.; CÂMARA, D.; LOUREIRO, A.A.F. ICA: Um novo algoritmo de roteamento para redes de sensores. *Simpósio Brasileiro de Redes de Computadores*, 2004.
- MANJESHWAR, A.; AGRAWAL, D.P. TEEN: a routing protocol for enhanced efficiency in wireless sensor networks. In: *1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*. [S.l.: s.n.], 2001.
- MANJESHWAR, A.; AGRAWAL, D.P. APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. In: IEEE COMPUTER SOCIETY. *Proceedings of the 16th International Parallel and Distributed Processing Symposium*. [S.l.], 2002. p. 48.
- MURUGANATHAN, SD et al. A centralized energy-efficient routing protocol for wireless sensor networks. *IEEE Communications Magazine*, v. 43, n. 3, p. S8–13, 2005.

- NICULESCU, D.; NATH, B. Trajectory based forwarding and its applications. In: *ACM. Proceedings of the 9th annual international conference on Mobile computing and networking*. [S.l.], 2003. p. 260–272.
- PHAM, DT et al. Data clustering using the bees algorithm. In: *Proceedings of the 40th CIRP Int. Seminar on Manufacturing Systems*. [S.l.: s.n.], 2007.
- PHAM, DT et al. Data clustering using the bees algorithm. In: *Proceedings of the 40th CIRP Int. Seminar on Manufacturing Systems*. [S.l.: s.n.], 2007.
- SADAGOPAN, N.; KRISHNAMACHARI, B.; HELMY, A. The ACQUIRE mechanism for efficient querying in sensor networks. In: *CITeseer. the Proceedings of the First International Workshop on Sensor Network Protocol and Applications, Anchorage, Alaska*. [S.l.], 2003.
- SCHURGERS, C.; SRIVASTAVA, MB. Energy efficient routing in wireless sensor networks. In: *IEEE Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force*. [S.l.: s.n.], 2001. v. 1.
- SHELOKAR, PS; JAYARAMAN, VK; KULKARNI, BD. An ant colony approach for clustering. *Analytica Chimica Acta*, Elsevier, v. 509, n. 2, p. 187–195, 2004.
- SUBRAMANIAN, L.; KATZ, RH. An architecture for building self-configurable systems. In: *Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC. 2000 First Annual Workshop on*. [S.l.: s.n.], 2000. p. 63–73.
- WAGNER, D.; WATTENHOFER, R. *Algorithms for sensor and ad hoc networks: advanced lectures*. [S.l.]: Springer-Verlag New York Inc, 2007.
- XU, Y.; HEIDEMANN, J.; ESTRIN, D. Geography-informed energy conservation for ad hoc routing. In: *ACM. Proceedings of the 7th annual international conference on Mobile computing and networking*. [S.l.], 2001. p. 84.
- YAO, Y.; GEHRKE, J. The cougar approach to in-network query processing in sensor networks. *SIGMOD record*, Citeseer, v. 31, n. 3, p. 9–18, 2002.
- YE, F. et al. A scalable solution to minimum cost forwarding in large sensor networks. In: *CITeseer. Tenth International Conference on Computer Communications and Networks*. [S.l.], 2001. p. 304–309.
- YU, Y.; GOVINDAN, R.; ESTRIN, D. Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. *UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023*, Citeseer, 2001.
- ZHAO, F.; GUIBAS, L.J. *Wireless sensor networks: an information processing approach*. [S.l.]: Morgan Kaufmann Pub, 2004.