

DISSERTAÇÃO
apresentada à UTFPR
para obtenção do título de

MESTRE EM CIÊNCIAS

por

RODRIGO CUNHA DA SILVA

**OTIMIZAÇÃO MULTIOBJETIVO BASEADA EM
COLÔNIA DE FORMIGAS PARA O ROTEAMENTO E
CONSUMO DE ENERGIA EM REDES DE SENSORES SEM
FIO**

Banca Examinadora:

Presidente:

PROF.^a DR.^a ANELISE MUNARETTO FONSECA **CPGEI-UTFPR**

Examinadores:

PROF. DR. EMILIO CARLOS GOMES WILLE **CPGEI-UTFPR**

PROF. DR. JÚLIO CÉSAR NIEVOLA **PUC-PR**

PROF.^a DR.^a MYRIAM REGATTIERI DELGADO **CPGEI-UTFPR**

Curitiba, Junho de 2008.

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

RODRIGO CUNHA DA SILVA

**OTIMIZAÇÃO MULTIOBJETIVO BASEADA EM COLÔNIA DE FORMIGAS
PARA O ROTEAMENTO E CONSUMO DE ENERGIA EM REDES DE
SENSORES SEM FIO**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná, como requisito parcial para a obtenção do grau de “Mestre em Ciências” – Área de Concentração: Telemática.

Orientadora: Prof.^a Dr.^a Anelise Munaretto Fonseca

Co-Orientadora: Prof.^a Dr.^a Myriam Regattieri
Delgado

Curitiba

2008

Ficha catalográfica elaborada pela Biblioteca da UTFPR – Campus Curitiba

S586o Otimização multiobjetivo baseada em colônia de formigas para o roteamento e consumo de energia em redes de sensores sem fio / Rodrigo Cunha da Silva.
Curitiba. UTFPR, 2008
X, 110 p. : il. ; 30 cm

Orientadora: Prof^ª. Dr^ª. Anelise Munaretto Fonseca

Co-orientadora: Prof^ª. Dr^ª. Myriam Regattieri Delgado

Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial. Curitiba. 2008

Bibliografia: p. 107 – 110

1. Redes de computadores – Protocolos. 2. Sistemas de Computação sem fio. 3. Roteadores (Redes de computadores). 4. Algoritmos. I. Fonseca, Anelise Munaretto, orient. II. Delgado, Myriam Regattieri, co-orient. III. Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial. IV. Título.

CDD: 004.62

AGRADECIMENTOS

A Deus, que sempre me deu força e perseverança para vencer até mesmo os maiores obstáculos da vida.

Aos meus pais, que sempre incentivaram meus estudos e me mostraram que buscar o conhecimento é a melhor forma de alcançar meus objetivos.

Com muito carinho, à minha companheira Lyana, que esteve o tempo todo presente dando o apoio fundamental e me ajudando muito a organizar a vida durante este período tão difícil.

Às professoras Anelise e Myriam, grandes incentivadoras deste trabalho, que acreditaram e investiram desde o início no meu potencial, usando sempre de sua paciência e experiência para resolver as inúmeras dúvidas e discussões necessárias durante esses dois anos.

À minha família, pelos ensinamentos e exemplos de vida e por reforçar minha motivação para este trabalho, ressaltando o grande valor que teria para o meu crescimento futuro.

A todos os amigos e colegas de trabalho, pelas palavras de incentivo, idéias, discussões e críticas, que contribuíram positivamente para os resultados do trabalho.

E por fim, à UTFPR, que me proporcionou a oportunidade de estudar novamente em uma universidade pública de excelente qualidade de infra-estrutura e ensino.

SUMÁRIO

LISTA DE FIGURAS.....	v
LISTA DE TABELAS.....	vii
LISTA DE ABREVIATURAS E SIGLAS.....	viii
RESUMO.....	ix
ABSTRACT.....	x
1 INTRODUÇÃO.....	1
1.1 Objetivos.....	2
1.2 Motivação e Justificativa	2
1.3 Contribuições Esperadas.....	3
1.4 Estrutura da Dissertação.....	3
2 REDES DE SENSORES SEM FIO	5
2.1 Aplicações de RSSF	7
2.2 Hardware.....	8
2.2.1 Processadores.....	10
2.2.2 Rádios	10
2.3 Arquitetura de Comunicação.....	11
2.3.1 Camada Física.....	11
2.3.2 Camada de Enlace.....	12
2.3.3 Camada de Rede	14
2.4 Classificação	16
2.5 Métricas de desempenho.....	19
2.6 Protocolos de Roteamento.....	21
2.6.1 Protocolos de roteamento plano.....	21
2.6.1.1 Protocolo Difusão Direcionada	21
2.6.1.2 Protocolo SPIN.....	23
2.6.1.3 Protocolo MULTI.....	25
2.6.2 Protocolos de roteamento hierárquico	27
2.6.2.1 Protocolo LEACH	28
2.6.2.2 Protocolo SHARP.....	29
2.6.2.3 Protocolo PEGASIS	30
2.6.3 Protocolos de roteamento geográfico	31
2.6.3.1 Protocolo ICA	31
2.6.3.2 Protocolo GPSR	32
2.6.4 Discussão e comparações	33
3 OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS	35
3.1 Descrição do Algoritmo Geral de ACO	36
3.2 Aplicações de ACO em redes de computadores	40
3.2.1 Protocolo <i>AntNet</i>	43
3.2.2 Protocolo <i>AntHocNet</i>	45
3.3 Protocolos para Redes de Sensores sem Fio.....	47
3.3.1 Roteamento direcionado	48
3.3.2 Inundação de agentes em avanço	50
3.3.3 Inundação de agentes de dados	50
3.3.4 Protocolos com ênfase na energia	50
3.4 Discussão	53
4 ANTSENSOR: ROTEAMENTO MULTIOBJETIVO EM RSSF BASEADO EM OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS.....	55
4.1 Introdução	55
4.2 Otimização Multiobjetivo	56

4.2.1	Critério de Dominância	56
4.2.2	Solução Eficiente (ou solução não-dominada).....	57
4.2.3	Método das ponderações	58
4.2.4	Exemplo prático	59
4.3	Descrição do Algoritmo	60
4.4	Abordagem alternativa – Seleção inspirada no método GRASP de busca local.....	67
4.5	Características do <i>AntSensor</i> frente a outros protocolos da literatura	68
4.6	Discussão	69
5	SIMULAÇÕES E RESULTADOS	71
5.1	Simulador.....	71
5.2	Parâmetros de configuração	72
5.3	Experimentos e Resultados	73
5.3.1	Avaliação estática – fonte simples	74
5.3.1.1	Rede de pequeno porte – 10 nós	74
5.3.1.2	Rede de médio porte – 100 nós	77
5.3.2	Avaliação estática por Fronteira de Pareto	85
5.3.3	Avaliação Dinâmica	89
5.3.3.1	Protocolos para Comparação.....	90
5.3.3.2	Métricas de Comparação.....	90
5.3.3.3	Cenários de simulação	91
5.3.3.4	Resultados.....	92
5.4	Análise dos Resultados	97
6	CONCLUSÕES E TRABALHOS FUTUROS	99
6.1	Conclusões	99
6.2	Trabalhos Futuros	99
	ANEXO 1 - ESTATÍSTICAS DOS TESTES DE AVALIAÇÃO DINÂMICA	101
	REFERÊNCIAS BIBLIOGRÁFICAS	107

LISTA DE FIGURAS

Num.	Título	Pág.
1	Dimensões de um nó sensor (FONSECA, 2008).....	5
2	Elementos de uma RSSF (BARROS,2002).....	6
3	Transporte de poluentes na água (GOVINDAN, 2008).....	7
4	Rede de sensores monitorando temperatura (WISL, 2008).....	8
5	Componentes de um nó sensor (AKYILDIZ, SU, SANKARASUBRAMANIAM, et al, 2002).....	9
6	Topologia de uma rede Bluetooth.....	13
7	Roteamento em redes de sensores.....	14
8	Funcionamento do Difusão Direcionada (INTANAGONWIWAT, GOVINDAN e ESTRIN, 2000).....	22
9	Agregação de Dados (NAKAMURA, FIGUEIREDO e LOUREIRO, 2004).....	23
10	Funcionamento do SPIN (HEINZELMAN, KULIK e BALAKRISHNAN, 1999)	25
11	Funcionamento do protocolo SID (FIGUEIREDO, NAKAMURA e LOUREIRO, 2004).....	26
12	Rotatividade de líderes no LEACH (HEINZELMAN, CHANDRAKASAN e BALAKRISHNAN, 2000).....	28
13	Regiões pró-ativas formadas pelo SHARP (RAMASUBRAMANIAN, HAAS e SIRER, 2003).....	29
14	Exemplo de corrente de nós no PEGASIS (LINDSEY e RAGAVENDRA, 2002)	30
15	Roteamento guloso do GPSR – x enviando mensagem para D (KARP e KUNG, 2000).....	32
16	Roteamento perimetral do GPSR – x enviando mensagem para D.....	32
17	Formigas encontrando o melhor caminho ao redor de um obstáculo (PARPINELLI, LOPES e FREITAS, 2001).....	36
18	Grafo com pesos.....	37
19	Rede de nós roteadores.....	41
20	Percurso de um agente reativo em avanço (DI CARO, DUCATELLE e GAMBARDELLA, 2005).....	46
21	Conjunto de soluções eficientes considerando-se duas funções objetivo convexas (FERREIRA, 1998).....	58
22	Conjunto de soluções possíveis para compra de um carro (FACCIOLI, 2007).....	59

23	Cenário de Implementação.....	61
24	Estabelecimento do grafo a partir das mensagens hello.....	62
25	Campos da mensagem agente.....	63
26	Tela do simulador Prowler.....	72
27	Rede com 10 nós.....	74
28	Melhor rota obtida para o Caso 1.....	76
29	Melhor rota obtida para o caso 3.....	77
30	Rede com 100 nós.....	78
31	Melhor rota para o caso 1 - 26 nós, nó mais fraco: 17,93 J.....	79
32	Evolução da solução para o caso 1.....	79
33	Melhor rota para o caso 2 – 18 nós, nó mais fraco: 17,93 J.....	80
34	Evolução da solução para o caso 2.....	81
35	Melhor rota para o caso 4 - 10 nós, nó mais fraco: 12,7 J.....	82
36	Evolução da solução para o caso 4.....	83
37	Melhor rota para o caso 5 - 15 nós, nó mais fraco: 21,02 J.....	84
38	Fronteira de Pareto das soluções médias obtidas.....	87
39	Exemplo de conjunto de rotas obtido em uma simulação.....	88
40	Taxa de Entrega do <i>AntSensor</i>	92
41	Atraso médio do <i>AntSensor</i>	92
42	Tempo de vida até a morte do primeiro nó no <i>AntSensor</i>	93
43	Tempo de vida até que metade dos nós esteja ativa no <i>AntSensor</i>	93
44	Eficiência Energética do <i>AntSensor</i>	94
45	Comparação das taxas de entrega.....	95
46	Comparação dos Atrasos Médios.....	95
47	Comparação do tempo de vida do primeiro nó.....	96
48	Comparação do tempo de vida da metade dos nós.....	96
49	Comparação da eficiência energética.....	97

LISTA DE TABELAS

Num.	Título	Pág.
1	Processadores utilizados em nós sensores.....	10
2	Circuitos de rádio utilizados em nós sensores.....	11
3	Algoritmos baseados em formigas para redes de computadores.....	40
4	Tabela de roteamento para o nó A.....	41
5	Mapeamento entre elementos de rede e elementos do ACO.....	42
6	Tabela de roteamento no AntNet.....	44
7	Tabela de vizinhos de um nó.....	63
8	Comparativo entre protocolos.....	68
9	Parâmetros configuráveis na implementação.....	73
10	Parâmetros utilizados na simulação de fonte simples.....	75
11	Parâmetros de simulação com uma rede de 100 nós.....	77
12	Média da distância (número de nós).....	86
13	Média da energia do nó mais fraco das rotas.....	87
14	Valores numéricos dos parâmetros para análise dinâmica.....	89

LISTA DE ABREVIATURAS E SIGLAS

ACO	<i>Ant Colony Optimization</i>
AD	<i>Adaptive Difusion</i>
ADV	<i>Advertisement</i>
AODV	<i>Ad Hoc On-Demand Distance Vector</i>
CDMA	<i>Code Division Multiple Access</i>
CSMA	<i>Carrier Sense Multiple Access</i>
CSMA/CA	<i>Carrier-Sense Multiple Access with Collision Avoidance</i>
CSMA/CD	<i>Carrier-Sense Multiple Access with Collision Detection</i>
EEABR	<i>Energy-Efficient Ant-Based Routing</i>
FDMA	<i>Frequency Division Multiple Access</i>
GPS	<i>Global Positioning System</i>
ISM	<i>Industrial, Scientific and Medical</i>
IABR	<i>Improved Ant-Based Routing</i>
IP	<i>Internet Protocol</i>
LEACH	<i>Low-Energy Adaptive Clustering Hierarchy</i>
MAC	<i>Medium Access Control</i>
MANET	<i>Mobile Ad Hoc Networks</i>
OFDM	<i>Orthogonal Frequency-Division Multiplexing</i>
QoS	Quality of Service
RF	<i>Radio Frequency</i>
REQ	<i>Request</i>
RSSF	Redes de Sensores sem Fio
SID	Source-Initiated Dissemination
TDMA	<i>Time Division Multiple Access</i>

RESUMO

Redes de Sensores Sem Fio (RSSF) são um tipo especial de rede *Ad Hoc*, sem hierarquia, onde os nós não possuem conhecimento prévio de sua posição em relação aos outros nós da rede e precisam se comunicar entre si para estabelecer a estrutura de comunicação da rede. Os nós dessas redes são bastante simples e possuem recursos limitados.

Um amplo ramo de pesquisa na área de Redes de Sensores sem Fio é o estudo da otimização do consumo de energia, tendo em vista que, em muitos casos, os nós são dispostos em regiões inóspitas, onde é impossível realizar a substituição das baterias. Dessa forma, os nós possuem uma vida útil limitada, conseqüentemente limitando também a vida útil da rede como um todo.

O presente trabalho tem como objetivo apresentar uma abordagem multiobjetivo para otimização do roteamento e consumo de energia para Redes de Sensores sem Fio. Esta abordagem é baseada no algoritmo de Otimização por Colônia de Formigas – ACO – *Ant Colony Optimization*. O método ACO é uma técnica da Computação Natural que procura estudar o comportamento das formigas em busca de alimento para a otimização de processos computacionais. Serão descritos também os principais protocolos da literatura para RSSF, como forma de enfatizar as principais características necessárias para um bom protocolo de roteamento em RSSF e compará-los com o protocolo proposto.

A abordagem proposta foi testada em simulação e se mostrou eficiente de acordo com os critérios de avaliação multiobjetivo, na avaliação com apenas um nó fonte de dados e no teste de otimização multiobjetivo baseado na Fronteira de Pareto. Para comparação com três outros protocolos propostos na literatura de RSSF (EF-Tree, IABR e EEABR), foram realizadas simulações de uma aplicação real baseada em eventos com variação da densidade de nós na área de monitoramento. O protocolo proposto obteve resultados bastante satisfatórios e em alguns casos superiores aos outros em termos de taxa de entrega de mensagens, atraso e tempo de vida da rede.

ABSTRACT

MULTIOBJECTIVE ANT-BASED OPTIMIZATION FOR ROUTING AND ENERGY CONSUMPTION IN WIRELESS SENSOR NETWORKS

Wireless Sensor Networks are a special kind of *Ad Hoc* networks, without infrastructure, where the nodes are normally not aware of other nodes spatial position. Thus, the nodes have to communicate with each other to establish the network communication infrastructure. Each node has a very simple hardware with limited resources.

A wide field of research in Wireless Sensor Networks is the study of the energy consumption optimization, as in many cases, the nodes are deployed in wild regions, where it is impossible to replace their batteries. Thus, each node has a limited lifetime, what consequently limits the overall network lifetime.

The goal of this work is to present a multiobjective optimization strategy for routing and energy optimization in Wireless Sensor Networks. This strategy is based on the ACO – Ant Colony Optimization. This method comes from the Natural Computing and aims to study the behavior of ants searching for food to optimize computational processes. The main methods from the Wireless Sensor Networks literature will also be described, as a way to emphasize the main characteristics needed for a good routing protocol in Wireless Sensor Networks and compare them with the proposed protocol.

The proposed approach has been tested under simulation environment and has been proven to be efficient according to the multiobjective evaluation criteria, in the evaluation with only a single data source node and in the multiobjective optimization test, based in the Pareto Front. For comparison with three other protocols from the literature (EF-Tree, IABR and EEABR), event-based application simulations were performed, with variable network density. The proposed protocol has obtained satisfactory and even better results in some cases in terms of message delivery ratio, network delay and network lifetime

1 INTRODUÇÃO

Redes de Sensores sem Fio (RSSF) são redes *Ad Hoc* especiais, que têm como objetivo monitorar uma região na medição de alguma grandeza física (temperatura, pressão, umidade, etc.) ou sobre a ocorrência de eventos específicos, como incêndios, movimentos de um adversário em campo de batalha, etc. (FIGUEIREDO, NAKAMURA e LOUREIRO, 2004). Uma RSSF é composta de diversos nós simples dispostos, em geral aleatoriamente, sobre uma região a ser monitorada. Uma idéia geral é a de que milhares de nós sensores minúsculos podem ser jogados de uma aeronave e ao atingirem o solo, entrem em operação, passando a monitorar a área de interesse, que pode ser uma floresta, campo, lago, etc. Em geral, uma RSSF possui um ou mais nós sorvedouros de informação que coletam a informação proveniente dos nós sensores e enviam a informação a um sistema de gerência da rede, que tem interesse na informação coletada e processa os dados obtidos na monitoração.

Um dos principais problemas em redes de sensores sem fio é a otimização da utilização das baterias, tendo em vista que os nós podem ser dispostos em regiões inóspitas, onde é impossível a substituição das baterias. Em redes com alta densidade de nós, a substituição de baterias demanda um esforço elevado, o que inviabiliza a renovação dos recursos de energia de cada nó. A rede deve então aproveitar ao máximo a carga disponível na bateria e transmitir a informação utilizando os caminhos mais curtos, além de evitar rotas com nós pobres em energia, de forma a maximizar o tempo de vida útil da rede.

Diversas técnicas podem ser utilizadas para otimização do roteamento em redes. Dentre estas técnicas, destaca-se o método ACO – *Ant Colony Optimization* - (DORIGO, MANIEZZO e COLORNI, 1996) o qual é originário da Computação Natural e busca estudar o comportamento das formigas em busca de alimento para a otimização de processos computacionais. Dorigo, Maniezzo e Colorni (1996) citam que uma das maiores curiosidades no estudo de colônias de formigas é entender como insetos quase cegos podem, com operações bastante simples, ser capazes de encontrar os caminhos mais curtos de ida e volta do seu ninho até uma fonte de alimento.

Neste trabalho, pretende-se utilizar a capacidade de busca de caminhos ótimos inerente ao método ACO no processo de roteamento em redes de sensores sem fio.

1.1 Objetivos

O problema de roteamento em RSSF consiste em encontrar a melhor rota, que é um caminho passando por vários nós, a partir de um determinado nó qualquer até o nó sorvedouro da rede. Este conceito de melhor rota é subjetivo e pode estar sujeito a alguns critérios, sendo caracterizado como um problema multiobjetivo. Os critérios a serem analisados neste trabalho são:

- A rota deve conter o mínimo número de nós possível
- A rota deve ter a maior energia possível no nó mais pobre em energia

Em resumo, uma rota com poucos nós possibilita uma transmissão mais rápida da informação e consome recursos de poucos nós da rede, enquanto uma rota com um alto nível de energia no nó mais pobre da rota tende a levar mais tempo para ser interrompida pela falência de um de seus nós.

O presente trabalho tem como objetivo apresentar uma abordagem multiobjetivo para resolver o problema de roteamento para cada um dos nós de uma RSSF, baseada no algoritmo de Otimização por Colônia de Formigas – ACO – (DORIGO, MANIEZZO e COLORNI, 1996). A proposta é otimizar as rotas de envio de informação dos nós até o sorvedouro considerando os dois objetivos principais mencionados anteriormente: minimização do número de nós nas rotas e maximização da energia no nó mais fraco de cada rota. O objetivo final é a economia de energia na transmissão ponto a ponto e uma conseqüente maximização da vida útil da rede.

Serão descritos também os principais protocolos da literatura para RSSF, como forma de enfatizar as principais características necessárias para um bom protocolo em RSSF e compará-los com o protocolo proposto.

1.2 Motivação e Justificativa

A técnica de ACO foi a abordagem escolhida porque o problema das formigas em busca de alimento é bastante parecido com o problema de escoamento de dados dos nós sensores para o nó sorvedouro em uma RSSF. De fato, a aplicação do ACO para o roteamento na área de Redes de Computadores (DI CARO, DUCATELLE e GAMBARDELLA, 2005) e

em outros problemas que envolvem otimização em grafos é bastante difundida na literatura. Além disso, o método ACO reúne todas as características necessárias para um protocolo de roteamento em RSSF: auto-organização, adaptabilidade, ótima escalabilidade e busca de caminhos ótimos. Outra vantagem é o fato de que o ACO necessita de pouca informação global do sistema, sendo necessário que cada nó conheça apenas informações sobre seus vizinhos mais próximos.

1.3 Contribuições Esperadas

A principal contribuição esperada deste trabalho é uma proposta de melhoria dos protocolos baseados em Colônia de Formigas para RSSF capaz de otimizar o tamanho das rotas e o consumo de energia de forma ponderada através dos pesos de uma função multiobjetivo. Ainda na metodologia, será apresentada uma abordagem alternativa inspirada em métodos de busca local, com o intuito de melhorar as soluções obtidas do método baseado em Colônia de Formigas.

Com relação à pesquisa bibliográfica será apresentado um completo estudo sobre protocolos de roteamento plano para RSSF, servindo como referência para trabalhos futuros na área.

Outras contribuições importantes estão voltadas à avaliação dos resultados obtidos, destacando-se a análise das soluções geradas pelo protocolo proposto utilizando a metodologia da fronteira de pareto e um conjunto de testes para comparação com outros protocolos da literatura, apresentando as métricas usuais utilizadas na área de RSSF.

1.4 Estrutura da Dissertação

Esta dissertação está organizada em seis capítulos. No Capítulo 2 faz-se uma revisão da literatura difundida sobre Redes de Sensores sem Fio, apresentando os principais métodos de roteamento existentes. O método de Otimização por colônia de Formigas (ACO) é discutido no Capítulo 3. O Capítulo 4 descreve o método de otimização de rotas e de energia proposto nesta dissertação e sua implementação. No Capítulo 5 relatam-se os resultados obtidos. E, finalmente, o Capítulo 6 apresenta a discussão dos resultados, as conclusões do trabalho e as propostas de trabalhos futuros.

2 REDES DE SENSORES SEM FIO

Redes de Sensores sem Fio (RSSF) são redes *Ad Hoc* especiais, que têm como objetivo monitorar uma região na medição de grandezas físicas (FIGUEIREDO, NAKAMURA e LOUREIRO, 2004). Uma RSSF é composta de diversos nós simples dispostos, em geral aleatoriamente, sobre uma região a ser monitorada. Um dado nó sensor da rede efetua a medição da grandeza física na sua posição e envia essa informação aos outros nós através de comunicação via rádio.

Cada nó sensor é em geral descartável e tem baixo custo, de forma que seus recursos são limitados. Um nó sensor pode ter, por exemplo, os seguintes componentes:

- Sensor
- Memória de 128 kB
- Processador de 8MHz
- Rádio 802.11
- Bateria

Os nós sensores na maioria dos casos possuem dimensões bastante reduzidas. No futuro, espera-se que existam até mesmo redes com nós microscópicos, monitorando o corpo humano ou microorganismos. As dimensões de um nó sensor existente atualmente são ilustradas na Figura 1.

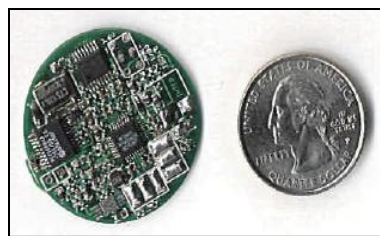


Figura 1: Dimensões de um nó sensor (FONSECA, 2008)

Em geral, uma RSSF possui um ou mais nós sorvedouros de informação que coletam a informação proveniente dos nós sensores e enviam a informação a um sistema de gerência da rede, que tem interesse na informação coletada e processa os dados obtidos na monitoração. O

roteamento dos dados, na maioria dos casos, é feito através de múltiplos saltos, ou seja, o nó que deseja enviar alguma informação ao sorvedouro envia uma mensagem a outro nó que esteja mais próximo do mesmo. Este repassa a mensagem para outro ainda mais próximo do sorvedouro e assim por diante, até que a mensagem chegue ao destino. A Figura 2 ilustra a disposição dos elementos em uma RSSF típica:

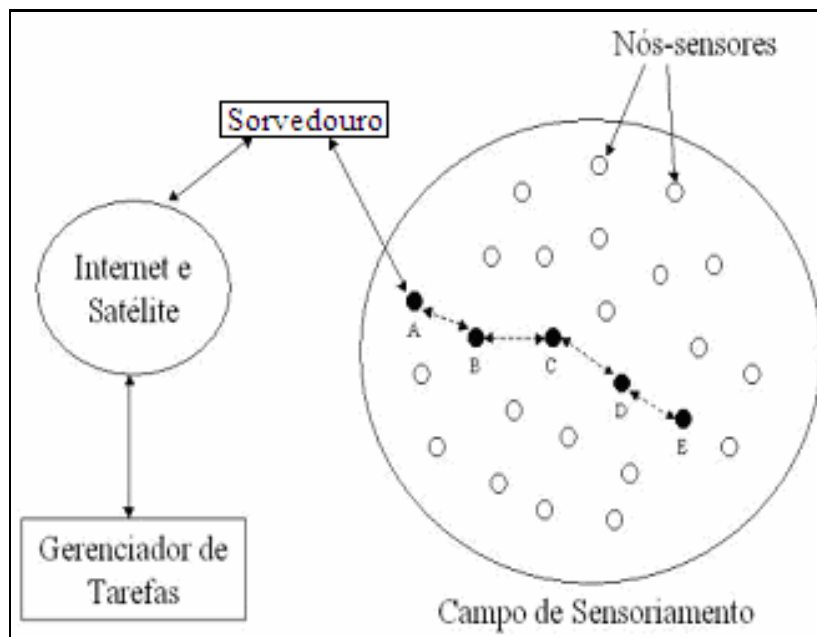


Figura 2: Elementos de uma RSSF (BARROS,2002)

A topologia em uma RSSF pode ser fortemente variável com o tempo. Inicialmente, os nós são dispostos aleatoriamente sobre o campo de monitoramento. A partir daí, devem se comunicar entre si para estabelecer a topologia de roteamento da rede e as rotas de escoamento dos dados. Com o tempo, alguns nós podem ser movidos por forças naturais: chuva, vento, etc. Além disso, a qualquer momento, nós podem deixar de operar devido ao fim das baterias. Em determinadas RSSF os nós podem até mesmo ser móveis, se equipados com mecanismos de locomoção. Dessa forma, os protocolos de roteamento devem ser tolerantes a falhas e adaptativos com relação a mudanças de topologia da rede.

Um dos principais problemas em redes de sensores sem fio é a otimização da utilização das baterias, tendo em vista que os nós podem ser dispostos em regiões inóspitas, onde é impossível a substituição dessas baterias. Em redes com alta densidade de nós, a substituição de baterias demanda um esforço elevado, o que inviabiliza a renovação dos recursos de energia de cada nó. A rede deve então aproveitar ao máximo a carga disponível na

bateria e transmitir a informação utilizando os caminhos mais curtos, além de evitar rotas com nós pobres em energia, de forma a maximizar o tempo de vida útil da rede.

2.1 Aplicações de RSSF

Dentre as aplicações em redes de sensores sem fio, destacam-se (GOVINDAN, 2008):

a) Resposta sísmica de estruturas – Sensores podem ser distribuídos em pontos estratégicos de um prédio ou construção para a observação das movimentações da estrutura. O objetivo é verificar a resposta da estrutura a movimentações no solo. A análise do histórico de tensões na estrutura também pode prevenir danos.

b) Transporte de poluentes – Nós com sensores sensíveis a poluentes podem ser espalhados no solo ou na água para monitorar a concentração de poluentes. Existem técnicas em que uma grande quantidade de sensores capazes de localizar com precisão sua posição espacial pode ser utilizada para modelar o fluxo de transporte dos poluentes, conforme mostrado na Figura 3. No caso de ecossistemas marinhos, as redes de sensores podem ainda ser aplicadas para monitorar a concentração de algas marinhas com base nos resíduos produzidos por essas algas.

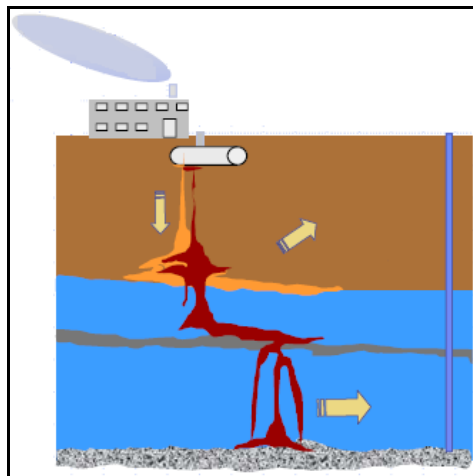


Figura 3: Transporte de poluentes na água (GOVINDAN, 2008)

c) Medição de Temperatura – Uma rede de sensores pode ser aplicada para monitorar gradientes de temperatura em uma região. Cada nó reporta o valor de temperatura medido em

sua localização e sua posição espacial. A aplicação que recebe os dados da rede é capaz de, com essas informações, mostrar graficamente o comportamento da temperatura na região de interesse, como ilustra a Figura 4.

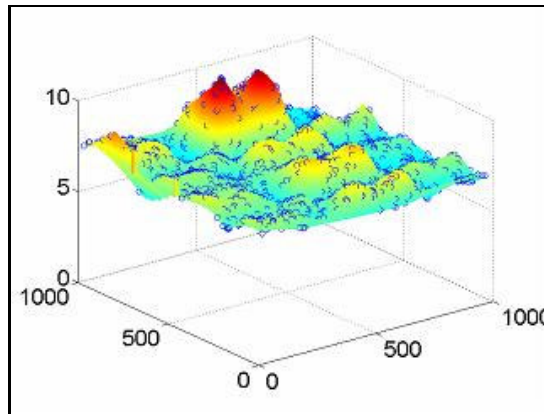


Figura 4: Rede de sensores monitorando temperatura (WISL, 2008)

d) Localização de soldados – Sensores instalados nos uniformes dos soldados permitem a uma central monitorar a posição e os movimentos de cada soldado. Este monitoramento pode até mesmo ser visualizado por um comandante em campo de batalha, através de um visor.

e) Detecção de incêndios - A distribuição de sensores em uma floresta permite que incêndios sejam detectados em pouco tempo e sejam localizados imediatamente e com precisão, o que permite o controle dos incêndios rapidamente, antes que se espalhem por uma área muito grande.

Conforme mostrado anteriormente, as aplicações para RSSF podem ser bastante diversas. A aplicação a que uma RSSF é destinada pode influenciar bastante nas características e no padrão de tráfego da rede e, conseqüentemente, no projeto do protocolo de roteamento a ser usado na mesma.

2.2 Hardware

Um nó sensor é formado por quatro componentes principais, conforme mostrado na Figura 5:

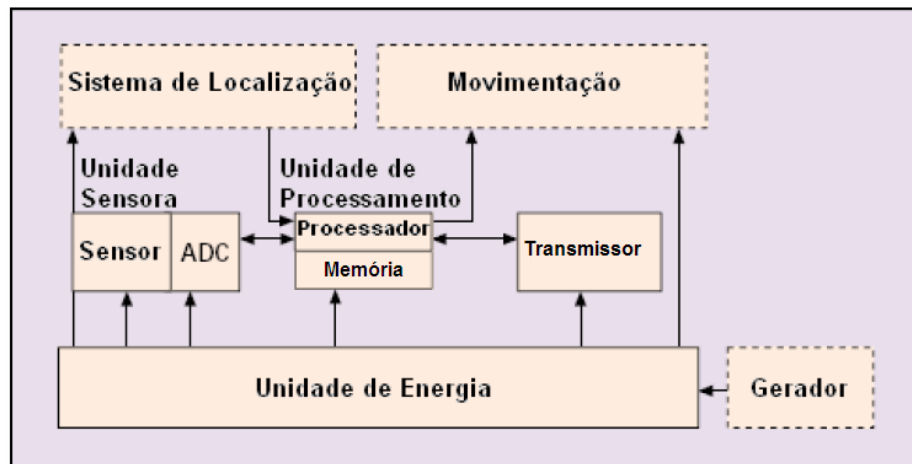


Figura 5: Componentes de um nó sensor (AKYILDIZ, SU, SANKARASUBRAMANIAM, et al, 2002)

a) Unidade sensora – Formada por um sensor e um conversor analógico-digital. Produz um sinal proporcional ao fenômeno observado, que serve de entrada para a unidade de processamento.

b) Unidade de processamento – Formada por um processador de baixo consumo e uma pequena unidade de armazenamento. Responsável por coordenar os procedimentos que fazem os nós sensores colaborarem entre si para executar as tarefas de interesse.

c) Transmissor – Conecta o nó sensor com a rede, através de rádio, infravermelho ou outros tipos de sinais físicos. No caso de infravermelho ou comunicação óptica, é necessário que haja linha de visada entre os nós, o que na maioria dos casos é impraticável.

d) Unidade de energia – Em geral são apenas baterias. A capacidade é inferior a 0,5 Ah – 1,2 V. Podem ser incrementadas usando painéis solares ou outras formas de geração de energia.

Opcionalmente, podem existir dois outros componentes auxiliares, de acordo com o tipo de aplicação:

e) Sistema de localização - É responsável por determinar a posição espacial do nó sensor. Algumas aplicações, como a medição de temperatura ou transporte de poluentes necessitam que o sensor tenha capacidade de detectar sua posição espacial para informá-la ao sistema de aplicação. A tecnologia mais utilizada nestes casos é o GPS (*global positioning system*).

f) Sistema de movimentação – Em algumas aplicações pode ser necessário que o nó sensor seja capaz de se movimentar. Neste caso é necessário prover o sensor de motores ou

mecanismos similares. Estes mecanismos são pouco utilizados em redes onde não é possível a reposição de baterias do nó sensor.

A seguir são mostrados alguns exemplos de componentes que são utilizados na indústria para a fabricação de nós sensores (GOVINDAN, 2008):

2.2.1 Processadores

A Tabela 1 mostra alguns processadores utilizados em nós sensores existentes. É possível verificar aqui que o processador Atmel, que é um microcontrolador, é o mais simples de todos, inclusive com o mais baixo consumo entre os três. Este processador é usado na família Mica2, fabricada pela Crossbow Technologies (CROSSBOW, 2008), uma das mais famosas na comunidade de pesquisa em RSSF. Os outros dois processadores são processadores mais poderosos, para aplicações com mais abundância de recursos energéticos.

Tabela 1: Processadores utilizados em nós sensores (GOVINDAN, 2008)

Processador	Frequência	Memória	Consumo
Atmel Atmega 128L	8 Mhz/16 MHz	128 kB – programas 4 kB – SRAM 4 kB – EPROM	Modo ativo: 16,5 mW Modo <i>sleep</i> : < 60 μ W
Intel Strong ARM SA1100	206 MHz	16 kB – programa 8 kB - dados	Modo ativo: 800 mW Modo <i>idle</i> : 210 mW Modo <i>sleep</i> : 50 μ W
Intel Xscale PXA-250	200/300/400 MHz	32 kB – programa 32 kB - dados	Modo ativo: 400 mW Modo <i>idle</i> : 160 mW Modo <i>sleep</i> : 50 μ W

2.2.2 Rádios

Os rádios utilizados em redes de sensores devem ser de baixo consumo e curto alcance. Na escolha de um rádio para um nó sensor, algumas características devem ser levadas em consideração:

- a) Frequência
- b) Esquema de modulação
- c) Taxa de dados

d) Diversidade de frequências

Em termos de energia, como os nós em geral estão separados de distâncias pequenas, a potência gasta na transmissão é em geral da mesma ordem da potência gasta na recepção. A Tabela 2 mostra alguns exemplos de rádios utilizados em nós sensores e suas características.

Tabela 2: Circuitos de rádio utilizados em nós sensores (GOVINDAN, 2008)

Rádio	Modulação	Taxa de dados	Frequência	Consumo
RFM TR1000	OOK/ASK	Até 115,2 kbps	433/916 Mhz	Rx: 3,8 mA Tx: 12 mA
Chipcon CC1000	FSK	76,8 KBaud	300 – 1000 MHz	26,7 mA
Bluetooth	GFSK	6-12 Kbps	2,4 GHz	Rx: 90 mW Tx: 150 mW

2.3 Arquitetura de Comunicação

2.3.1 Camada Física

A camada física é responsável pela seleção de frequências, geração de onda portadora, detecção de sinal, modulação e codificação de dados. Em Redes de Sensores, a faixa de frequência sugerida é ISM (*industrial, scientific and medical*), utilizada para aplicações industriais, científicas e médicas.

Sabe-se que a comunicação sem fio a longas distâncias pode ser bastante dispendiosa em termos de energia e complexidade de implementação. No projeto de redes de sensores, a otimização do consumo de energia assume um papel fundamental. Em geral, para a transmissão de um sinal, a mínima potência de saída do transmissor a uma distância d , é proporcional a d^n , onde $2 \leq n \leq 4$ (AKYILDIZ, SU, SANKARASUBRAMANIAM, *et al*, 2002). O expoente n é próximo de 4 para antenas baixas e canais próximos ao solo, como é o caso típico em redes de sensores. Isto pode ser atribuído ao cancelamento parcial devido a raios refletidos no solo (modelo 2 raios). Alguns estudos citados em (AKYILDIZ, SU, SANKARASUBRAMANIAM, *et al*, 2002) mostram que a potência começa a cair com expoentes mais elevados em curtas distâncias no caso de antenas baixas.

Para resolver esses problemas, é importante que o projeto utilize técnicas especiais. A comunicação em múltiplos saltos em redes de sensores pode ser bastante eficiente para

resolver problemas como nós escondidos e perda de percurso (*path loss*) (AKYILDIZ, SU, SANKARASUBRAMANIAM, *et al*, 2002). O Roteamento em múltiplos saltos é uma das técnicas implementadas na abordagem proposta neste trabalho.

Valores típicos de taxa de dados entre sensores são em geral menores que 1Mbps e os pacotes são relativamente pequenos, menores que 5 kbits. As distâncias entre os nós são normalmente pequenas (menos de 10 metros). Em contrapartida, a distância entre um nó sensor e o nó sorvedouro pode ser da ordem de dezenas de quilômetros. Como a comunicação a longas distâncias é dispendiosa em termos de energia, a comunicação com o nó sorvedouro deve ser projetada de modo a ocorrer apenas quando necessário.

2.3.2 Camada de Enlace

A camada de enlace em redes de sensores tem um objetivo principal: dividir corretamente os recursos de comunicação entre os diversos nós sensores. Entre os serviços executados pela camada de enlace estão a multiplexação dos fluxos de dados, detecção dos quadros, acesso ao meio e controle de erro (TANENBAUM, 2003).

Nas redes sem fio tradicionais, a camada de enlace deve controlar o acesso ao meio físico para evitar colisões e interferência. Devido às características do canal de rádio, a transmissão de sinal por um nó pode causar interferência e ruído observado por todos os nós de sua vizinhança. Dessa forma, a camada de enlace deve prover algoritmos distribuídos para dividir o canal de rádio entre os diversos nós e evitar que estes transmitam utilizando a mesma banda. Apesar de os nós terem bateria limitada, é possível que o usuário faça a reposição (TANENBAUM, 2003).

Para isso, existem diversos protocolos. O mais comum é o CSMA (*Carrier Sense Multiple Access*), também utilizado no padrão Ethernet, com um mecanismo que evita colisões. No caso do Ethernet, é comum utilizar-se o padrão CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*), que evita colisões escutando o meio antes e durante a transmissão. Porém, como isto não é possível nas redes sem fio, utiliza-se o CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*). (TANENBAUM, 2003).

Bluetooth e *mobile ad hoc network* (MANET) (IETF, 2008) são os tipos de redes mais próximos das redes de sensores (AKYILDIZ, SU, SANKARASUBRAMANIAM, *et al*,

2002). *Bluetooth* é um sistema sem infra-estrutura para comunicação a curtas distâncias, que tem o objetivo de substituir os cabos em terminais de usuário por canais RF (TANENBAUM, 2003). A topologia do *Bluetooth* é uma rede em estrela, onde um nó mestre pode ter até 7 nós escravos conectados a si para formar uma picorede (TANENBAUM, 2003), conforme ilustra a Figura 6. Cada picorede usa um agendamento TDMA (Multiplexação por Divisão de Tempo) próprio. A potência é tipicamente próxima de 20 dBm e o alcance da ordem de dezenas de metros.



Figura 6: Topologia de uma rede *Bluetooth*

Em contraste a isso, em redes de sensores o número de nós pode ser muito maior, a potência de transmissão perto de 0 dBm e o alcance muito menor do que nas redes *Bluetooth*. As mudanças de topologia são mais frequentes nas redes de sensores, podendo ser atribuídas a mobilidade dos nós ou falhas. A taxa de mobilidade normalmente é menor do que nas MANET. O controle de colisão é um fator bastante complexo em redes de sensores, no caso de redes com altas densidades de nós.

Uma técnica especial utilizada pela camada de enlace em RSSF é a desativação coordenada dos nós que não estão ativos (YE, HEIDERMAN e ESTRIN, 2002). Os nós alternam períodos de escuta (*listen*) e inatividade (*sleep*). Isto permite que a rede continue em operação por um longo período, por economizar energia durante os períodos de inatividade dos nós. Essa técnica é utilizada em redes de sensores e não será abordada no protocolo proposto por este trabalho.

2.3.3 Camada de Rede

A camada de rede é responsável pelo roteamento de dados entre os sensores. Os protocolos de roteamento utilizados devem suportar a comunicação em múltiplos saltos, e devem buscar sempre o uso mais eficiente possível da energia do sensor (FONSECA, 2008).

A seleção da rota mais eficiente entre um sensor que deseja transmitir uma mensagem e o nó sorvedouro pode levar em conta diversos critérios. Tomando a rede da Figura 7 como exemplo, se o sensor “A” detectou um evento e deseja enviar esta informação até o nó sorvedouro, este sensor pode utilizar diversas rotas.

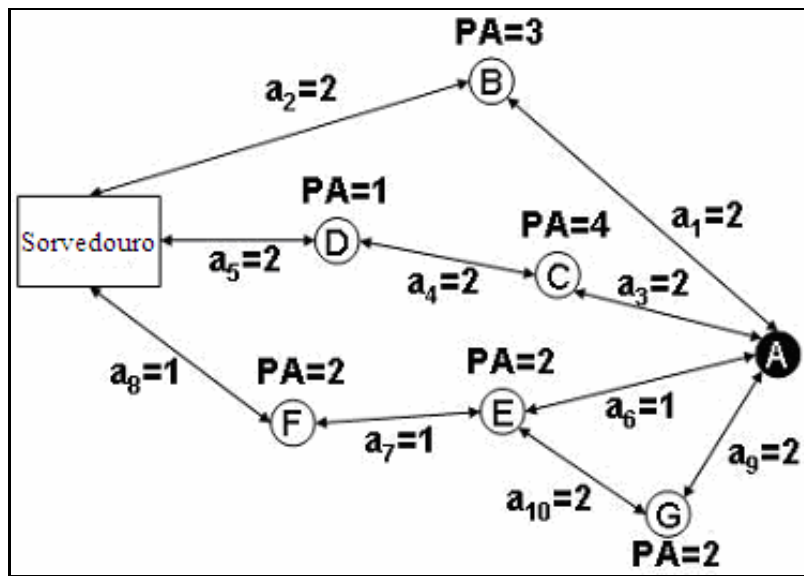


Figura 7: Roteamento em redes de sensores

Na figura acima, PA representa a energia disponível em cada sensor, e “ a_i ” é a energia necessária para a transmissão de um pacote passando por aquele canal de rádio. As possíveis rotas entre A e sorvedouro são:

- Rota 1: A-B-Sorvedouro, $PA_{total} = 3$, $a_{total} = 4$.
- Rota 2: A-C-D-Sorvedouro, $PA_{total} = 5$, $a_{total} = 6$.
- Rota 3: A-E-F-Sorvedouro, $PA_{total} = 4$, $a_{total} = 3$.
- Rota 4: A-G-E-F-Sorvedouro, $PA_{total} = 6$, $a_{total} = 6$.

A escolha da rota que será utilizada para transmissão neste exemplo pode utilizar um dos seguintes critérios:

a) Maior energia disponível:

Escolhe-se a rota cuja energia total disponível, ou seja, a soma da energia disponível em todos os sensores da rota, seja a maior possível. Seguindo este critério, a rota escolhida no exemplo acima seria a rota 4. Porém, observa-se que esta rota inclui todos os sensores da rota 3 e mais um, ou seja, esta não é a rota mais eficiente. Por isso, não se deve considerar rotas que sejam extensões de outras rotas. Neste caso, eliminando-se a rota 4, a rota 2 será a escolhida.

b) Menor consumo de energia:

Segundo este critério, a melhor rota é aquela na qual o consumo de energia na transmissão de pacotes entre sensores é o menor. Assim, a rota escolhida é a que tem o menor valor na soma da energia consumida em cada trecho até o nó sorvedouro. No exemplo acima, a rota 3 seria escolhida.

c) Menor número de saltos:

A escolha da rota é de acordo com o número de sensores pelo qual o pacote tem que passar até chegar ao nó *sorvedouro*. A rota com o menor número de sensores é escolhida. Assim, a rota 1 seria escolhida no exemplo acima.

d) Maior PA mínima:

Este critério considera o PA mínimo de cada rota, ou seja, o sensor que tem menos energia em cada uma das rotas. A rota que tiver o maior valor dentre estes é escolhida. No exemplo acima, o menor PA da rota 1 é 3 (sensor B); o da rota 2 é 1 (sensor D); o da rota 3 é 2 (sensores E e F); e o da rota 4 é 2 (sensores E, F e G). Assim, a rota escolhida é a rota 1. Este critério tem como objetivo evitar utilizar na transmissão dos pacotes os sensores com menos energia.

É importante ressaltar aqui que os critérios que serão utilizados para medir a qualidade das rotas no protocolo que será proposto neste trabalho são os critérios c) e d), ou seja, menor número de saltos e maior energia mínima, com o objetivo de obter rotas curtas, consumindo energia de poucos nós e que não estejam próximas de serem interrompidas pelo fim das baterias de um de seus nós. O critério que envolve a energia é utilizado pelo fato de que em

RSSF a principal preocupação é a economia de energia. Já o critério de menor número de saltos se justifica porque além de ser uma métrica simples de ser calculada, procura minimizar o consumo de recursos da rede, utilizando poucos nós para envio das mensagens.

2.4 Classificação

De acordo com (LUZ, 2004) as RSSF podem ser classificadas segundo diferentes critérios:

a) Classificação segundo a composição:

- a.1) Homogênea: todos os nós possuem as mesmas capacidades físicas (*hardware*).
- a.2) Heterogênea: compostas por nós com diferentes capacidades físicas.

b) Classificação segundo a organização:

- b.1) Hierárquica: os nós estão organizados em aglomerados (*clusters*). Cada grupo de nós possui um líder (*cluster-head*), eleito pelos demais nós do grupo.
- b.2) Plana: não há organização em aglomerados. Todos os nós possuem a mesma hierarquia e mesmas funções.

c) Classificação segundo a mobilidade:

- c.1) Estacionária: todos os nós permanecem no mesmo local no espaço onde foram depositados durante toda a vida da rede.
- c.2) Móvel: os nós sensores podem mudar de posição ao longo do tempo.

d) Classificação segundo a densidade:

- d.1) Balanceada: concentração e distribuição dos nós em uma região (unidade de área) ideal para o objetivo da rede.
- d.2) Densa: alta concentração de nós por região.
- d.3) Esparsa: baixa concentração de nós por região.

e) Classificação segundo a distribuição:

- e.1) Irregular: distribuição dos nós não uniforme na área de monitoramento.

e.2) Regular: distribuição dos nós uniforme na área de monitoramento.

f) Classificação segundo a regularidade da coleta de dados:

f.1) Periódica: os nós sensores coletam dados sobre o fenômeno de interesse em intervalos regulares.

f.2) Reativa: os nós sensores coletam dados quando ocorrem eventos de interesse ou quanto solicitado pelo operador.

g) Classificação quanto à disseminação de dados na rede:

g.1) Programada: os nós disseminam os dados em intervalos regulares.

g.2) Sob demanda: os nós disseminam os dados em resposta a uma consulta do observador ou à ocorrência de eventos.

h) Classificação quanto ao tipo de conexão:

h.1) Simétrica: todas as conexões de rádio entre os nós sensores, com exceção do nó sorvedouro, têm o mesmo alcance.

h.2) Assimétrica: as conexões entre os nós comuns podem ter alcances diferentes.

i) Classificação quanto ao tipo de transmissor:

i.1) *Simplex*: os nós sensores possuem um rádio que permite apenas transmissão de informação.

i.2) *Half-Duplex*: os nós sensores possuem transceptor que permite apenas transmitir ou receber dados em um determinado instante.

i.3) *Full-Duplex*: os nós sensores possuem transceptor que permite transmitir e receber dados simultaneamente.

j) Classificação de acordo com a alocação de canal:

j.1) Estática: neste tipo de rede, se existem “n” nós, os recursos de transmissão são divididos em “n” partes iguais utilizando uma das seguintes técnicas de divisão: na frequência (FDMA), no tempo (TDMA), no código (CDMA), no espaço (SDMA) ou ortogonal (OFDM) (RAPPAPORT, 2001). A cada nó é atribuída uma parte privada do canal de comunicação, minimizando a interferência.

j.2) Dinâmica: não existe atribuição fixa de recursos de transmissão. Os nós disputam o canal para comunicação de dados.

k) Classificação quanto ao fluxo de informação:

k.1) Inundação (*flooding*): neste tipo de rede os nós sensores fazem *broadcast* de suas informações para os seus vizinhos. Estes por sua vez, também fazem *broadcast* dos dados recebidos para outros até alcançarem o nó de destino. Esta abordagem causa um alto *overhead*, mas está imune às mudanças dinâmicas de topologia e alguns ataques de impedimento de serviço (DoS – *Denial of service*).

k.2) *Multicast*: neste tipo de rede os nós formam grupos onde cada nó envia dados para os demais membros de seu grupo.

k.3) *Unicast*: neste tipo de rede os nós sensores podem se comunicar enviando mensagens para apenas um nó de cada vez e atingir o nó sorvedouro através de protocolos de roteamento em múltiplos saltos.

k.4) *Gossiping*: neste tipo de rede os nós selecionam randomicamente o nó para o qual será enviado cada pacote de dados.

k.5) *Bargaining*: neste tipo de rede os nós enviam os dados para um determinado destino somente se houver interesse do receptor, isto é, existe um processo de negociação anterior ao envio de dados.

l) Classificação segundo a cooperação entre os nós:

l.1) Infra-estrutura: os nós sensores executam procedimentos relacionados à infra-estrutura da rede, como algoritmos de controle de acesso ao meio, roteamento, eleição de líderes, descoberta de localização e criptografia.

l.2) Localizada: os nós executam além dos procedimentos de infra-estrutura algum tipo de processamento localizado, como por exemplo, a tradução dos dados coletados pelos sensores com base na calibração.

l.3) Correlação: os nós estão envolvidos em procedimentos de correlação de dados, como fusão, supressão seletiva, contagem, compressão e agregação, ou seja, cada nó coleta a informação recebida dos vizinhos e faz um pré-processamento antes de reenviá-la para próximos nós.

Considerando os critérios discutidos anteriormente, a abordagem proposta neste trabalho será avaliada em uma rede com as seguintes características: a) Homogênea; b) Plana; c) Estacionária; d) Densa; e) Regular; f) Reativa; g) Sob demanda; h) Simétrica; i) *Half-duplex*; j) Dinâmica; k) *Unicast*; l) Infra-estrutura.

2.5 Métricas de desempenho

De acordo com (LUZ, 2004), as principais métricas de desempenho para avaliar protocolos de roteamento em RSSF são: eficiência de uso da energia, vida útil do sistema, latência, precisão, tolerância a falhas, escalabilidade e exposição dos sensores.

a) Eficiência de energia e vida útil do sistema:

Esta é uma das métricas utilizadas para avaliar o protocolo proposto neste trabalho. Existem várias métricas para mensurar o tempo de vida útil de uma rede, dentre elas:

- Tempo de nós ativos: por exemplo, o tempo desde o início da operação da rede até o momento em que a metade dos nós esteja ativa.

- Tempo de envio da informação para a aplicação: por exemplo, o tempo até o momento em que a rede para de fornecer informação sobre um determinado fenômeno para o nó sorvedouro.

b) Latência e precisão:

A latência é um espaço de tempo no qual o observador está interessado em estudar um determinado tipo de fenômeno. Portanto, a semântica para a métrica relativa a latência depende do fenômeno e da aplicação.

A precisão é um dos objetivos do observador do fenômeno e sua métrica depende da aplicação.

Um protocolo de roteamento ótimo deve garantir que a aplicação obtenha os dados sensoriados com a precisão e latência desejadas e garantindo um consumo mínimo de energia. Por exemplo, alguns dos nós sensores que fornecem dados podem ser requisitados com mais

freqüência ou o roteamento da disseminação de dados podem ocorrer mais freqüentemente nos mesmos nós sensores quando a origem de dados é a mesma.

Para avaliar o protocolo proposto neste trabalho, será utilizada como métrica de latência o tempo médio que um pacote leva para se enviado do nó fonte de um determinado evento até o nó sorvedouro da rede, métrica que será chamada de *atraso* médio da rede.

c) Tolerância a falhas

As más condições do meio físico onde a RSSF se encontra podem fazer com que um sensor falhe. O término da bateria é outro fator que faz com que um nó sensor deixe de fazer parte da rede. Diferentes fatores podem ocasionar a falha de um nó sensor. Dependendo das condições e localização da rede, a troca dos sensores pode ser um fator crítico e, portanto, a rede deve ser tolerante a falhas. É desejável que esta tolerância a falhas seja transparente para a aplicação.

Os sensores realizam o roteamento de dados sensorizados para uma estação base como se fosse uma rede conectada em forma de árvore onde sua raiz corresponde ao nó sorvedouro. Assim, a falha de um único nó pode resultar na interrupção da transmissão numa porção da rede, ou seja, todo ramo correspondente a esse nó que falhou pararia de enviar dados à estação base, caso não exista um outro caminho de rede até o nó sorvedouro.

Um protocolo de roteamento pode ser tolerante a falhas utilizando técnicas de replicação de dados, mas isto implica em um maior consumo de energia e o protocolo deve garantir esta tolerância com um mínimo consumo de energia.

No caso de falha de poucos nós a rede deve adotar novas características para o roteamento de informações. Quando a falha ocorre em um alto número de nós devido a alguma destruição ou acidente, a estação base deve identificar que o funcionamento da rede de sensores não é mais garantido. Uma forma de garantir isso seria manter uma lista com todos os nós inativos. Uma segunda forma, como ocorre em alguns protocolos de roteamento, é descobrir as rotas novamente para garantir o funcionamento da rede de sensores. Uma terceira forma é o estabelecimento de múltiplas rotas para aumentar a confiabilidade da comunicação da rede.

Serão descritos a seguir, os principais protocolos existentes para redes de sensores, enfatizando suas principais características, para posterior comparação com o protocolo proposto por este trabalho.

2.6 Protocolos de Roteamento

Conforme mencionado no capítulo 1, o objetivo dos protocolos de roteamento para RSSF é encontrar as melhores rotas para transmitir dados de cada um dos nós sensores até o nó sorvedouro da rede, tentando na medida do possível minimizar o consumo de energia e maximizar o tempo de vida útil da rede. Um bom protocolo de roteamento para RSSF deve ainda ser capaz de adaptar-se a possíveis mudanças na topologia da rede, devido à mobilidade dos nós, inclusão de novos nós da rede e fim da vida útil de nós.

É importante ressaltar também que a escolha do melhor protocolo é fortemente dependente da classificação da rede de interesse, com base nos critérios citados na seção 2.4. Dado o fato de que essa classificação possui muitos quesitos, nenhum dos protocolos até então desenvolvidos é capaz de obter o desempenho ótimo para todos os tipos de redes existentes. Cada um dos autores trabalha com cenários específicos e desenvolve estratégias adaptadas a situações bem definidas. É tarefa do projetista avaliar da melhor forma possível em que classificação a rede se encontra para escolher o protocolo que irá obter o melhor desempenho de acordo com o objetivo da aplicação em questão.

2.6.1 Protocolos de roteamento plano

Os protocolos de roteamento plano consideram que todos os nós são iguais do ponto de vista funcional, ou seja, o mesmo algoritmo de roteamento é executado por todos os nós, sem diferenciação hierárquica ou física entre eles. Esses protocolos são projetados para redes homogêneas e planas. O protocolo proposto neste trabalho é pertencente a esta categoria. Os principais protocolos desta categoria são descritos a seguir.

2.6.1.1 Protocolo Difusão Direcionada

O protocolo de Difusão Direcionada (INTANAGONWIWAT, GOVINDAN e ESTRIN, 2000) é o protocolo de roteamento plano mais citado na literatura de RSSF. O objetivo do protocolo é estabelecer um mecanismo de disseminação de dados na rede. A difusão direcionada considera que diferentes nós da rede possuem informações de sensoriamento diferentes. Alguns dos nós que não possuem determinada informação

propagam mensagens de interesse pela rede. Os nós que recebem as mensagens de interesse verificam se possuem a informação requisitada e, em caso afirmativo, enviam a informação em direção ao nó que efetuou a requisição.

O protocolo de difusão direcionada utiliza o *roteamento orientado a dados*: ao invés de estabelecer um endereço de origem e um de destino para um determinado dado, como no caso dos endereços IP, os nós são endereçados pelos dados que monitoram. Assim, não é necessário que os nós guardem tabelas de roteamento para cada um dos elementos da rede.

O algoritmo de funcionamento do protocolo é ilustrado na Figura 8 e descrito resumidamente a seguir:

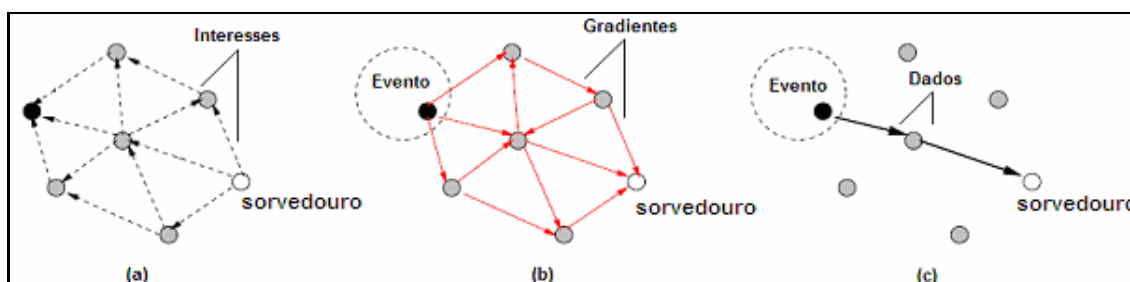


Figura 8: Funcionamento do Difusão Direcionada (INTANAGONWIWAT, GOVINDAN e ESTRIN, 2000)

a) A inicialização do protocolo ocorre a partir de um nó que tem interesse em uma determinada informação. Esse nó cria uma mensagem de interesse que pode ter, por exemplo, o seguinte formato:

```
tipo = animal de quatro patas
intervalo: 1s
coordenadas = [-100, 200, 200, 400]
início = 01:20:40
expiraEm = 01:30:40
```

A mensagem instrui todos os nós contidos dentro do retângulo definido pelas coordenadas a transmitirem a cada segundo uma mensagem de dados para o nó de origem, caso seus sensores detectem a presença de qualquer animal de quatro patas.

b) A mensagem de interesse é transmitida em forma de *broadcast* para todos os vizinhos do nó de origem. Cada nó que recebe a mensagem de interesse, guarda na memória uma nova entrada, que contém o interesse a ser monitorado e o nó do qual foi recebida a mensagem.

c) Após guardar a informação do interesse, cada nó que recebeu a mensagem transmite a mesma novamente em *broadcast* para seus vizinhos. Cada um deles novamente reenvia a mensagem para seus vizinhos. Os autores enfatizam que as informações armazenadas na memória de cada um dos nós estabelecem gradientes em direção ao nó de origem para aquela informação específica, conforme mostrado na Figura 8b. Esses gradientes são disseminados por toda a rede dentro de um certo tempo após a origem das mensagens.

d) Assim que algum dos nós detecta algum evento na rede, este verifica sua tabela de interesses, para verificar se recebeu alguma requisição relacionada à informação detectada. Caso afirmativo, o nó origina uma mensagem de dados e a envia para um dos nós para o qual possui um gradiente apontado. Essa mensagem é enviada apenas para um dos nós (*unicast*), conforme mostra a Figura 8c. É importante ressaltar que o nó que originou o evento envia as mensagens apenas para os seus nós vizinhos, sem nenhuma informação de onde está o destino final da mensagem.

Outro conceito utilizado na difusão direcionada é a *agregação de dados*: cada nó intermediário em uma cadeia de transmissão pode agregar dados recebidos de diferentes nós em um simples pacote para reduzir o número de transmissões e, conseqüentemente, o consumo de energia. A agregação de dados é ilustrada na Figura 9.

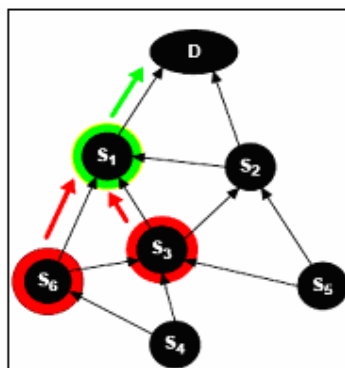


Figura 9: Agregação de Dados (NAKAMURA, FIGUEIREDO e LOUREIRO, 2004)

2.6.1.2 Protocolo SPIN

O protocolo SPIN, definido por (HEINZELMAN, KULIK e BALAKRISHNAN, 1999) utiliza informações da quantidade de energia disponível em cada nó para realizar o roteamento. Como a Difusão Direcionada, é reativo e baseado em disseminação de dados, ou

seja, procura disseminar uma informação para todos os nós da rede que têm interesse naquela informação. No SPIN, quando um nó detecta que está abaixo de um certo limite de energia em sua bateria, este procura participar menos do processo de roteamento, prolongando a vida útil da rede.

O SPIN possui três tipos diferentes de mensagens:

a) ADV – *Advertisement* – Quando um nó detecta uma nova informação, envia esta mensagem para a rede para alertar que possui um novo tipo de dado que pode ser de interesse da rede. Esta mensagem carrega *metadados*, que são uma forma sucinta de descrever a informação que o nó obteve do sensoriamento. O objetivo dos metadados é evitar a propagação de um grande volume de informação desnecessária para nós que não possuem interesse.

b) REQ – *Request* – Esta mensagem é um pedido de dados. Após receber uma mensagem ADV, um determinado nó pode enviar uma mensagem REQ para requisitar os dados completos, caso tenha interesse na informação anunciada.

c) DATA – Contém a informação completa relativa a um determinado evento. Carrega os dados descritos resumidamente por uma mensagem ADV anterior.

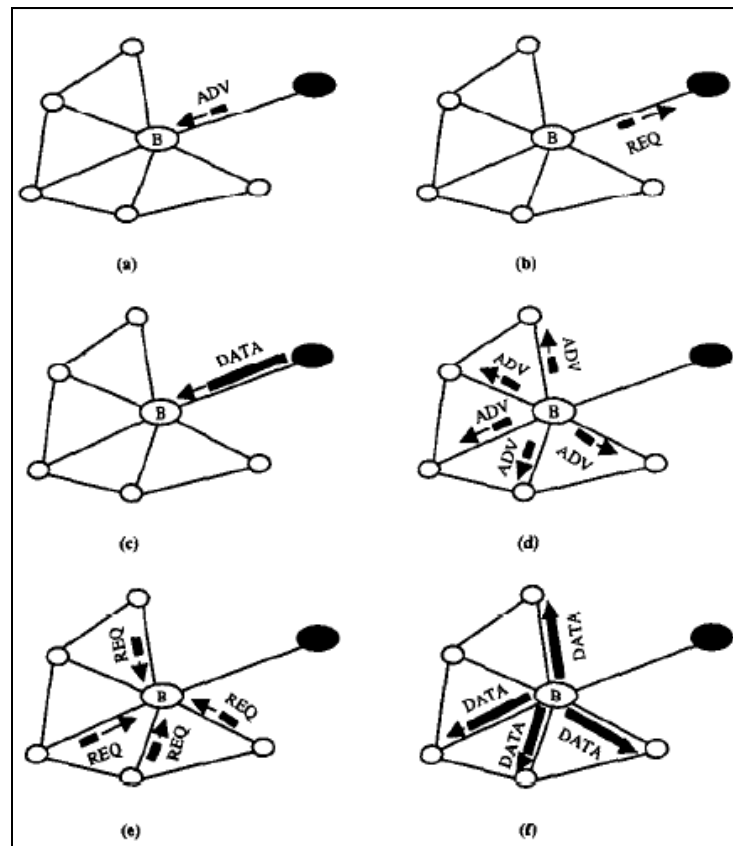


Figura 10: Funcionamento do SPIN (HEINZELMAN, KULIK e BALAKRISHNAN, 1999)

As etapas de funcionamento do SPIN são ilustradas na Figura 10. Ao obter uma nova informação de sensoriamento, um nó envia uma mensagem ADV para seus vizinhos (a). Os nós que receberem esta mensagem verificam se têm interesse na informação. Caso afirmativo, enviam uma mensagem REQ para o nó de origem, requisitando os dados completos (b). Ao receber a mensagem REQ, o nó de origem envia os dados através de uma mensagem DATA (c). Após os dados recebidos, o nó destino possui também uma nova informação, que pode ser anunciada através de uma nova mensagem ADV (d), dando continuidade ao processo de disseminação dos dados.

2.6.1.3 Protocolo MULTI

O protocolo MULTI (FIGUEIREDO, NAKAMURA e LOUREIRO, 2004), protocolo híbrido e adaptativo para disseminação de dados em Redes de Sensores sem Fio, é um protocolo que reúne características de dois outros protocolos definidos pelos mesmos autores:

o SID e o EF-Tree. O SID (*source initiated dissemination*) é um algoritmo reativo, onde o processo de disseminação é iniciado a partir do nó de origem dos dados. O EF-Tree (*Earliest-First tree*) é um algoritmo que constrói e mantém pró-ativamente uma árvore para disseminação de dados por toda a rede.

Os autores destacam que protocolos reativos se adaptam melhor a redes orientadas a eventos (baixo tráfego), assim como protocolos pró-ativos, que estabelecem as rotas independentemente da ocorrência de eventos, se adaptam melhor a redes com coleta de dados periódica e comunicação intensa. Também são sugeridos cenários dinâmicos, onde o volume de tráfego pode variar bastante, fugindo do ponto de operação de um protocolo específico. O MULTI procura adaptar-se a essas variações no tráfego, estabelecendo um limiar, a partir do qual o protocolo passa a atuar como pró-ativo (alto tráfego - EF-Tree) ou reativo (baixo tráfego – SID).

O SID funciona de forma semelhante ao SPIN, descrito na seção 2.6.1.2, com a diferença de que o SID não possui mecanismo de controle associado a energia dos nós. O nó que detecta uma nova informação faz a difusão da mesma na rede (Figura 11a) e os nós que têm interesse na informação enviam mensagens de requisição para o nó de origem (Figura 11b), que por sua vez envia os dados através de um caminho *unicast* (Figura 11c). Diferentemente do SPIN, no SID, o sorvedouro envia a requisição dos dados para o nó de origem também em modo *unicast*, escolhendo a cada salto o vizinho do qual a mensagem de anúncio de dados foi recebida primeiro.

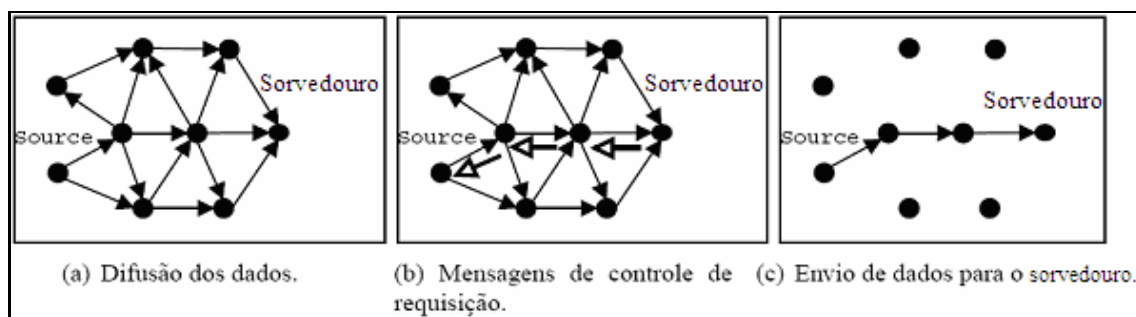


Figura 11: Funcionamento do protocolo SID (FIGUEIREDO, NAKAMURA e LOUREIRO, 2004)

O EF-Tree funciona de modo pró-ativo, num processo iniciado pelo nó sorvedouro. Para criar a árvore de roteamento, o nó sorvedouro envia em *broadcast* uma mensagem de controle para seus vizinhos. Cada vizinho que recebe essa mensagem, a reenvia para seus vizinhos. Cada nó guarda a informação do primeiro nó do qual recebeu essa mensagem de

controle (nó pai), criando para aquele nó o caminho padrão para roteamento dos dados recebidos. Esse esquema gera uma árvore sem ciclos, na qual a raiz é o nó sorvedouro. No capítulo 5, o EF-Tree será um dos protocolos comparados com o protocolo proposto neste trabalho.

O MULTI baseia-se no tráfego da rede para comutar entre o funcionamento do SID e do EF-Tree. Inicialmente o protocolo se comporta como o SID, recebendo as mensagens de dados a cada evento detectado. O nó sorvedouro monitora a cada período a quantidade de dados recebidos. Caso o volume de dados ultrapasse um valor pré-estabelecido, o nó sorvedouro passa a enviar mensagens de construção de árvore, conforme o funcionamento do EF-Tree. As mensagens de construção de árvore possuem uma validade. Vencida a validade, o sorvedouro pode reconstruir a árvore, se julgar que o tráfego ainda continua elevado, ou deixar que a árvore seja extinta, voltando a funcionar reativamente como SID.

Os autores fazem uma comparação entre o SID, o EF-Tree, o MULTI e o Difusão Direcionada, descrito na seção 2.6.1.1, para diferentes cenários de distribuição de tráfego e número de nós. Conforme esperado, o SID foi superior na situação de tráfego eventual com poucos nós, o EF-Tree foi superior na condição de tráfego contínuo ou com um número grande de nós e o MULTI, combinando as características dos dois protocolos, conseguiu obter um desempenho superior em energia e número de pacotes entregues em todas as condições de tráfego.

2.6.2 Protocolos de roteamento hierárquico

No roteamento hierárquico existe uma divisão dos nós da rede em duas classes: nós comuns e líderes de grupo (*cluster-heads*). A área de monitoramento é dividida em um certo número de regiões: os nós contidos em cada uma das regiões formam um grupo e elegem um dos nós para ser seu líder de grupo. Os demais nós enviam dados apenas para o líder de seu grupo, que os repassa para o nó sorvedouro. Os líderes de grupo podem também executar algumas funções, como agregação de dados ou roteamento apenas entre os líderes de grupo, antes de enviar os dados para o sorvedouro.

2.6.2.1 Protocolo LEACH

O protocolo LEACH (*Low-Energy Adaptive Clustering Hierarchy*) é o principal dos protocolos de roteamento hierárquico na literatura. É idealizado para redes pró-ativas e tem como objetivo a redução do consumo de energia em redes homogêneas e sem mobilidade (HEINZELMAN, CHANDRAKASAN e BALAKRISHNAN, 2000).

O protocolo divide o tempo em ciclos. No início de cada ciclo, todos os nós tentam se eleger como líderes. A eleição é feita através de um método probabilístico que garante a rotatividade dos nós como líderes de grupo. Essa rotatividade de líderes garante que o tempo de vida útil dos nós será homogeneamente distribuído, visto que, usando líderes de grupo fixos, estes seriam sobrecarregados com um excesso de mensagens, vindo a serem desativados mais cedo que os outros nós devido ao fim das baterias. A rotatividade de líderes é ilustrada na Figura 12 - Os pontos escuros representam os líderes escolhidos em dois ciclos distintos. As linhas delimitam os grupos de nós controlados por cada líder de grupo.

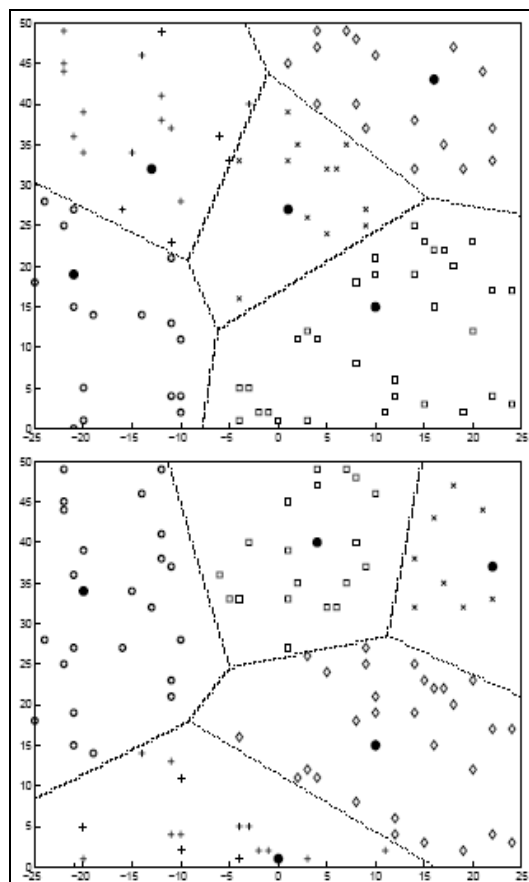


Figura 12: Rotatividade de líderes no LEACH (HEINZELMAN, CHANDRAKASAN e BALAKRISHNAN, 2000)

Conforme já mencionado, ao final de cada ciclo, cada líder de grupo agrega os dados dos nós de seu grupo, compactando o máximo possível e removendo as redundâncias de dados, enviando um pacote para o nó sorvedouro com as informações relevantes.

(HEINZELMAN, CHANDRAKASAN e BALAKRISHNAN, 2000) citam que o LEACH leva até 8 vezes mais tempo para a morte do primeiro nó e até 3 vezes mais tempo para a morte do último nó da rede em relação aos protocolos com o qual é comparado.

2.6.2.2 Protocolo SHARP

O SHARP (RAMASUBRAMANIAN, HAAS e SIRER, 2003), procura encontrar um equilíbrio entre o roteamento reativo e o roteamento pró-ativo em redes onde o tráfego é muito dinâmico, com presença de baixa e alta carga de pacotes em diferentes regiões da rede.

Numa rede SHARP, cada nó monitora continuamente as métricas de QoS (qualidade de serviço) da região ao seu redor. Fatores como número de pacotes perdidos, atraso de pacotes, tempo médio de vida dos enlaces, entre outros, são utilizados para julgar se a região em torno de um certo nó se encontra em uma situação de alto ou baixo tráfego. Caso o tráfego esteja alto, o nó estabelece uma região de roteamento pró-ativo em torno de si, com um raio que é função do volume de tráfego na região. Dentro da região pró-ativa, os nós estabelecem caminhos de roteamento pré-definidos, que são utilizados para encaminhar os pacotes recebidos. Fora das regiões pró-ativas, o roteamento se dá de forma reativa, estabelecendo os caminhos apenas quando necessário para a transmissão de dados. A formação de regiões pró-ativas é mostrada na Figura 13.

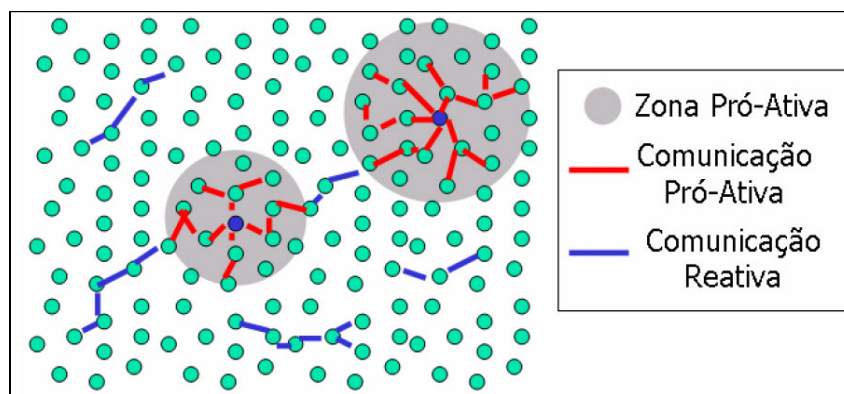


Figura 13: Regiões pró-ativas formadas pelo SHARP (RAMASUBRAMANIAN, HAAS e SIRER, 2003)

A construção de caminhos nas regiões pró-ativas é realizada utilizando um protocolo de construção de árvore, semelhante ao usado pelo protocolo EF-Tree (seção 2.6.1.3): o nó destino propaga um pacote de construção de árvore, carregando o raio da região pró-ativa que deve ser criada. Os nós que recebem esse pacote guardam o identificador do nó do qual o receberam como seu nó pai e repassam o pacote para seus vizinhos. Nós que estiverem fora do raio requisitado ignoram o pacote recebido.

Os experimentos dos autores variam o raio das zonas pró-ativas entre 0 (puramente reativo) e o raio total do campo de monitoramento (puramente pró-ativo). Os resultados mostram que é possível balancear os raios de forma a otimizar o tráfego de roteamento, o atraso e a perda de pacotes na rede. A grande vantagem do protocolo é possibilitar uma adaptação localizada do roteamento, favorecendo a operação em casos de tráfego heterogêneo.

2.6.2.3 Protocolo PEGASIS

O protocolo PEGASIS (LINDSEY e RAGAVENDRA, 2002) é baseado no conceito de correntes de nós: alguns nós se interligam, formando uma corrente, conforme mostrado na Figura 14.

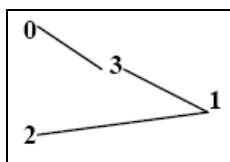


Figura 14: Exemplo de corrente de nós no PEGASIS (LINDSEY e RAGAVENDRA, 2002)

A cada turno, um dos nós da corrente é eleito líder. Dada uma corrente com N nós, a expressão utilizada para eleger o líder no i -ésimo turno é $i \bmod N$, ou seja, o resto da divisão de i por N . Isso garante o revezamento de líderes, como realizado no LEACH. O líder de cada turno recebe os dados de cada um dos nós da corrente, através de múltiplos saltos, e os transmite ao nó sorvedouro. A cada salto, é feita uma agregação de dados, de forma a diminuir o tamanho das mensagens e o número de mensagens na rede.

De acordo com os autores, o PEGASIS consegue obter o dobro do tempo de vida de rede em comparação com o LEACH, para uma rede de 50x50 m e o triplo do tempo de vida para uma rede de 100x100 m.

2.6.3 Protocolos de roteamento geográfico

Os protocolos de roteamento geográfico utilizam informações de posicionamento espacial para efetuar o roteamento dos dados. Essas informações podem ser obtidas através de um equipamento de localização (GPS), ou mesmo através de um sistema de posicionamento local, definido entre os nós vizinhos em uma determinada região. Os principais protocolos de roteamento geográfico são descritos a seguir:

2.6.3.1 Protocolo ICA

O protocolo ICA (HABIB, CAMARA e LOUREIRO, 2004) é baseado no LEACH e possui alguns mecanismos para aumentar o tempo de vida da rede. É idealizado para redes com coleta de dados periódica.

A seleção dos líderes é realizada da mesma forma como ocorre no LEACH. Cada líder, ao ser eleito, envia sua posição geográfica aos nós vizinhos, de modo a auxiliar na decisão.

No ICA, os líderes de grupo não enviam as mensagens diretamente ao nó sorvedouro. Ao invés disso, enviam as mensagens para o líder de grupo mais próximo deles, na direção do nó sorvedouro. Isso economiza energia, devido ao fato de que não é mais necessário transmitir dados a grandes distâncias, como ocorre no LEACH.

O ICA propõe um mecanismo que impede que os líderes mais próximos do nó sorvedouro morram mais cedo que os outros líderes: caso um líder caia abaixo de um nível definido de bateria, este pode se recusar a repassar mensagens de outros líderes. O nó que enviou a mensagem para ser repassada, deve procurar um outro líder ou enviar a mensagem diretamente ao nó sorvedouro. O objetivo é obter um tempo de vida o mais homogêneo possível para todos os nós da rede.

O ICA consegue obter uma duração da rede de até seis vezes maior, se comparado ao protocolo LEACH, de acordo com experimentos realizados pelos autores.

2.6.3.2 Protocolo GPSR

O protocolo GPSR (*Greedy Perimeter Stateless Routing*), proposto por (KARP e KUNG, 2000), baseia-se em um roteamento guloso, onde cada nó envia seus pacotes de dados para o nó mais próximo do nó sorvedouro que estiver ao seu alcance (Figura 15).

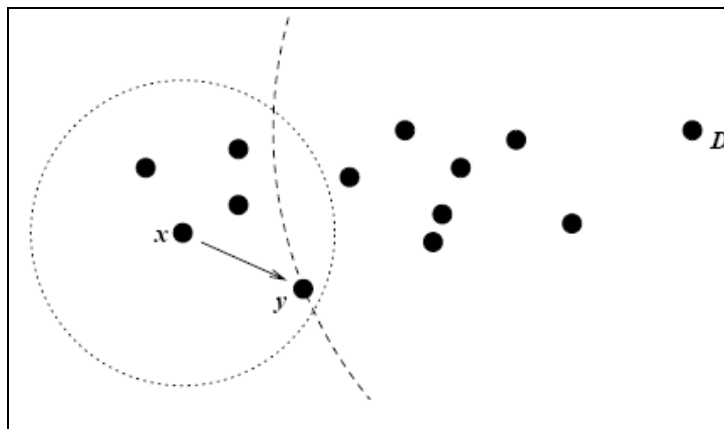


Figura 15: Roteamento guloso do GPSR – x enviando mensagem para D (KARP e KUNG, 2000)

Um problema enfrentado pelo GPSR é o roteamento de pacotes para nós mais próximos do sorvedouro que o nó atual, mas que não possuam conectividade com nenhum outro nó mais próximo do sorvedouro. Isso significa que o nó atual deve rotear os pacotes dando a volta por nós mais distantes para chegar aos nós mais próximos do sorvedouro. Essa situação é mostrada na Figura 16.

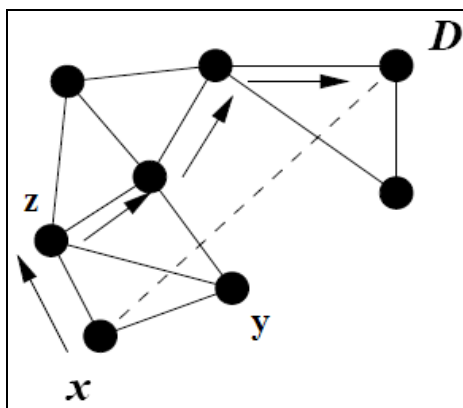


Figura 16: Roteamento perimetral do GPSR – x enviando mensagem para D (KARP e KUNG, 2000)

A solução encontrada pelos autores foi utilizar um algoritmo de *roteamento perimetral*. O algoritmo perimetral funciona da seguinte forma: ao chegar a um nó que é o mais próximo do sorvedouro entre seus vizinhos, o algoritmo traça uma linha imaginária entre o nó atual e o sorvedouro. Essa linha é representada pela linha tracejada na Figura 16. O algoritmo roteia o pacote para o nó mais próximo seguindo o polígono formado pelos arcos que saem do nó atual e pela linha tracejada no sentido anti-horário. O percurso anti-horário deve passar obrigatoriamente pela linha tracejada.

Assim que o pacote detecta que chegou a um nó mais próximo do sorvedouro do que aquele onde se iniciou o roteamento perimetral, este volta a ser roteado de forma gulosa, até chegar ao sorvedouro ou em uma outra área onde seja necessário o roteamento perimetral.

2.6.4 Discussão e comparações

Verifica-se nesta seção que existem na literatura diversas classes de protocolos que podem ser utilizados em diferentes RSSF, de acordo com as características específicas da aplicação. Os protocolos hierárquicos podem otimizar a transmissão dos dados, estabelecendo multiplexação do tempo para a transmissão de cada nó e evitando colisões e interferência. Por outro lado, protocolos hierárquicos tendem a apresentar maior atraso (latência), devido à espera pela transmissão de todos os nós dentro de um grupo antes de transmitir os dados ao sorvedouro.

Em termos de tráfego de controle, o roteamento plano normalmente gera menos pacotes, pois precisa apenas estabelecer caminhos entre nós vizinhos, sem se preocupar em definir líderes, esquemas de multiplexação de dados etc. Esse fator pode ser usado para economizar energia, dependendo do tipo de aplicação.

Outras vantagens do roteamento plano são a maior robustez com relação à perda do sincronismo entre os relógios dos nós e o fato de a inteligência da rede não estar concentrada em apenas um único elemento (*cluster-head*), o que faz com que a rede não seja tão dependente de alguns nós críticos em determinados momentos.

Os protocolos de roteamento geográfico, no que diz respeito à formação de rotas, se comportam alguns como planos e alguns como hierárquicos, estando sujeitos às mesmas vantagens e desvantagens encontradas nos dois esquemas. Naturalmente, é de se esperar que os equipamentos de localização (GPS) necessários para esses protocolos, aumentem

consideravelmente o consumo de energia, não sendo viáveis para aplicações com forte restrição de baterias.

No próximo capítulo serão discutidos os protocolos baseados em colônia de formigas, os quais serviram de inspiração para o protocolo proposto neste trabalho. Uma tabela comparativa das principais características dos protocolos discutidos neste capítulo e no capítulo 3 será apresentada no capítulo 4.

3 OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS

Estudos na área de Colônia de Formigas verificaram que o meio que estes insetos utilizam para comunicar os caminhos uns com os outros e decidir para onde ir consiste de *trilhas de feromônio*. Uma formiga em movimento pode depositar feromônio no solo, em quantidade variável, criando uma trilha desta substância para marcar um determinado caminho que tenha seguido. Enquanto uma formiga isolada se move aleatoriamente no espaço, outra que encontre uma trilha de feromônio tende a segui-la com maior probabilidade, reforçando ainda mais a trilha com seu próprio feromônio.

O comportamento das formigas pode ser descrito resumidamente da seguinte forma:

- Formigas em busca de alimento saem do ninho sem informação prévia da localização potencial deste e se movimentam de forma aleatória procurando por este alimento.
- Durante a busca, as formigas depositam uma substância no solo, o feromônio, que serve como trilha para o retorno ao ninho quando do encontro de alimento.
- Ao encontrar uma fonte de alimento, uma formiga passa a carregar alimento de volta ao ninho, guiando-se pela trilha depositada anteriormente e depositando mais feromônio pela trilha, de forma a reforçar a trilha já demarcada.
- Outras formigas que estiverem à procura de alimento, ao encontrar uma trilha de feromônio, passam a segui-la, por saberem instintivamente que a trilha leva a uma fonte de alimento.
- A cada momento, mais e mais formigas encontram as trilhas de feromônio que levam ao alimento, reforçando ainda mais a trilha e gerando um comportamento emergente – *estigmergia* (DORIGO, BONABEAU e THERAULAZ, 2000) - no qual a maioria das formigas passa a seguir a mesma trilha, indo à fonte de alimento e retornando ao ninho para o armazenamento da carga.

Vale ressaltar aqui o potencial adaptativo do procedimento utilizado pelas formigas: sem nenhuma informação inicial sobre a topologia do terreno ou a localização do alimento, as formigas conseguem encontrar o alimento e comunicarem-se indiretamente umas com as outras de forma a indicar os melhores caminhos (auto-organização).

O comportamento coletivo derivado desse processo é chamado por Dorigo, Maniezzo e Colorni (1996) de comportamento *autocatalítico*, ou seja, caracteriza-se por uma

realimentação positiva, de forma que cada vez mais a trilha seguida seja reforçada e mais formigas sigam por aquele caminho. Se não existir nenhum mecanismo que limite a quantidade de feromônio nos caminhos, o sistema tende a uma convergência muito rápida, podendo gerar até mesmo consequências nocivas para o sistema, como uma sobrecarga do caminho de passagem de formigas, devido ao excesso de tráfego.

O que limita o excesso de feromônio em determinados caminhos é a evaporação. Com o tempo as trilhas marcadas perdem um pouco da intensidade, levando a uma estabilização do nível de feromônio em cada caminho. Se as formigas encontram por acaso um caminho melhor que o originalmente marcado, estas passam a reforçar mais o novo caminho. Com o passar do tempo, o feromônio do caminho anterior evapora, evitando que as formigas sigam por um caminho de qualidade inferior ao melhor já encontrado.

Além disso, novos caminhos podem ser criados, caso obstáculos interrompam os caminhos existentes (adaptabilidade). Esse potencial é ilustrado na Figura 17.

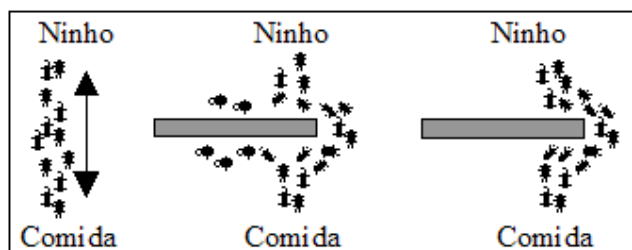


Figura 17: Formigas encontrando o melhor caminho ao redor de um obstáculo (PARPINELLI, LOPES e FREITAS, 2001)

O algoritmo de Otimização por Colônia de Formigas (ACO), criado com base nesse comportamento, será descrito na próxima seção.

3.1 Descrição do Algoritmo Geral de ACO

O comportamento das formigas reais, descrito anteriormente, serviu como base para um algoritmo de otimização bastante útil em especial para busca em grafos. O algoritmo proposto é detalhado por (DORIGO, MANIEZZO e COLORNI, 1996) e pode ser descrito resumidamente conforme segue:

a) Dado um grafo com pesos, onde o peso de cada arco pode representar, por exemplo, a distância de um nó a outro, deseja-se encontrar a rota mais curta entre dois nós de interesse. Esse objetivo é ilustrado na Figura 18 .

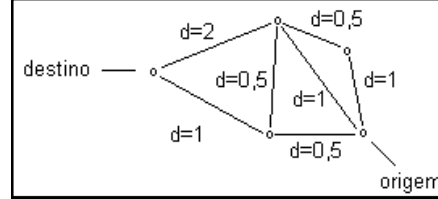


Figura 18: Grafo com pesos

b) É possível mapear a origem como sendo o ninho das formigas e o destino como sendo a fonte de comida. Assume-se então que existem N agentes (formigas) no nó de origem (ninho).

c) Supondo tempo discreto, a cada passo de tempo t , cada agente pode se mover uma distância de um nó, atravessando qualquer arco. Dado que um agente esteja, por exemplo, no nó j , a probabilidade de escolher cada um dos caminhos é dada por:

$$P_{ij} = \frac{\tau_{ij}}{\sum_{vizinhos} \tau} \quad (1)$$

onde: - P_{ij} = Probabilidade de seleção do vizinho i do nó atual

- τ_{ij} = peso do arco que leva o nó atual ao vizinho i

- $\sum_{vizinhos} \tau$ = somatório dos pesos de todos os arcos saindo do nó atual

d) Inicialmente, o peso em direção a cada nó do grafo é 0, de forma que a seleção é aleatória. Cada agente que chega ao nó de destino retorna ao nó de origem atualizando o peso probabilístico em cada arco sobre o qual passou na busca pelo nó de destino. Esta variação é calculada dependendo do problema considerado e da variável que se deseja otimizar. Se, por exemplo, o objetivo fosse otimizar a distância no problema da Figura 18, uma função de incremento para cada arco poderia ser dada por:

$$\tau_{ij} = \frac{1}{\sum_k d_k} \quad (2)$$

onde: $\sum_k d_k$ = somatório das distâncias percorridas em todos os arcos da origem até o destino.

e) O peso probabilístico de cada arco é decrementado a cada instante de tempo t , segundo uma taxa pré-estabelecida dada por:

$$\tau_{ij}(t+1) = \rho \tau_{ij}(t) \quad (3)$$

onde ρ é um coeficiente tal que $(1-\rho)$ é a evaporação da trilha num intervalo de tempo t . O objetivo da evaporação é manter o comportamento probabilístico do sistema, eliminando com o tempo caminhos que não estejam mais sendo utilizados pelos agentes em movimento pelo grafo.

Percebe-se no algoritmo que cada agente que retorna do nó de destino incrementa o peso probabilístico do caminho percorrido, induzindo outras formigas a seguirem o mesmo caminho. Quanto mais curto o caminho (menor distância) maior será o incremento do peso, e maior a probabilidade de que outras formigas sigam pelo mesmo caminho. Apesar de o caminho ser reforçado a cada retorno de um agente, o comportamento da movimentação de novos agentes ainda assim é probabilístico, deixando a possibilidade de que alguns agentes sigam por caminhos diferentes e até mesmo encontrem caminhos melhores.

Existe a possibilidade de se incluir no algoritmo a capacidade de guardar a lista dos nós percorridos no caminho. Iniciando o algoritmo com N agentes e aguardando que todos retornem ao nó inicial, armazena-se a melhor rota encontrada. Ao repetir o processo M vezes, mantendo a quantidade de feromônio depositada nos caminhos de uma iteração para outra, é possível observar a evolução da otimização. O armazenamento, a cada iteração, da melhor rota encontrada permite observar graficamente o comportamento dinâmico das soluções. A solução produzida pelo método é obtida ao final das $M \times N$ iterações. Quanto maior o número de iterações, maior a chance de uma das soluções obtidas ser ótima ou muito próxima da ótima.

A técnica de ACO pode ser estendida para trabalhar com outros objetivos, de acordo com o problema a ser abordado. Para isto, a expressão (2) deve ser modificada, de forma a

incrementar os pesos dos arcos proporcionalmente ao parâmetro do sistema a ser otimizado. Assim, os caminhos que apresentarem melhores soluções, do ponto de vista do parâmetro de interesse serão mais reforçados, tendo preferência para percursos futuros.

Dentre as principais vantagens do método de ACO, destacam-se:

i) Baixo custo computacional: as formigas executam operações bastante simples: apenas chegam a um nó e calculam qual o próximo nó a partir da quantidade de feromônio nos arcos do grafo. Outros métodos matemáticos de otimização em grafos demandam muito mais processamento para chegar à solução ótima.

ii) Não necessita informação global sobre o sistema: cada agente precisa saber apenas informações sobre o nó atual e sobre os arcos que saem dele. Não é necessário conhecer mais nenhum parâmetro global do grafo. Isso possibilita que o algoritmo possa ser aplicado a muitos outros problemas combinatórios, como o conhecido “Caixeiro Viajante” (APPLEGATE, BIXBY, CHVÁTAL *et al*, 2006).

iii) Tempo de execução reduzido: para grafos com muitos nós, a complexidade dos métodos matemáticos de otimização tradicionais pode significar um tempo muito grande de execução. A técnica de ACO é capaz de chegar a uma solução ótima, ou muito próxima da ótima, com um número pequeno de iterações (agentes), mesmo para grafos com um grande número de nós.

iv) Busca em um espaço reduzido de soluções: como todos os métodos meta-heurísticos, o método ACO se baseia em iterações, que testam uma solução (caminho) inicialmente aleatória e calcula o quão boa é esta solução – busca global. A seguir são testadas soluções (caminhos) semelhantes à anterior, com pequenas modificações, para verificar se uma solução melhor é encontrada – busca local. O processo se repete até um número pré-definido de iterações, no fim das quais é obtida uma solução final ótima ou próxima da ótima. Em geral, este método é muito mais vantajoso em termos de tempo e complexidade computacional do que um método matemático ou exaustivo, que teria que varrer todo o espaço de soluções até encontrar uma solução ótima.

A única limitação a ser citada é a de que a técnica de ACO, como outras meta-heurísticas, não garante a solução ótima ao final da execução. O método pode convergir para um ótimo local, ou chegar próximo ao ótimo global, sem, no entanto atingir a solução exata no número definido de iterações. É tarefa do projetista ajustar os parâmetros do problema de forma a obter a melhor solução possível dentro do tempo disponível.

A técnica escolhida para resolver o problema do roteamento em RSSF abordado neste trabalho foi baseada em ACO uma vez que a aplicação de ACO para o roteamento na área de Redes de Computadores (DI CARO, DUCATELLE e GAMBARDELLA 2005) e em outros problemas que envolvem otimização em grafos é bastante difundida na literatura. Alguns algoritmos da área (em especial os dedicados aos problemas de roteamento) são descritos a seguir.

3.2 Aplicações de ACO em redes de computadores

A técnica de ACO é uma heurística largamente aplicada em redes de computadores. O problema de roteamento de pacotes em uma rede de computadores é bem semelhante ao das formigas em busca de alimento: descobrir um caminho ótimo entre dois nós de um grafo. Grafos são o modelo mais tradicional para representar redes. Garbe (2004) faz um apanhado histórico sobre os diversos algoritmos baseados em formigas utilizados em redes de computadores. Os principais algoritmos citados são mostrados na Tabela 3.

Tabela 3: Algoritmos baseados em formigas para redes de computadores

Algoritmo	Ano	Autores
<i>ABC</i>	1996	Schoonderwoerd <i>et al.</i>
<i>AntNet e AntNet-FA</i>	1997	Di Caro e Dorigo
<i>AntNet-FS</i>	1997	Di Caro e Dorigo
<i>ASGA</i>	1997	White.
<i>BeeHive</i>	2004	Farooq.
<i>AntHocNet</i>	2005	Di Caro, Ducatelle e Gambardella.

O objetivo de um algoritmo de roteamento é obter uma tabela de roteamento. Cada um dos nós da rede deve guardar sua própria tabela de roteamento, que relaciona cada nó da rede com um dos nós vizinhos do nó em questão. Ao receber um pacote de dados, o nó verifica para que endereço o pacote está direcionado e consulta sua tabela de roteamento para saber para qual de seus vizinhos deve enviar os dados para que cheguem ao destino final. O roteamento é feito nó a nó da origem ao destino, com base nas tabelas de roteamento de cada nó, como um algoritmo distribuído. Um exemplo de rede com nós roteadores é mostrado na

Figura 19. A figura ilustra uma rede com 4 nós roteadores. Tomando como exemplo o nó A, todo pacote que chega a este nó pode ter como destino qualquer um dos outros nós da rede. A tabela de roteamento deste nó deve conter o vizinho para qual cada pacote de dados deve ser enviado, com base no destino final do pacote. Um exemplo de tabela de roteamento para o nó A é mostrado na Tabela 4.

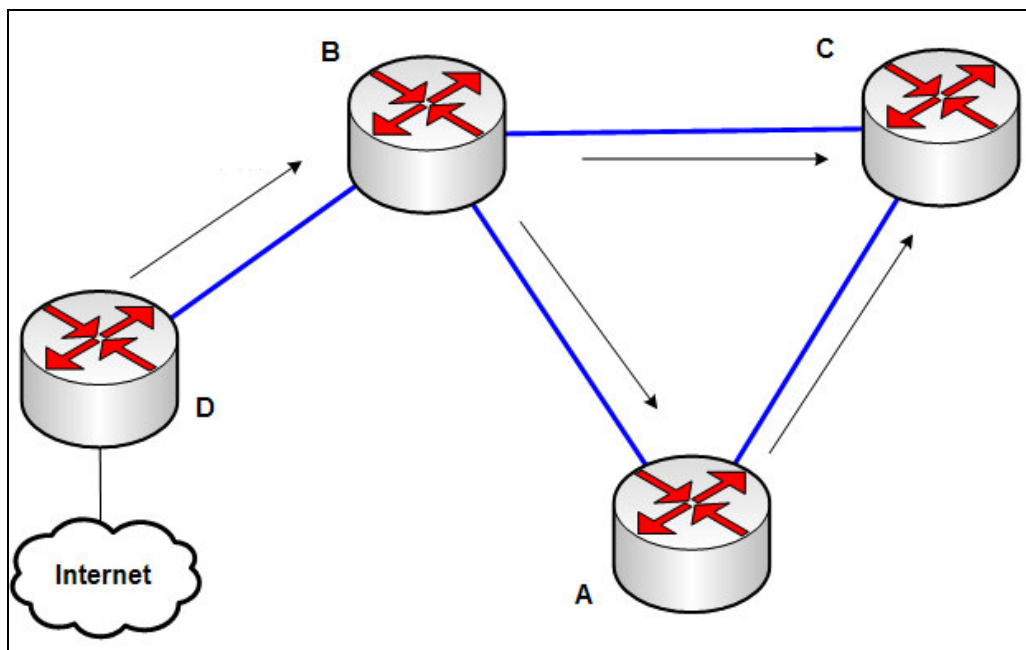


Figura 19: Rede de nós roteadores

Tabela 4: Tabela de roteamento para o nó A

Destino	Próximo nó
B	B
C	C
D	B
Padrão (<i>default</i>)	B

A Tabela 4 mostra que cada pacote que chega ao nó A com destino a outros nós deve ser redirecionado para o nó C, caso o destino final seja este nó, ou para o nó B, caso o destino final sejam os nós B, D ou nós desconhecidos (rota padrão). Esta tabela de roteamento é apenas ilustrativa. Numa tabela real, os nós estariam representados por seus endereços de rede (IP ou similar).

No caso da rede ser representada por um grafo, aplica-se ACO para descobrir o melhor caminho de um determinado nó até todos os outros nós. O problema poderia ser visto por cada nó como se este nó fosse o ninho das formigas e cada um dos outros nós fossem fontes de alimento. A busca baseada em ACO encontraria então os melhores caminhos entre todos os nós da rede, atualizando a tabela de roteamento de cada nó periodicamente.

A grande vantagem do uso do ACO em redes de computadores é o fato do método poder ser aplicado em tempo real, durante o funcionamento da rede. O cálculo em tempo real da tabela de roteamento é importante porque as redes podem possuir um comportamento dinâmico, com variações de tráfego e de capacidade de enlaces, de forma que a qualidade das rotas pode variar com o tempo. O mapeamento entre os elementos de uma rede e os elementos de um ACO é mostrado na Tabela 5.

Tabela 5: Mapeamento entre elementos de rede e elementos do ACO

Elemento de Rede	Elemento ACO
Nó roteador	Nó de grafo
Enlace de dados entre dois nós	Arco de grafo entre dois nós
Pacotes de controle	Agentes (formigas)
Peso probabilístico de cada enlace	Quantidade de feromônio em cada arco
Tabela de roteamento	Tabela de feromônio
Ninho das formigas	Nó de origem
Fonte de alimento	Nó destino

Os pacotes de controle enviados na rede se comportam como agentes que são enviados de um roteador a outro aleatoriamente até atingir o destino. Chegando lá, retornam pelo mesmo caminho para o nó de origem, atualizando os pesos probabilísticos de roteamento para cada nó. Os próximos pacotes enviados são influenciados pelos pesos atualizados, tendendo a seguir pelos melhores caminhos, como ocorre com as formigas em busca de alimento.

Outra vantagem é a de que o algoritmo é executado de forma distribuída, ou seja, cada nó é responsável apenas por receber as mensagens (agentes) e selecionar o próximo nó para o qual a mensagem será enviada. O custo computacional destas operações é bastante baixo, não causando influência significativa sobre a carga de processamento de cada roteador, nem sobre os atrasos dos pacotes. Além disso, não é necessário enviar toda a informação para um servidor centralizado que execute o cálculo das tabelas, o que evita gerar mais tráfego na rede.

Vale ressaltar aqui que os pacotes de agentes podem ser os mesmos pacotes que carregam os dados da aplicação da rede ou podem ser pacotes dedicados apenas para o método de otimização. Caso os agentes estejam embutidos nos pacotes de dados da aplicação, é necessário haver um tráfego mínimo para que o método possa executar um número de iterações satisfatório. No caso de pacotes de controle dedicados, o algoritmo pode ser executado paralelamente ao tráfego da rede, apenas atualizando as tabelas de roteamento com as melhores rotas encontradas, para que os pacotes de dados as utilizem.

Nas seções a seguir serão descritos alguns dos principais protocolos baseados em ACO utilizados para o roteamento em redes.

3.2.1 Protocolo *AntNet*

O protocolo *AntNet* (DI CARO e DORIGO, 1998) foi uma das primeiras aplicações da Otimização por Colônia de Formigas para o roteamento em redes. Trata-se da aplicação direta de ACO ao cenário descrito na seção 3.2. O *AntNet* atualiza as tabelas de roteamento de cada nó de uma rede de acordo com a movimentação de agentes que modificam os pesos probabilísticos de cada arco enquanto se movimentam pela rede. O algoritmo é descrito resumidamente a seguir:

a) Em intervalos regulares, cada nó s da rede cria um *agente em avanço* (*forward agent*) $F_{s \rightarrow d}$, que é uma mensagem de dados com destino a outro nó d qualquer da rede escolhido aleatoriamente. Os destinos são escolhidos aleatoriamente para igualar os padrões de tráfego.

b) Inicialmente o peso probabilístico para a seleção dos caminhos é 0, de forma que a seleção dos próximos vizinhos é aleatória com distribuição uniforme. O agente em avanço percorre nó a nó escolhendo caminhos aleatórios até chegar ao nó d .

c) O identificador de cada nó visitado k e o tempo decorrido desde a criação do agente é armazenado em uma pilha de memória $S_{s \rightarrow d}(k)$ carregada pelo agente.

d) Se for detectado um ciclo, ou seja, se o agente retornar a um nó já visitado, toda a pilha de nós sobre aquele agente é removida, com o objetivo de remover ciclos no caminho do agente.

e) Quando o nó de destino d é alcançado, o nó de destino cria um *agente em retorno* (*backward agent*) $B_{d \rightarrow s}$, que guarda na memória todos os dados trazidos pelo agente em avanço correspondente.

f) O agente em retorno volta ao nó de origem pelo mesmo caminho percorrido pelo agente em avanço correspondente. A cada nó k no caminho, o agente desempilha sua pilha $S_{s \rightarrow d}(k)$ para saber qual é o nó do próximo salto.

g) Chegando a um nó k , vindo de um nó f vizinho, o agente atualiza na memória do nó k a tabela de roteamento deste nó. A tabela de roteamento do nó k , que tenha i vizinhos, para uma rede com N nós tem a forma mostrada na Tabela 6.

Tabela 6: Tabela de roteamento no *AntNet*

Destino	Vizinho 1	Vizinho 2	...	Vizinho i
Nó 1	p_{11}	p_{12}	...	p_{1i}
Nó 2	p_{21}	p_{22}	...	p_{2i}
...	...	...	...	...
Nó $k-1$	$p_{(k-1)1}$	$p_{(k-1)2}$	...	$p_{(k-1)i}$
Nó $k+1$	$p_{(k+1)1}$	$p_{(k+1)2}$	...	$p_{(k+1)i}$
...	...	...	...	...
Nó N	p_{N1}	p_{N2}	...	p_{Ni}

A tabela relaciona cada nó destino possível na rede com uma probabilidade p_{di} , que representa a probabilidade de um pacote com destino ao nó d , ser roteado para o vizinho i . A probabilidade atualizada por cada agente é aquela que relaciona o destino final do qual o agente está retornando com o último nó visitado pelo mesmo. O incremento do peso segue a expressão (4).

$$p_{di} = p_{di} + r(1 - p_{di}) \quad (4)$$

onde r é uma constante entre 0 e 1 que é calculada com base na qualidade do caminho percorrido. Essa qualidade pode ser calculada considerando-se o tempo de percurso entre origem e destino, a capacidade de tráfego dos enlaces, etc. Percebe-se aqui que o crescimento do peso é dependente do próprio peso atual do arco em questão, de forma que o incremento é

mais agressivo no início do processo, e mais suave conforme o peso aumenta, minimizando os efeitos de convergência prematura do sistema para máximos locais.

h) Os pacotes de dados gerados pelas aplicações da rede são roteados também de forma probabilística, utilizando a tabela de roteamento de cada nó, como mostra a Tabela 6.

3.2.2 Protocolo *AntHocNet*

O protocolo *AntHocNet*, criado por Di Caro, Ducatelle e Gambardella (2005) é uma aplicação de ACO para redes móveis *Ad Hoc* (MANET). Essas redes são redes sem fio onde os nós são móveis e que possibilitam inclusão e retirada de nós da rede durante o seu funcionamento. É necessário assim, um protocolo que possa ser adaptativo e tolerante a falhas, devido ao comportamento dinâmico da topologia da rede.

O *AntHocNet*, ao contrário do *AntNet*, não mantém pró-ativamente uma tabela de roteamento o tempo todo para todos os destinos. Ao invés disso, efetua a busca de caminhos reativamente, quando é necessário abrir uma sessão de dados entre dois nós quaisquer. Neste momento, são criados *agentes reativos em avanço*, para encontrar múltiplos caminhos em relação ao destino. Em resposta, o destino envia *agentes em retorno*, que retornam à origem para consolidar os caminhos encontrados. Da mesma forma que nos outros protocolos, os agentes em retorno atualizam as tabelas de pesos probabilísticos (feromônio) no caminho e os agentes ou pacotes de aplicação futuros seguem caminhos definidos de acordo com os pesos atribuídos.

Durante as sessões de dados de aplicação, os caminhos criados são monitorados, mantidos e melhorados por *agentes pró-ativos em avanço*. Esses agentes reforçam os caminhos obtidos pelos agentes reativos, de forma a manter em boa qualidade a sessão de dados já estabelecida.

O algoritmo do *AntHocNet* é descrito brevemente a seguir:

a) Com o sistema em repouso, caso um nó s qualquer deseje estabelecer uma sessão de dados com um outro nó d , aquele cria um agente reativo em avanço F_s^d . Este agente é transmitido em *broadcast* para todos os vizinhos do nó s .

b) Os nós vizinhos propagam novamente o agente em *broadcast*, de forma que rapidamente o agente se propaga pela rede e chega ao destino. Um exemplo de percurso de um agente em avanço é mostrado na Figura 20.

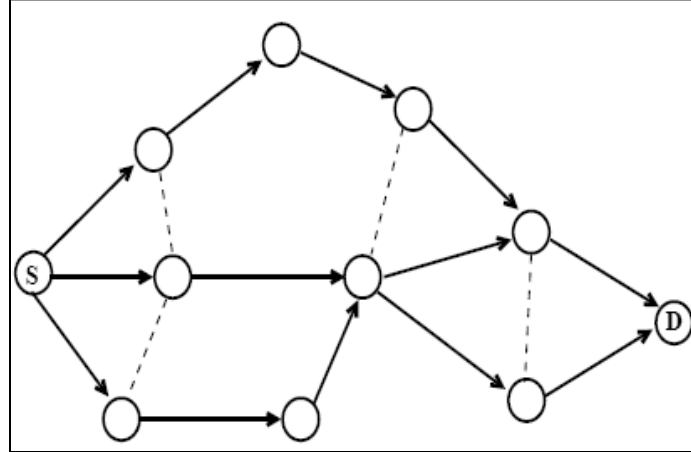


Figura 20: Percurso de um agente reativo em avanço (DI CARO, DUCATELLE e GAMBARDELLA, 2005)

c) O agente então se torna um agente em retorno e volta para a origem pelo mesmo caminho seguido anteriormente, atualizando as tabelas de pesos probabilísticos para aquele destino em todos os nós. No retorno, a transmissão é *unicast*, ou seja, apenas um nó de cada vez até chegar à origem. É importante ressaltar, que devido aos *broadcasts* iniciais, o agente em avanço é multiplicado, de forma que vários agentes em retorno chegam à origem e múltiplos caminhos são marcados com trilhas para o nó destino.

d) Daí em diante, os pacotes de dados da aplicação são roteados probabilisticamente nó a nó, de acordo com os pesos probabilísticos de cada arco, atribuídos no passo c). Ao chegar a um determinado nó i , a probabilidade de seleção do próximo nó é calculada pela expressão (5).

$$P_{nd} = \frac{(\tau_{nd}^i)^{\beta_2}}{\sum_{j \in N_d^i} (\tau_{jd}^i)^{\beta_2}} \quad (5)$$

onde: P_{nd} = probabilidade de escolha do vizinho n dado que o destino é d

(τ_{nd}^i) = peso probabilístico (quantidade de feromônio) do arco entre n e d

$\sum_{j \in N_d^i} (\tau_{jd}^i)^{\beta_2}$ = somatório dos pesos (feromônio) de todos os arcos saindo do nó atual i , com direção ao nó d .

β_2 = expoente maior ou igual a 1, que diminui o caráter exploratório dos agentes, aumentando a velocidade da convergência quanto maior seu valor.

e) Durante o período em que a seção de dados estiver ativa, novos *agentes pró-ativos em avanço* são enviados da origem ao destino, de modo a manter o caminho já existente, encontrar novos caminhos em caso de falha de algum dos nós e, se possível, encontrar caminhos melhores. Esses agentes seguem em modo *unicast* com o roteamento seguindo as trilhas de feromônio existentes (equação 5). Porém, a cada nó visitado, um agente pró-ativo em avanço tem uma probabilidade de 10% de ser enviado em *broadcast*, como forma de manter o caráter exploratório da busca.

Os autores realizam experimentos comparando o *AntHocNet* com o conhecido AODV (*Ad Hoc On-Demand Distance Vector*) (PERKINS e ROYER, 1999), obtendo resultados superiores em termos de número de pacotes entregues, atraso na rede e variação estocástica dos atrasos.

3.3 Protocolos para Redes de Sensores sem Fio

No caso das RSSF, o problema de roteamento passa a ser mais crítico, levando em consideração que é necessário racionar ao máximo o consumo de energia. Os protocolos devem ser de baixa complexidade computacional, de forma a minimizar o consumo do processador e principalmente evitar o excesso de comunicação de rádio, que é o principal fator de consumo de energia em uma rede de sensores. Apesar de não desprezíveis, fatores como, atraso de pacotes e porcentagem de pacotes entregues, passam a ser secundários em relação à vida útil da rede dependendo das necessidades da aplicação.

Um ponto que pode tornar os protocolos em RSSF mais simples em relação aos de redes de computadores tradicionais, é o fato de que, na maior parte dos casos, existe apenas um destino para os pacotes: o nó sorvedouro. Dessa forma, é necessário que cada nó mantenha apenas uma entrada em sua tabela de roteamento, que é a rota para chegar ao nó sorvedouro. O isomorfismo entre uma rede de sensores sem fio e os elementos do ACO passa a ser mapeado como uma região com vários ninhos de formigas e, aqui, com apenas uma

fonte de alimento. Cada ninho tem seus agentes específicos, porém, todas as trilhas de feromônio podem apontar para um único destino, o que é vantajoso do ponto de vista cooperativo, já que a partir de que os agentes do primeiro nó encontram o nó sorvedouro, todos os outros agentes podem seguir as mesmas trilhas para encontrar o destino.

Zhang, Kuhn e Fromherz (2004) definiram três variantes de ACO para obter melhor desempenho em redes de sensores dinâmicas e com alta densidade. Partindo do algoritmo *AntNet*, são propostas as seguintes variantes: roteamento direcionado, inundação de agentes em avanço e inundação de agentes de dados. As variantes aqui descritas não são aplicadas no protocolo proposto por este trabalho.

3.3.1 Roteamento direcionado

Zhang, Kuhn e Fromherz (2004) afirmam que no início do processo de busca, os agentes podem demorar muito para descobrir o destino. Dessa forma, a rede fica por muito tempo sem referência, até que o primeiro agente encontre o destino e percorra o caminho até a origem. O algoritmo de roteamento direcionado estabelece um gradiente inicial de roteamento em direção ao nó sorvedouro. Os primeiros agentes são guiados com uma maior tendência para os vizinhos mais próximos do sorvedouro, tornando a busca inicial mais rápida. O algoritmo pode ser descrito resumidamente a seguir:

- a) Na inicialização da rede, cada nó estima sua distância até o sorvedouro. Essa distância pode ser a distância geográfica, caso os nós possuam sistema de localização, ou em nós, caso contrário. No caso da ausência de sistemas de localização, o nó sorvedouro propaga uma mensagem de inicialização em *broadcast*, com o valor inicial zero.
- b) Cada nó que recebe a mensagem salva esse valor como a distância estimada, incrementa o valor e efetua novo *broadcast* da mensagem. Somente a primeira mensagem recebida é considerada.
- c) Tendo cada nó armazenado sua distância em relação ao sorvedouro, essa distância é interpretada como o custo local Q_k , ou seja, nós mais próximos do sorvedouro possuem um custo mais baixo para chegar ao destino do que os mais distantes.
- d) Cada nó envia uma mensagem *hello* em *broadcast* para todos os seus vizinhos. Essa mensagem contém o custo Q_k daquele nó. As mensagens *hello* servem apenas para a

descoberta dos vizinhos e estabelecimento dos gradientes iniciais. As mensagens agentes para descoberta de rotas são trocadas posteriormente a essa fase.

- e) Ao receber as mensagens *hello* de seus vizinhos, cada nó k executa o seguinte pseudocódigo:

```

recebida mensagem hello  $i$  do nó  $u$ 
     $Q_u = i.custo$ 
     $i.custo = \min_{n \in N} (Q_u + Q_n)$ 
    envia  $i$  em broadcast
fim

```

onde N é o conjunto de vizinhos de k . Esse código recebe os custos de cada um dos nós vizinhos e envia para os próximos vizinhos a soma entre o custo local e o menor dos custos dos vizinhos de k , fazendo com que sejam criados caminhos de custo decrescente do destino até as bordas da rede

- f) Após todos os nós terem recebido as mensagens *hello*, cada nó pode começar a criar seus agentes. Inicialmente, enquanto não existem trilhas de feromônio, os nós são guiados pelos gradientes estabelecidos na inicialização. A probabilidade de que o nó k escolha um vizinho n para enviar seu agente é dada pela expressão (6).

$$p_n = \frac{e^{(Q_k - Q_n)^\beta}}{\sum_{i \in N} e^{(Q_k - Q_i)^\beta}} \quad (6)$$

Na expressão (6), o expoente β é maior que 1 e tem como objetivo acelerar a convergência do método, enfatizando ainda mais os caminhos com menor custo, quanto maior for seu valor.

Pela expressão, é possível verificar que a tendência inicial é que os agentes sigam pelos caminhos de menor custo. Tornando mais rápida a descoberta inicial do nó sorvedouro.

3.3.2 Inundação de agentes em avanço

Essa estratégia é a mesma utilizada pelo *AntHocNet* (DI CARO, DUCATELLE e GAMBARDELLA, 2005), com seus agentes reativos em avanço. A idéia é enviar os agentes inicialmente em *broadcast*, multiplicando-os pela rede até que o nó sorvedouro seja encontrado. Dessa forma, múltiplos agentes em retorno são criados e múltiplos caminhos são marcados com feromônio, de forma que a busca inicial pode se tornar bastante rápida. Zhang, Kuhn e Fromherz (2004) ainda citam que esta variante pode causar um excesso de colisões e atrapalhar o tráfego da rede, sendo necessário calibrar adequadamente o intervalo entre as criações de agentes.

3.3.3 Inundação de agentes de dados

Nesta variante, os pacotes de dados são combinados com os agentes de busca de rotas. Zhang, Kuhn e Fromherz (2004) citam que esta estratégia funciona bem quando a rede é altamente dinâmica. Os pacotes de dados enviados na rede são roteados até o destino guardando a lista de nós visitados e lá geram os agentes de retorno, que atualizam as tabelas de feromônio na rede.

Os autores fizeram os experimentos em um cenário de 49 nós, possibilitando mobilidade. Uma comparação é realizada entre os 3 protocolos propostos e o *AntNet* básico. Os resultados mostram que o roteamento direcionado é mais eficiente em energia, a inundação de agentes em avanço tem o menor atraso e a inundação de agentes de dados obtém a maior taxa de entrega de pacotes.

3.3.4 Protocolos com ênfase na energia

Camilo, Carreto, Silva e outros (2006) definiram um protocolo eficiente em energia baseado em colônia de formigas para RSSF. A principal preocupação dos autores é com o tamanho das mensagens agentes: como cada agente carrega uma lista com os nós já visitados, o tamanho da mensagem pode ficar muito grande, principalmente em redes com muitos nós. A transmissão de mensagens muito grandes devido a grandes listas de nós é o principal fator do excesso de consumo de energia em RSSF.

Os autores propõem duas variantes do ACO para RSSF: a primeira é o IABR (*Improved Ant-Based Routing*). Este protocolo propõe que a deposição de feromônio pelos agentes deve ser dependente não da distância entre origem e destino, mas do nível de energia dos nós presentes na rota. A função utilizada para a atualização das tabelas de feromônio é mostrada pela equação (7).

$$\Delta T_k = \frac{1}{C - (Avg(E_k) - 1/\min(E_k))} \quad (7)$$

onde: C = quantidade de energia correspondente à bateria cheia

E_k = vetor carregado pelo agente com energia de todos os nós percorridos

$Avg(E_k)$ = média do vetor E_k

$\min(E_k)$ = menor valor do vetor E_k

A expressão acima mostra que o protocolo proposto privilegia os caminhos com um melhor nível médio de energia e onde o nó mais fraco da rota - $\min(E_k)$ não está próximo de parar de funcionar, interrompendo o caminho estabelecido.

O segundo protocolo é o EEABR (*Energy-Efficient Ant-Based Routing*), que, conforme mencionado anteriormente, procura reduzir o tamanho das mensagens que representam os agentes. O protocolo funciona da seguinte forma:

- a) A memória carregada por cada mensagem é reduzida a apenas duas entradas: os dois últimos nós visitados. O consumo de energia é reduzido sensivelmente, dado que a lista de nós de um agente pode conter dezenas de nós.
- b) Para que os agentes possam encontrar os caminhos de retorno, cada nó visitado por um agente, guarda na memória uma entrada relacionada com o identificador do agente. São guardados o nó de onde o agente veio e o nó para onde foi enviado
- c) Caso um nó receba um agente que já esteja em sua tabela, o agente é eliminado, de forma a evitar ciclos nos caminhos. Ao ser encaminhado para um próximo nó, o nó atual inicia um temporizador, que elimina a entrada correspondente da tabela, caso o agente não retorne no tempo previsto.

- d) Para possibilitar o roteamento baseado em energia, apenas o valor de $Avg(E_k)$ e $min(E_k)$ são carregados por cada agente, o que possibilita que cada nó ao receber um agente em retorno, possa calcular a expressão (8) e atualizar as tabelas de feromônio.

$$\Delta T_k = \frac{1}{C - \left(\frac{EMin_k - Fd_k}{EAvg_k - Fd_k} \right)} \quad (8)$$

onde: C = quantidade de bateria correspondente à bateria cheia

$EMin_k$ = valor da energia do nó de menor energia visitado pelo agente

$EAvg_k$ = valor médio da energia de todos os nós visitados pelo agente

Fd_k = número de nós visitados pelo agente

Aqui, verifica-se que as tabelas de feromônio são atualizadas com base na energia dos nós visitados e também na distância percorrida, caracterizando o algoritmo como multiobjetivo. Uma definição de otimização multiobjetivo será dada no capítulo 4. A utilização da expressão (8) tenta garantir um compromisso entre o tempo de propagação dos dados (distância) e situação da energia dos nós, evitando passar por regiões de baixa energia e nós que estejam próximos do fim de suas baterias.

Os autores realizam simulações comparando o IABR, o EEABR e o *AntNet* básico, mostrando que tanto o EEABR quanto o IABR consomem menos energia do que o protocolo básico, sendo o EEABR o mais eficiente.

Tanto o IABR como o EEABR são implementados e utilizados para comparação com o protocolo proposto neste trabalho. O IABR se diferencia do protocolo proposto aqui apenas na expressão de atualização das tabelas de feromônio (equação 7). Já o EEABR tem como diferenças a expressão de atualização das tabelas de feromônio (equação 8) e a técnica de redução do tamanho das mensagens. É importante ressaltar aqui que o IABR não é um protocolo multiobjetivo, pois visa otimizar apenas um objetivo.

3.4 Discussão

Verifica-se neste capítulo que a técnica de ACO é largamente utilizada em redes de comunicação de todos os tipos, dada a semelhança de seus elementos com os elementos de uma rede. O fato de ser uma heurística simples, com poucas operações e pouco conhecimento global sobre a rede, possibilita que se implemente o método de forma distribuída nos nós sensores. Cada nó executa um conjunto de operações bastante simples, que tem baixo impacto no processamento da aplicação e necessita pouca memória para guardar as variáveis do protocolo, o que é bastante interessante, visto que os nós sensores possuem recursos limitados. Implementações centralizadas, onde os nós têm que enviar as informações de roteamento para um elemento central realizar o cálculo, podem se tornar inviáveis por gerar um alto tráfego na rede, em especial para redes com alta densidade e alto número de nós.

Outro aspecto importante é o fato do método de Otimização por Colônia de Formigas (ACO) poder ser calibrado para melhorar o consumo de energia da rede, privilegiando os caminhos com alto nível de energia e evitando rotas com nós próximos do fim das baterias. Essa estratégia será explorada no próximo capítulo, onde será proposto um protocolo multiobjetivo desenvolvido com base em ACO para redes de sensores sem fio.

4 ANTSENSOR: ROTEAMENTO MULTIOBJETIVO EM RSSF BASEADO EM OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS

4.1 Introdução

Neste capítulo é apresentada a metodologia utilizada para o desenvolvimento de um protocolo multiobjetivo baseado em Otimização por Colônia de Formigas denominado *AntSensor* o qual é aplicado ao problema de roteamento em Redes de Sensores sem Fio. Serão descritos os objetivos do protocolo, o algoritmo desenvolvido, suas vantagens em relação a outros protocolos, e uma tabela comparativa de características entre o protocolo proposto e outros protocolos, descritos nos capítulos 2 e 3.

O objetivo geral do protocolo aqui descrito é estabelecer as rotas de encaminhamento de pacotes de cada nó sensor até o nó sorvedouro de uma Rede de Sensores sem Fio. Ao final de cada ciclo do protocolo, cada nó sensor da rede deve ter armazenado o conjunto de nós que um pacote de dados deve seguir a partir dele para chegar ao nó sorvedouro. Na medida do possível, o protocolo deve tentar obter rotas com as seguintes características:

- a) As rotas devem ter o menor número de nós possível.
- b) As rotas devem ter o maior nível de energia possível no nó mais fraco, ou seja: considerando E_k o vetor que contém os níveis de energia de todos os nós de uma rota k , o protocolo deve obter rotas com o maior valor possível de $\min(E_k)$.

Como discutido em (SILVA, MUNARETTO e DELGADO, 2007), rotas com poucos nós possibilitam uma transmissão mais rápida da informação (menor atraso na rede) e consomem energia de poucos nós. Rotas com um alto nível de energia no seu nó mais fraco tendem a levar mais tempo para ser interrompidas devido à falência de um de seus nós. O objetivo final é a economia de energia nas transmissões e um conseqüente aumento da vida útil da rede.

Por se tratar de um protocolo multiobjetivo, faz-se necessário avaliar o *AntSensor* utilizando-se as técnicas de otimização multiobjetivo. A próxima seção descreve os principais critérios que devem ser avaliados na análise de um processo de otimização multiobjetivo.

4.2 Otimização Multiobjetivo

Seja $f = (f_1, f_2, \dots, f_m), (m \geq 2)$, um vetor de objetivos e $\Omega \subset \mathbb{R}^n$, um subconjunto de decisões factíveis sobre o qual f está definida (ZELENY, 1982, CHANKONG, 1983 e FERREIRA, 1998). O problema multiobjetivo de maximização pode ser definido como segue:

$$\text{maximizar } f(x) \text{ s.a. } x \in \Omega$$

Em geral, máximos globais não podem ser obtidos. Note que $f : \Omega \rightarrow y \subset \mathbb{R}^m$ e y é parcialmente ordenado: em geral, qualquer que seja $\varepsilon > 0$,

$$\nexists x^* \in \Omega : f(x^*) \geq f(x) \forall x \in \Omega \cap \text{vizinhança}_\varepsilon(x^*)$$

A expressão acima ressalta que não existe nenhum vetor x^* (conjunto de valores para todas as variáveis-objetivo do problema), cujo valor de $f(x^*)$ é maior do que $f(x)$ para qualquer outro vetor x de valores para as variáveis-objetivo. Esse vetor x^* seria definido como máximo global.

4.2.1 Critério de Dominância

Para um ponto qualquer $x^0 \in \mathbb{R}^n$, define-se:

$$\Omega_{>}(x^0) := \{x : f(x) \geq f(x^0), f(x) \neq f(x^0)\}$$

$$\Omega_{\leq}(x^0) := \{x : f(x) \leq f(x^0)\}$$

$$\Omega_{\sim}(x^0) := \{x : f(x) \not\geq f(x^0), f(x) \not\leq f(x^0)\}$$

Onde $\mathbb{R}^n : \Omega_{>}(x^0) \cup \Omega_{\leq}(x^0) \cup \Omega_{\sim}(x^0)$, sendo que $\Omega_{>}(x^0)$ e $\Omega_{\leq}(x^0)$ contêm os pontos que dominam e são dominados por x^0 (respectivamente), segundo a definição de $\geq$;

e $\Omega_{\sim}(x^0)$ contém o conjunto de soluções não comparáveis a x^0 ($f(x)$ não é menor nem maior do que $f(x^0)$ – soluções que são melhores que x^0 em alguns objetivos e piores em outros).

4.2.2 Solução Eficiente (ou solução não-dominada)

De acordo com (FERREIRA, 1998) uma solução eficiente (também denominada de solução não-dominada, não-inferior ou Pareto-ótima) pode ser definida como segue:

$x^* \in \Omega$ é chamada de solução eficiente se não existir qualquer outro ponto $x \in \Omega$ tal que $f(x) \geq f(x^*)$, $f(x) \neq f(x^*)$.

Portanto as principais características das soluções de Pareto (soluções não-dominadas) para o problema de maximização tratado neste trabalho são:

- Para ser considerada uma solução eficiente, $x^* \in \Omega$ deve atender à condição de inexistência de qualquer outro ponto $x \in \Omega$ tal que $f(x) \geq f(x^*)$, $f(x) \neq f(x^*)$, ou seja, as soluções não são dominadas por quaisquer outros pontos de Ω ;
- Qualquer ponto $x \in \Omega$ que proporcione um acréscimo que algum objetivo, deve levar ao decréscimo de pelo menos algum outro, relativamente aos valores produzidos por uma solução de Pareto;
- Em geral existem infinitas soluções de Pareto (não-dominadas) em Ω representadas pelo conjunto $efi(\Omega)$.

Como em grande parte das situações estes objetivos são conflitantes (como será mostrado na seção 5.3.2), o sistema deve ser capaz de produzir uma solução que privilegie um ou outro objetivo (conforme definição dos pesos) ou ainda alguma solução de compromisso entre os dois objetivos no caso de uma ponderação adequada das funções-objetivo.

4.2.3 Método das ponderações

Sejam $f_1, f_2, \dots, f_m$ funções convexas sobre Ω , a Figura 21 ilustra a interpretação do conjunto $efi(\Omega)$ quando são considerados 2 objetivos.

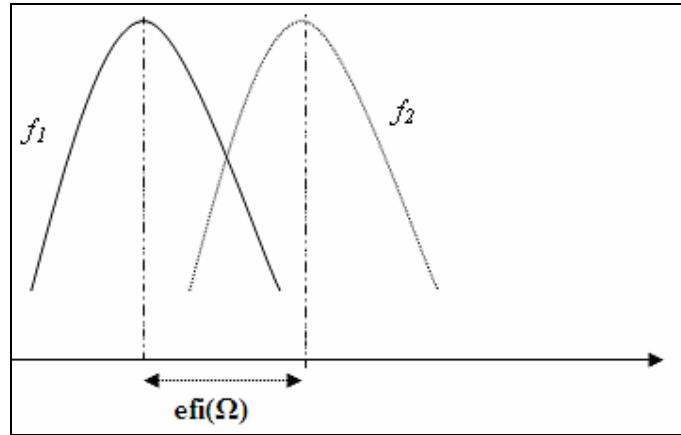


Figura 21: Conjunto de soluções eficientes considerando-se duas funções objetivo convexas (FERREIRA, 1998)

Conforme discutido em (FERREIRA, 1998), neste caso, o conjunto $efi(\Omega)$ pode ser gerado resolvendo-se o problema:

$$P_w := \max_{x \in \Omega} \langle w, f(x) \rangle$$

$$\text{varrendo-se o vetor } w \text{ em } W := \{w : w \geq 0, \sum_{i=1}^m w_i = 1\}.$$

Na prática, a geração de $efi(\Omega)$ por P_w (ou por qualquer outro método) é inviável. A geração de subconjuntos de $efi(\Omega)$, caracterizados por informações (preferências) contidas em $w \in W$ é bastante usual.

Neste trabalho, inspirou-se no método das ponderações, para se avaliar um conjunto de soluções do problema mesmo no caso da não garantia da convexidade das funções-objetivo. Conforme será mostrado na seção 5.3.2, o método que busca a análise de soluções não-dominadas, consiste em variar os pesos das funções objetivo de forma complementar e produzir um conjunto de soluções a serem entregues ao usuário para que o mesmo possa

escolher aquela que apresentar a melhor relação custo-benefício para o problema considerado. Embora não haja garantia de obtenção do conjunto $efi(\Omega)$, as simulações mostram que grande parte das soluções produzidas é não dominada.

Os critérios de otimização multiobjetivo definidos aqui serão utilizados posteriormente para avaliar o protocolo *AntSensor*, que será descrito em detalhes na próxima seção.

4.2.4 Exemplo prático

Segundo Faccioli (2007), resolver um problema de otimização multiobjetivo implica considerar vários aspectos visando encontrar a melhor solução. Pode haver exemplos em que considerando apenas os resultados quantitativos não é possível determinar qual é a melhor solução. Toma-se como exemplo a tomada de decisão na compra de um carro, onde podem existir dois objetivos: minimizar o preço e maximizar o conforto. A Figura 22 ilustra um exemplo de opções de escolha:

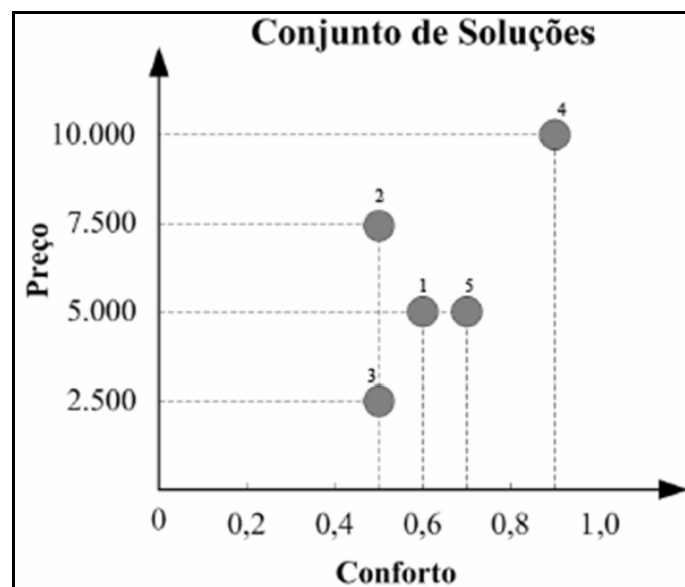


Figura 22: Conjunto de soluções possíveis para compra de um carro (FACCIOLI, 2007)

Percebe-se que as soluções 1 e 2 podem ser descartadas, pois a solução 3 pode obter o mesmo conforto com um preço menor que a solução 2 e a solução 5 pode obter o mesmo preço da solução 1, com um maior conforto. As soluções 3, 4 e 5 são então as melhores alternativas de compra. Porém, não é possível só com os critérios quantitativos definir qual é

a melhor delas (FACCIOLI, 2007). Pode se atribuir um compromisso entre as soluções, se é sabido que, por exemplo, um maior conforto implica em um maior preço, no caso de objetivos conflitantes.

Não tendo mais informações sobre as soluções, pode-se afirmar que as soluções 3, 4 e 5 são as soluções ótimas para este problema, representando o *conjunto não dominado* (FACCIOLI, 2007). Logo, as soluções 1 e 2 formam o *conjunto dominado*.

O conjunto das soluções não dominadas pode ser representado no espaço cartesiano e formam a chamada *Fronteira de Pareto* (FACCIOLI, 2007). As *soluções Pareto Ótimas* formam o conjunto de soluções não dominadas em relação a todas as outras soluções possíveis para o problema. Cada uma destas soluções é melhor que todas as outras possíveis em pelo menos um dos objetivos do problema.

4.3 Descrição do Algoritmo

O algoritmo do *AntSensor* é executado em cada um dos nós de uma RSSF, beneficiando-se das características de cooperação do ACO para obter os resultados desejados.

Para o projeto do algoritmo, assumem-se as seguintes características para a RSSF de interesse:

- Os N nós da rede estão espalhados aleatoriamente, com distribuição uniforme, sobre uma área plana e quadrada de $L \times L$ m², conforme mostra a Figura 23;
- Todos os nós são inicializados com uma quantidade de $E_{\max}$ Joules de energia;
- O consumo do rádio de cada nó é de E_{bit} nJ por bit transmitido ou recebido;
- O rádio de cada nó tem um alcance fixo de A m. Caso um nó transmita um pacote, todos os nós vizinhos que estejam a uma distância menor ou igual ao alcance do rádio recebem o pacote;
- O nó sorvedouro está localizado em uma posição aleatória da rede.

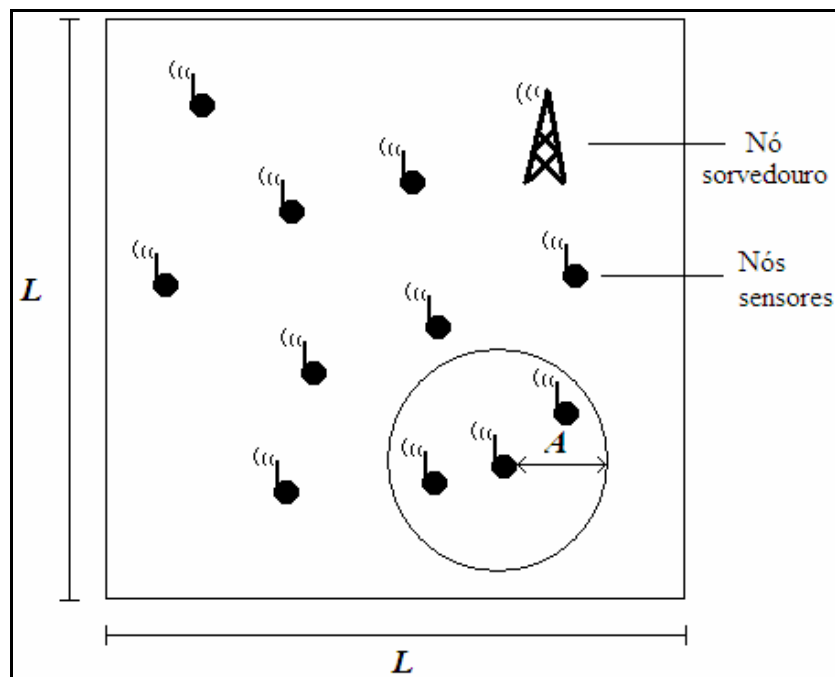


Figura 23: Cenário de Implementação

No algoritmo *AntSensor* os nós possuem dois estados possíveis: *otimização*, onde os nós estabelecem as rotas até o nó sorvedouro através do protocolo proposto e *dados*, onde as rotas estabelecidas são utilizadas para a comunicação de dados da aplicação. A ideia geral do algoritmo assume que após a inicialização, os passos da otimização são executados continuamente até que todos os agentes vão até o sorvedouro (definição de grafos através das mensagens *hello*) e retornem (marcando as trilhas com feromônio) e o tempo estabelecido da fase de otimização termine. Com o término da fase de otimização, inicia-se a fase de dados, onde os nós utilizam as rotas obtidas até então para enviar dados da aplicação para o nó sorvedouro. Passado o tempo pré-estabelecido da fase de dados, inicia-se uma nova fase de otimização, agora com novas condições de topologia e energia nos nós. Neste momento, é possível ter nós fora de operação, nós movidos de posição e até mesmo novos nós incluídos na rede. O algoritmo irá então estabelecer o novo grafo e procurar novas rotas para serem utilizadas pelos nós, num processo adaptativo. O algoritmo proposto é descrito detalhadamente a seguir:

Passo 1: Inicialização -

Passo 1.1 - Distribuem-se os N nós na área de monitoramento e os nós são ativados.

Passo 1.2 - Cada um dos nós da rede sabe *a priori* seu identificador, que é um número único, no intervalo $[1;N]$.

Passo 1.3 - O nó sorvedouro é identificado com o identificador de número N .

Passo 2: Otimização –

enquanto ($T < T_o^1$)

Passo 2.1 – Definição dos grafos de cada nó origem –

Passo 2.1.1 - No início da fase de otimização, cada nó envia em *broadcast* algumas mensagens *hello*, anunciando-se aos seus vizinhos. Essas mensagens contêm o identificador do nó. Ao receber uma mensagem *hello*, um nó qualquer adiciona o nó originário da mensagem em sua lista de vizinhos na memória. Estabelecendo um paralelo para a notação de grafos, esta etapa equivale a estabelecer os arcos do grafo que representa a rede, conforme ilustra a Figura 24.

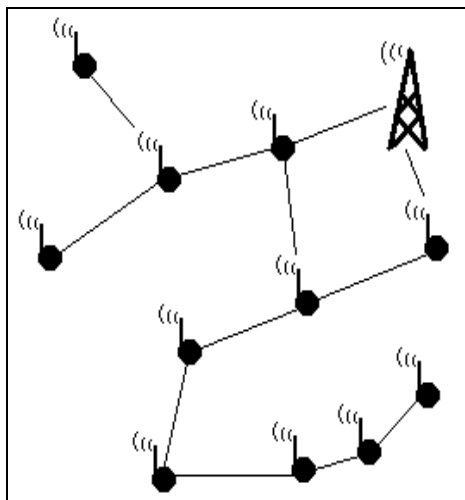


Figura 24: Estabelecimento do grafo a partir das mensagens hello

Passo 2.1.2 - A lista de vizinhos de cada nó contém uma entrada para cada vizinho, onde é guardado o identificador do nó e a intensidade da trilha (feromônio) que aponta para aquele vizinho. Inicialmente, a intensidade da

¹To é o tempo de duração da fase de otimização, medido em segundos.

trilha para todos os vizinhos é nula, representando a ausência de feromônio no grafo. A tabela de vizinhos de cada nó é exemplificada pela Tabela 7.

Tabela 7: Tabela de vizinhos de um nó

Vizinho	Identificador	Intensidade da trilha
1	Id_1	τ_1
2	Id_2	τ_2
...	...	...
N	Id_n	τ_n

Passo 2.1.3 - Os nós que recebem *hello* com o identificador do nó sorvedouro, concluem que agora já têm a rota para o mesmo, removendo automaticamente todos os outros nós de sua lista de vizinhos e ignorando futuras mensagens *hello* de outros nós. A partir daqui, tem-se um grafo formado, sobre o qual pode-se utilizar o ACO para descobrir os melhores caminhos entre cada nó e o sorvedouro.

Passo 2.2 – Criando Lista de agentes –

Passo 2.2.1 - Depois de estabelecido o grafo, cada nó cria Num_{ag} mensagens em sua memória. Essas mensagens representam os agentes do ACO. A estrutura de uma mensagem agente é mostrada na Figura 25.

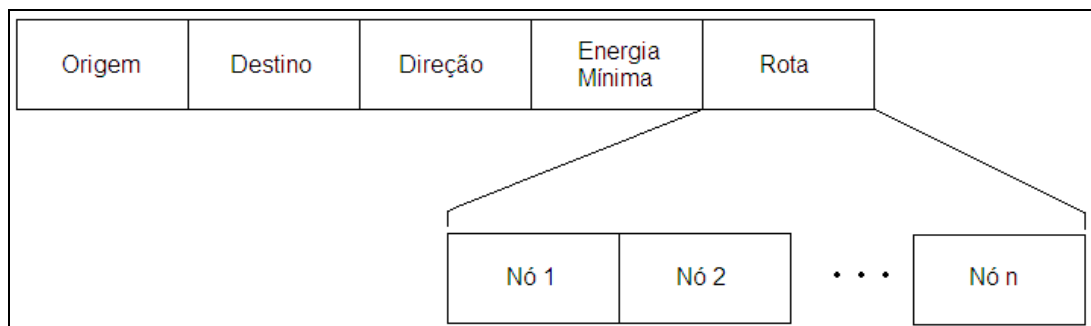


Figura 25: Campos da mensagem agente

onde: origem – nó onde o agente foi originado

destino – próximo nó para onde a mensagem será enviada

direção – agente em avanço ou em retorno

energia mínima – valor mínimo da energia de um nó visitado até o momento

rota – lista de nós visitados. No início do processo, a lista de nós é vazia.

Passo 2.2.2 - Os agentes criados são colocados em uma fila, sendo que cada um dos agentes é enviado a cada pulso de relógio do nó. Para os testes práticos, foi estabelecido um pulso de relógio a cada 1s. Cada nó insere um pequeno erro aleatório de $\pm 10\%$ no intervalo entre pulsos, de forma a evitar que todos os nós enviem mensagens exatamente ao mesmo tempo. Isso é feito com o objetivo de minimizar a interferência e o número de colisões, diminuindo o número de mensagens perdidas no canal de rádio.

Passo 2.2.3 - Futuros agentes recebidos de outros nós são inseridos no fim da fila e seguem o mesmo regime de uma mensagem a cada pulso de relógio. É importante mencionar aqui que uma das vantagens do protocolo proposto é que este não depende de um relógio sincronizado entre todos os nós. Outros protocolos, como o LEACH (HEINZELMAN, CHANDRAKASAN e BALAKRISHNAN, 2000), podem enfrentar sérios problemas de desempenho se os nós entrarem em determinados estados fora de sincronismo.

Passo 2.2.4 - A cada pulso de relógio, cada nó escolhe um vizinho para enviar o agente que está na frente da fila. A probabilidade de seleção de cada vizinho é proporcional à intensidade da trilha no arco que liga aquele vizinho ao nó atual, conforme a equação (1). Depois de feita a seleção, o nó atual insere seu identificador no fim da lista de nós visitados pelo agente, salva também o valor de seu nível de energia e o envia para o nó selecionado.

Passo 2.2.5 - Cada nó que recebe um agente insere seu identificador no fim da lista de nós visitados, substitui o valor de seu nível de energia na mensagem, caso seja menor que o já contido na mesma e insere o agente no fim da sua fila de mensagens, para ser encaminhado para um próximo vizinho.

Passo 2.3 – Movimentação dos agentes –

Passo 2.3.1 - Os agentes se movimentam inicialmente pela rede (movimento origem – sorvedouro) através da repetição dos passos 2.2.4 e 2.2.5. No início, como não existem trilhas depositadas, o movimento é totalmente aleatório, até que um dos agentes chegue ao nó sorvedouro.

Passo 2.3.2 - Quando qualquer agente chega ao sorvedouro, este verifica a rota percorrida pelo agente, remove os ciclos que por acaso existam, e reenvia o agente pela mesma rota, de volta ao nó de origem. Os ciclos são removidos no sorvedouro por este possuir maior capacidade de processamento e energia.

Passo 2.3.3 - O agente em retorno atualiza as tabelas de trilha (feromônio) em cada nó visitado no retorno para a origem. A expressão utilizada aqui leva em consideração tanto o número de nós da rota percorrida, como a energia do nó mais fraco da mesma, conforme mostra a equação (9).

$$\tau_i = \frac{\frac{k_E \times E_i}{E_{\max}} + \frac{k_d}{d}}{2} \quad (9)$$

Onde: k_E = constante de ponderação para a Energia.

E_i = energia do nó mais fraco da rota

$E_{\max}$ = máxima energia que um nó pode ter (bateria cheia)

k_d = constante de ponderação para o comprimento da rota em nós

d = distância em nós total da rota

A expressão utilizada privilegia rotas com menor número de nós e que estejam mais longe de serem interrompidas. Nota-se que se k_E e k_d forem iguais a 1, a expressão obtém o valor máximo quando o número de nós da rota é 1 (vizinho do sorvedouro) e todos os nós da rota estão com as baterias cheias. Deve ser ressaltado o fato de que a expressão 9 é o fator que inclui a característica multiobjetivo no método proposto. É essa expressão que irá direcionar a busca de soluções influenciada pelos dois objetivos que devem ser otimizados pelo protocolo definido aqui.

Passo 2.3.4 - Ao chegar à origem, o agente é eliminado. O nó de origem então verifica se a rota obtida é melhor que as obtidas anteriormente, através do cálculo da expressão (9). A rota que obtiver o maior valor é armazenada para uso durante a fase de dados.

Passo 2.3.5 –A cada pulso de relógio, a tabela de feromônio é atualizada através da evaporação, conforme definido na seção 3.1– equação 3.

fim enquanto

Passo 3: Envio de Dados –

enquanto ($T < T_D^2$)

Na fase de dados, eventos são gerados a cada pulso de relógio, com uma probabilidade P_{evento} . Caso um nó qualquer observe um evento, este envia uma mensagem de dados para o nó sorvedouro, utilizando a rota obtida durante a fase de otimização. Essa mensagem contém o identificador do nó de origem, o valor da grandeza física observada e os nós que devem ser percorridos da fonte ao sorvedouro.

fim enquanto

Passo 4 – Retorna à fase de otimização (Passo 2)

Os passos de 2 a 4 são executados continuamente até a extinção de todos os nós por falta de baterias. A cada novo início de uma fase de otimização, é possível ter novas condições de energia e posicionamento dos nós. O algoritmo irá então estabelecer o novo grafo, através das mensagens *hello*, e procurar novas rotas para serem utilizadas pelos nós na nova topologia, num processo adaptativo.

O algoritmo aqui definido atua como uma busca global de soluções, ou seja, efetua uma busca em todo o espaço de soluções (rotas) possíveis. A função multiobjetivo pode não ser monotônica, de forma que existem diversos máximos locais em regiões diferentes do espaço de busca. Um algoritmo de busca global deve explorar todo o espaço de busca o mais uniformemente possível, diferentemente de outros algoritmos de otimização iterativos que iniciam em um ponto e fazem uma busca local em direção ao máximo mais próximo (máximo local).

Uma versão aprimorada do algoritmo, inspirada em um algoritmo de busca local foi proposta para aumentar a velocidade de convergência do método. A diferença dessa abordagem está na etapa de seleção (passo 2.2.4), onde a escolha dos possíveis vizinhos privilegia aqueles com as melhores rotas até o sorvedouro, encontradas até o momento. Essa versão do algoritmo é descrita na próxima seção

² T_D é o tempo de duração da fase de dados, medido em segundos.

4.4 Abordagem alternativa – Seleção inspirada no método GRASP de busca local

Com o objetivo de aprimorar a busca, é possível utilizar um algoritmo que aumente a velocidade de convergência das soluções com o intuito de tentar obter uma solução final de melhor qualidade. Um algoritmo de busca local tem como função explorar uma região restrita do espaço de soluções possíveis. Em termos de redes, a busca local significa analisar as rotas obtidas pelos agentes e testar soluções próximas às aquelas obtidas até o momento, verificando se os resultados são superiores.

O algoritmo para a seleção de vizinhos aqui utilizado é inspirado no GRASP (*Greedy Randomized Adaptive Search Procedure*) definido em (RESENDE e PISTOULIS, 2002). O algoritmo GRASP toma uma solução obtida por um processo combinatório e testa uma modificação desta solução em uma vizinhança próxima. Por exemplo: se o problema considerado for a obtenção de um vetor de três números binários, uma solução possível poderia ser $x = (0,1,0)$. A vizinhança de um movimento dessa solução é o conjunto formado pelas soluções onde há a mudança de apenas um elemento, ou seja: $x = \{(0,0,0), (1,1,0), (0,1,1)\}$. Dependendo do problema, uma ou mais das soluções vizinhas podem ser melhores que a original.

O GRASP propõe que a cada iteração seja realizado um único movimento, escolhido aleatoriamente dentro de um espaço de $\alpha \times R$ soluções, onde R é o número total de soluções vizinhas de um movimento da solução atual e α é um número qualquer no intervalo $[0;1]$. Esse fator representa o quão guloso é o algoritmo: quanto menor o valor de α , mais rápida é a convergência do método e menor o caráter exploratório do algoritmo.

A aplicação inspirada no GRASP para o algoritmo de formigas proposto se insere no momento da seleção dos vizinhos para a movimentação dos agentes – seção 4.3, passo 2.2.4, equação (1). Com a utilização do GRASP, cada agente executa a seleção do próximo vizinho considerando apenas os $\alpha \times Num_{vizinhos}$ vizinhos com maior intensidade de trilha em sua direção. Portanto, na abordagem proposta neste trabalho, o algoritmo GRASP foi adaptado para o processo de seleção dos nós vizinhos, onde apenas aqueles nós que apresentem boas rotas do ponto atual até o sorvedouro sejam candidatos. Isso equivale a explorar o espaço de busca de forma mais gulosa, privilegiando mais ainda os caminhos que já se apresentaram melhores soluções no passado. Vale ressaltar aqui, que utilizar um coeficiente $\alpha = 1$ equivale a não utilizar o método GRASP para a seleção, visto que todos os vizinhos serão considerados para a escolha.

4.5 Características do *AntSensor* frente a outros protocolos da literatura

Conforme citado no Capítulo 3, a Tabela 8 mostra um comparativo de características entre todos os protocolos descritos neste trabalho:

AntSensor: proposta apresentada neste trabalho;

Difusão direcionada (INTANAGONWIWAT, GOVINDAN e ESTRIN, 2000);

SPIN (HEINZELMAN, KULIK e BALAKRISHNAN, 1999);

MULTI (FIGUEIREDO, NAKAMURA e LOUREIRO, 2004);

LEACH (HEINZELMAN, CHANDRAKASAN e BALAKRISHNAN, 2000);

SHARP (RAMASUBRAMANIAN, HAAS e SIRER, 2003);

PEGASIS (LINDSEY e RAGAVENDRA, 2002);

ICA (HABIB, CAMARA e LOUREIRO, 2004);

GPSR (KARP e KUNG, 2000)

AntNet (DI CARO e DORIGO, 1998);

AntHocNet (DI CARO, DUCATELLE e GAMBARDELLA, 2005);

IABR (CAMILO, CARRETO, SILVA *et al*, 2006);

EEABR (CAMILO, CARRETO, SILVA *et al*, 2006).

Com base na tabela, é possível definir quais protocolos são mais vantajosos para cada tipo de aplicação desejada. No caso do *AntSensor*, verifica-se então que este tenderá a obter melhores resultados quando aplicado a redes planas e com tráfego de dados elevado, por estabelecer as rotas pró-ativamente, antes da ocorrência dos eventos de dados.

Tabela 8: Comparativo entre protocolos

Protocolo	Multiobjetivo	Baseado em Formigas	Plano	Hierárquico	Geográfico	Reativo	Pró-Ativo
<i>AntSensor</i>	X	X	X				X
Difusão Direcionada			X			X	
SPIN			X			X	
MULTI			X			X	X
LEACH				X			X
SHARP				X			X
PEGASIS				X			X

ICA					X		X
GPSR					X	X	
<i>AntNet</i>		X				X	
<i>AntHocNet</i>		X					X
IABR		X					X
EEABR	X	X	X				X

4.6 Discussão

Este capítulo apresentou o protocolo de roteamento proposto *AntSensor* o qual foi baseado em Otimização por Colônia de Formigas e empregado na otimização de rotas em redes de sensores sem fio. As principais vantagens do protocolo criado são: possibilidade de obtenção de rotas curtas e com boa vida útil; adaptabilidade e tolerância a falhas, visto que novas rotas são criadas periodicamente e implementação distribuída, com base apenas em informações da vizinhança. A escalabilidade é uma outra importante vantagem, uma vez que o algoritmo não precisa ser modificado com o aumento do número de nós da rede. Mensagens em *broadcast* são utilizadas apenas no início, para formar o grafo da rede. Isso evita uma inundação excessiva da rede, o que causaria alta interferência e perda de pacotes, com um conseqüente aumento de consumo em retransmissões. Assim, acredita-se que o método possui todas as características necessárias para um protocolo de roteamento para RSSF.

Como se trata de um protocolo pró-ativo, este deverá ser mais eficiente em situações de redes com grande número de nós e com monitoramento periódico de dados, aproveitando as rotas já criadas para a disseminação dos dados ao invés de gerar um grande atraso para criar as rotas reativamente.

O principal diferencial da abordagem aqui apresentada está na forma como são considerados os dois objetivos no processo de otimização multiobjetivo: uma formiga (agente) que retorna do nó sorvedouro para a origem deposita nos arcos do grafo que representa a rede certa quantidade de feromônio. Essa quantidade é definida por uma nova expressão que considera tanto o número de nós da rota percorrida, quanto a energia do nó mais fraco da rota. A maior parte dos métodos que utilizam ACO para redes de computadores considera apenas o número de nós nessa expressão. Isso se deve ao fato de que a maioria das redes existentes não possui restrições tão fortes ao consumo de energia como as RSSF.

O próximo capítulo discutirá os resultados de simulação obtidos e as comparações com outros métodos, como forma de validar a eficiência do protocolo desenvolvido e observar também o funcionamento prático de outros protocolos.

As comparações serão realizadas entre os seguintes protocolos:

- *AntSensor* proposto por este trabalho,
- EF-Tree por se tratar de um protocolo bastante difundido nas aplicações existentes dada sua simplicidade de implementação,
- IABR e o EEABR, por serem protocolos baseados em colônia de formigas além de serem específicos para redes de sensores sem fio, com características semelhantes ao protocolo proposto.

No caso do EEABR, por ser também um protocolo multiobjetivo, sua comparação com o *AntSensor* proposto é fundamental.

5 SIMULAÇÕES E RESULTADOS

Este capítulo descreve os experimentos realizados visando validar o protocolo proposto no capítulo anterior. Serão descritos aqui os materiais e ferramentas utilizadas para a simulação do protocolo *AntSensor* e os diversos cenários de teste a que este foi submetido. Serão apresentados resultados da avaliação isolada do protocolo e também em comparação com outros protocolos descritos no capítulo 2. Ao final do capítulo será apresentada uma discussão sobre os resultados obtidos e o desempenho do protocolo proposto.

5.1 Simulador

O ambiente para simulação foi implementado utilizando-se o simulador *Prowler*, pacote de expansão para o MATLAB, o qual foi proposto por (SIMON, VOLGYESI, MARÓTI *et al*, 2003). O pacote permite simular uma rede de sensores sem fio com uma topologia determinada pelo usuário. Esse pacote simula a camada física, como um canal de rádio com desvanecimento em larga escala (propagação do sinal) e pequena escala (efeito *Rayleigh*). A camada de enlace é simulada utilizando uma versão simplificada do protocolo CSMA/CA. Os parâmetros das camadas física e de enlace podem ser configurados de acordo com as necessidades do usuário. Para a implementação do protocolo *AntSensor*, foram feitas alterações no código da camada de rede (roteamento).

É possível definir aplicações customizadas baseadas em uma interface provida pelo simulador, que é similar ao sistema operacional *TinyOS*, utilizado nas famílias de nós sensores reais. A interface é caracterizada pela implementação das funções executadas por um único nó, na ocorrência de cada um dos eventos de rede possíveis: recebimento e envio de um pacote, pulso de relógio, colisão, etc. Cada uma dessas funções foi modificada de acordo com a necessidade de cada protocolo de roteamento específico e, durante a operação do simulador, o comportamento implementado é executado por todos os nós da mesma forma. Essa foi uma das principais motivações para a escolha do simulador, já que este provê todas as camadas de base e uma interface bastante prática para implementação das funções por nó, sem a necessidade da implementação de todo o controle, temporização e eventos básicos da rede.

O simulador provê também uma interface gráfica que mostra a disposição dos nós na rede, onde é possível habilitar luzes indicadoras de transmissão, recepção e colisão. É possível também traçar setas entre dois nós quaisquer, como forma de marcar as rotas de transmissão

de dados, fator bastante útil para a avaliação qualitativa da aplicação desenvolvida neste trabalho. Um exemplo da tela de simulação do *Prowler* é apresentado na Figura 26.

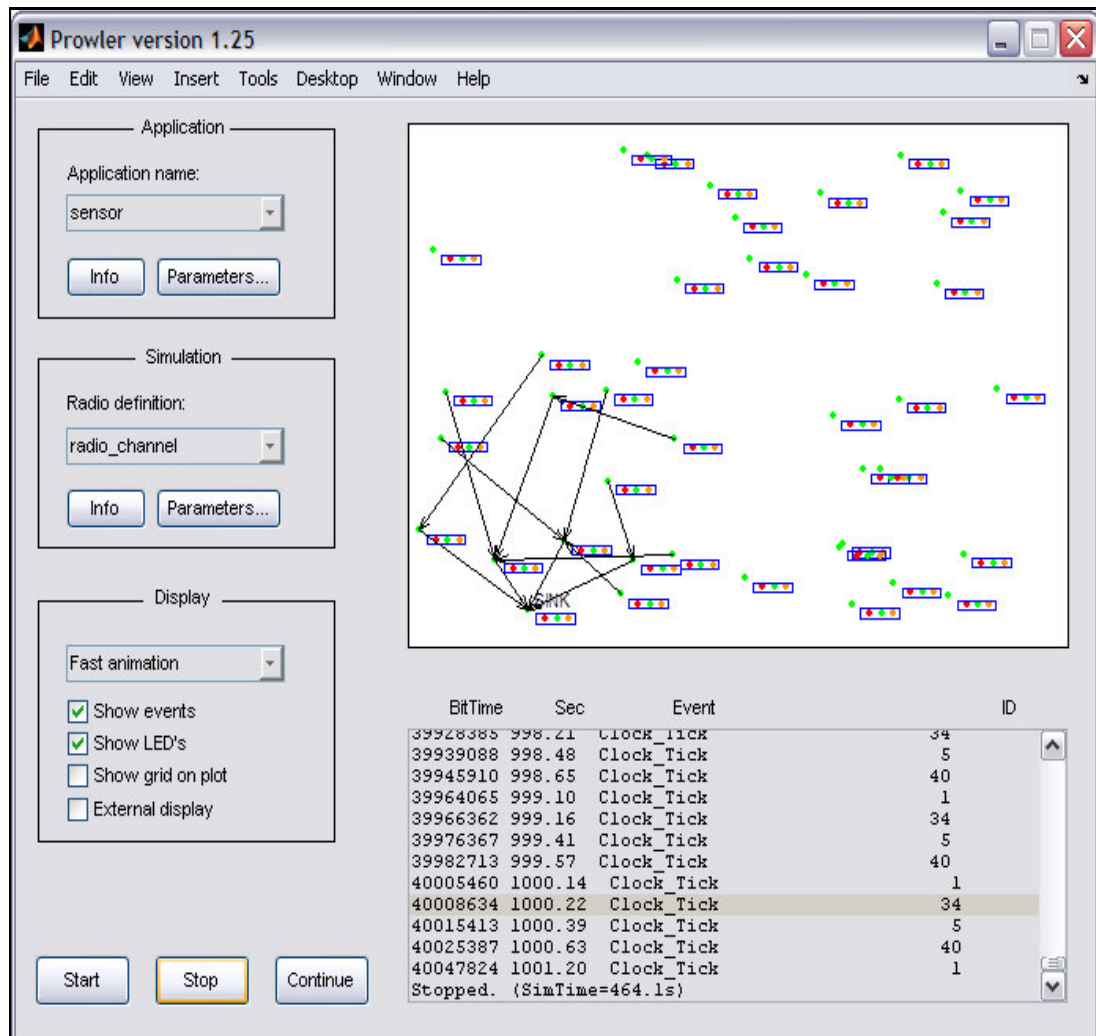


Figura 26: Tela do simulador Prowler

O computador utilizado para as simulações com o *prowler* foi um Notebook com processador Centrino Pentium-M 1,73 GHz e 1,5 GB de memória.

5.2 Parâmetros de configuração

Na implementação do algoritmo de formigas, os parâmetros configuráveis são mostrados na Tabela 9.

Tabela 9: Parâmetros configuráveis na implementação

Parâmetro	Notação	Unidade
Número de Agentes (formigas)	Num_{ag}	Agentes
Fator de evaporação do feromônio	ρ	Adimensional
Dimensões da rede	$L \times L$	m^2
Número de nós da rede	N	Nós
Número de mensagens <i>hello</i> enviadas no início da fase de otimização	Num_{hello}	Mensagens
Energia inicial de cada nó	E_{max}	J (joules)
Alcance do rádio	A	m
Coefficiente do método GRASP	α	Adimensional
Peso da energia no método de otimização	k_E	Adimensional
Peso da distância do método de otimização	k_d	Adimensional
Consumo do rádio	E_{bit}	nJ/bit
Duração da fase de otimização	T_O	s (segundos)
Duração da fase de dados	T_D	s (segundos)
Probabilidade de ocorrência de eventos a cada pulso de relógio na fase de dados	P_{evento}	%

5.3 Experimentos e Resultados

A etapa de experimentos foi dividida em duas fases: avaliação estática e avaliação dinâmica. A avaliação estática foi subdividida em dois tipos: uma baseada em fonte simples, na qual apenas um dos nós da rede pode gerar agentes; e a avaliação baseada em Fronteira de Pareto, que visa obter uma avaliação média, onde todos os nós podem ser nós geradores de agentes. Na avaliação estática, apenas o ciclo de otimização (descrito na seção 4.3) é realizado (não há envio de mensagens de dados). As camadas física e de enlace são ideais no caso da avaliação estática por fonte simples e real no caso da Fronteira de Pareto. No caso da avaliação dinâmica todos os ciclos do algoritmo são considerados e a camadas física e de enlace simulam situações reais.

5.3.1 Avaliação estática – fonte simples

A primeira avaliação realizada com o protocolo proposto foi a avaliação com fonte simples. Nestes testes, apenas um dos nós da rede pode gerar agentes (nó fonte), enquanto os outros nós se comportam apenas como roteadores. O objetivo é observar as rotas obtidas entre o único nó fonte e o nó sorvedouro, inclusive com a variação manual dos pesos de otimização k_E e k_d . Nestas simulações, considera-se apenas a camada de rede, ou seja, as camadas de enlace e física são ideais e livres de colisões, interferências ou erros de transmissão.

5.3.1.1 Rede de pequeno porte – 10 nós

Para uma avaliação local preliminar da eficiência do protocolo, foi realizada uma simulação com uma rede de 10 nós, mostrada na Figura 27. Cada um dos quadrados pretos representa um nó sensor. Os nós estão localizados em posições aleatórias na área de monitoramento e a condição inicial de energia de cada um dos nós está distribuída aleatoriamente no intervalo $[0;80]$ J. Para efeitos de simulação, o nó 8 foi escolhido como fonte e o nó 4 como sorvedouro. Assim, o nó 8 lança os agentes de otimização, que se espalham pela rede, com o objetivo de obter as rotas até o nó 4.

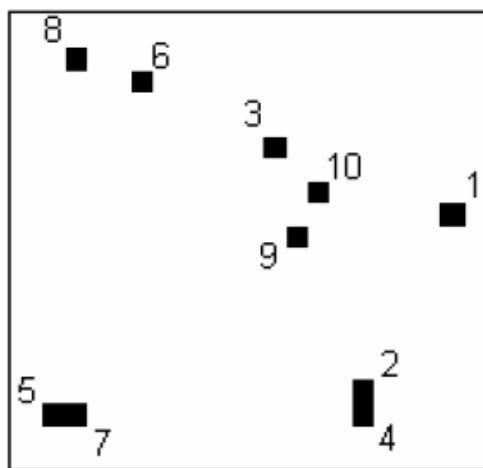


Figura 27: Rede com 10 nós

A Tabela 10 apresenta os parâmetros utilizados para a simulação deste cenário. Neste teste estático, apenas o processo de otimização é considerado, assumindo-se uma condição aleatória de posicionamento e energia dos nós.

Tabela 10: Parâmetros utilizados na simulação de fonte simples

Parâmetro	Valor
Área de monitoramento	20x20 m ²
Número de nós da rede	10 nós
Alcance do rádio de cada sensor	10 m
Energia máxima dos nós (bateria cheia)	80 J
Energia inicial do nó 1	1,877 J
Energia inicial do nó 2	4,590 J
Energia inicial do nó 3	66,919 J
Energia inicial do nó 4	Infinita (nó sorvedouro)
Energia inicial do nó 5	43,345 J
Energia inicial do nó 6	58,926 J
Energia inicial do nó 7	4,906 J
Energia inicial do nó 8	36,770 J
Energia inicial do nó 9	20,088 J
Energia inicial do nó 10	78,828 J

Caso 1: Apenas otimização de distância ($k_E=0$, $k_d=1$, $Num_{ag}=20$, $\rho=0,99$, $E_{bit}=50\text{nJ/bit}$)

A melhor rota obtida é mostrada na Figura 28, passando pelos nós 8-6-9-4. De fato, observando a figura, é possível concluir que se trata de uma das rotas mais curtas que se pode obter neste cenário.

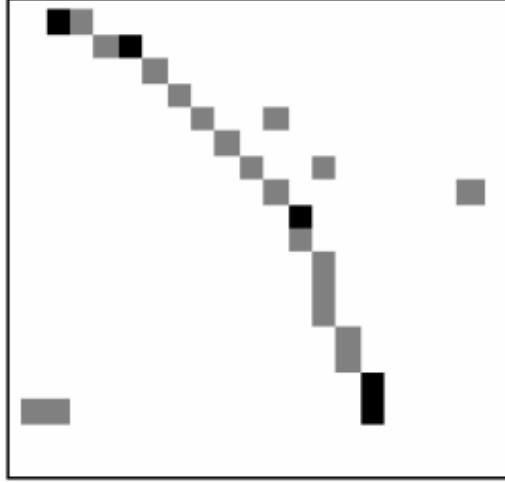


Figura 28: Melhor rota obtida para o Caso 1

Para o Caso 1, o valor da expressão $\tau_i = \frac{\frac{k_E \times E_i}{E_{\max}} + \frac{k_d}{d}}{2}$ para a melhor rota foi 0,125.

Caso 2: Otimização de distância e energia ($k_E=1$, $k_d=1$, $Num_{ag}=20$, $\rho=0,99$, $E_{bit}=50nJ/bit$)

A melhor rota obtida para este experimento foi a mesma do Caso 1, passando pelos nós 8-6-9-4. O nó com menor energia desta rota é o nó 9, com 20,088 J de energia.

Caso 3: ($k_E=1$, $k_d=1$, $Num_{ag}=20$, $E_{bit}=50nJ/bit$, $\rho=0,99$, Energia do nó 1=20,088J, Energia do nó 9=1,877J)

Para ilustrar a otimização de energia, neste caso trocou-se a condição inicial de energia entre o nó 1 e o nó 9, de forma que o nó 9 ficasse com uma quantidade de energia muito baixa. A Figura 29 ilustra a melhor rota obtida para este caso, passando pelos nós (8-3-1-4). Aqui fica evidenciado que o algoritmo priorizou uma rota que não passasse pelo nó 9, que possui uma energia muito baixa, conforme esperado.

Para o caso 3, o valor de $\tau_i = \frac{\frac{k_E \times E_i}{E_{\max}} + \frac{k_d}{d}}{2}$ para a melhor rota foi de 0,2505.

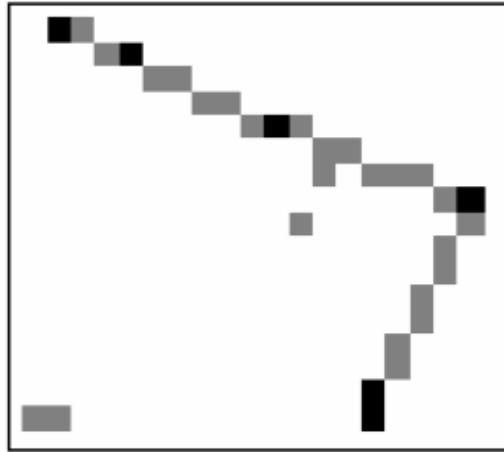


Figura 29: Melhor rota obtida para o caso 3

5.3.1.2 Rede de médio porte – 100 nós

Após os exemplos que mostram que o método prioriza a escolha das rotas de acordo com a influência dos parâmetros da função multiobjetivo, foram realizados testes com uma rede de maior porte, que é o real objetivo do trabalho. Os parâmetros da rede utilizada são mostrados na Tabela 11 e a disposição dos nós na área de monitoramento é ilustrada pela Figura 30.

Tabela 11: Parâmetros de simulação com uma rede de 100 nós

Parâmetro	Valor
Área de monitoramento	200x200 m ²
Número de nós da rede	100 nós
Alcance do rádio de cada sensor	40 m
Energia máxima dos nós (bateria cheia)	80 J

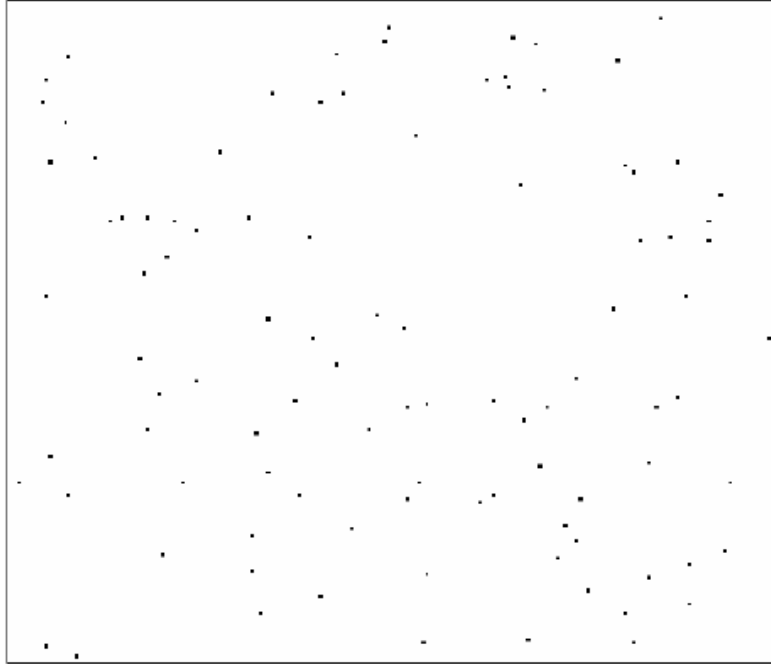


Figura 30: Rede com 100 nós

Aqui, novamente as condições de posicionamento e energia dos nós são aleatórias, sendo a energia distribuída uniformemente no intervalo $[0;80]$ J. Como no cenário anterior, o nó fonte é o do canto superior esquerdo e o nó sorvedouro é o do canto inferior direito.

Será observado aqui também o comportamento do sistema com um número maior de iterações do método. Isto significa que foram realizados vários ciclos de otimização sucessivos, mantendo as tabelas de feromônio de um ciclo para o outro. Vale apenas lembrar que, como já mencionado anteriormente, numa aplicação real, diversos ciclos de otimização são executados, em alternância com os períodos de transmissão de dados da aplicação. Entretanto, neste caso, novamente apenas a fase de otimização é observada, com a diferença de que os ciclos de otimização se repetem sucessivamente. A tendência é que as soluções a cada ciclo “evolam”, ou seja, rotas melhores possam ser descobertas a cada iteração. Os resultados obtidos com a variação de alguns parâmetros são mostrados a seguir:

Caso 1: ($k_E=1$, $k_d=1$, $Num_{ag}=20$, $\rho=0,99$, $E_{bit}=50\text{nJ/bit}$, número de iterações=10)

A melhor rota obtida para o caso 1 é mostrada na Figura 31, com $\tau_i=0,1313$. Percebe-se aqui visualmente que em termos de número de nós, a solução obtida não é a melhor possível. É possível variar alguns parâmetros do protocolo de modo a tentar obter uma solução melhor, conforme será apresentado nos casos de teste 2, 3 e 4.

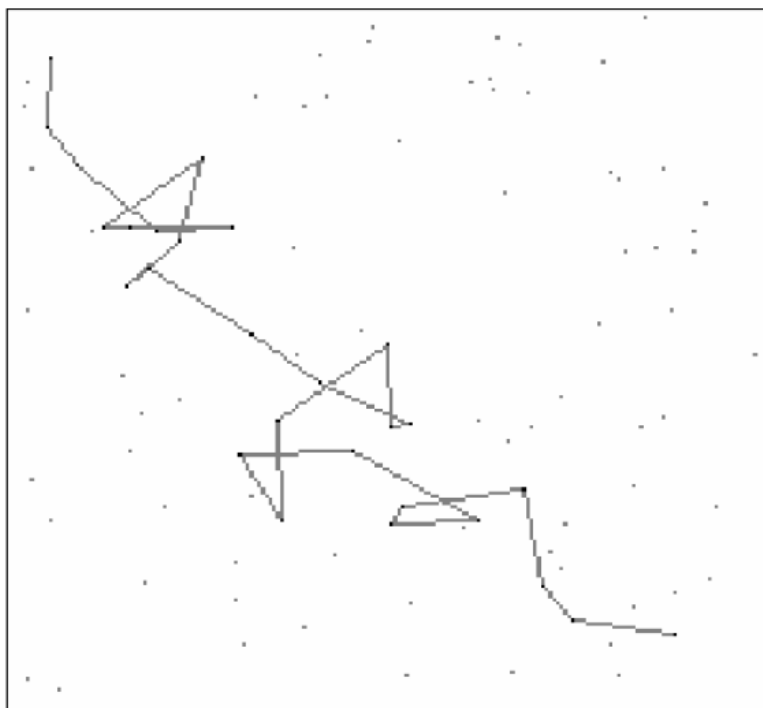


Figura 31: Melhor rota para o caso 1 - 26 nós, nó mais fraco: 17,93 J

A evolução da solução a cada iteração é mostrada na Figura 32.

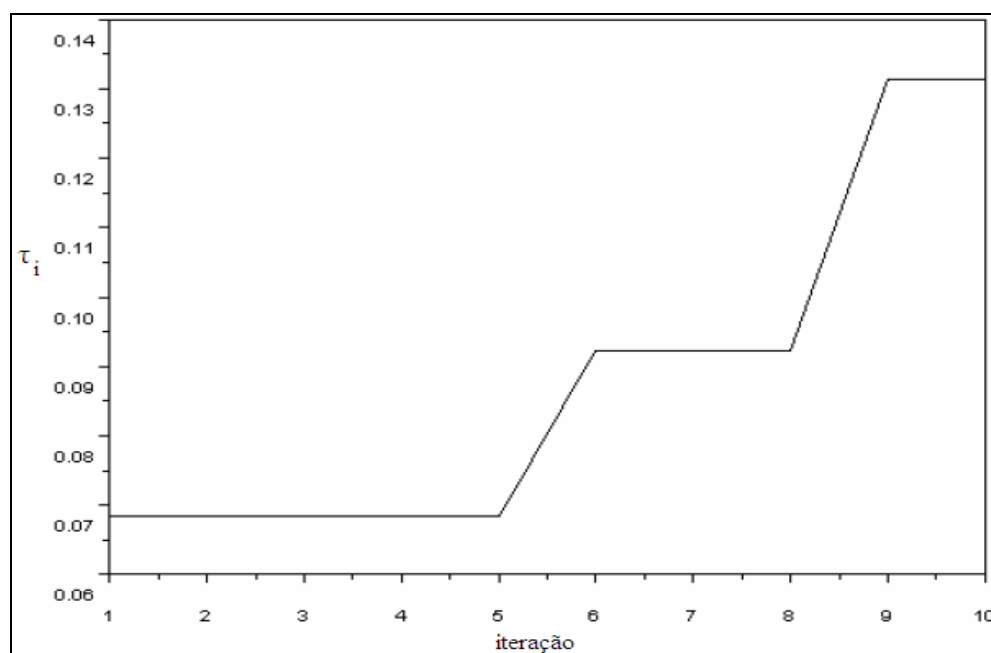


Figura 32: Evolução da solução para o caso 1

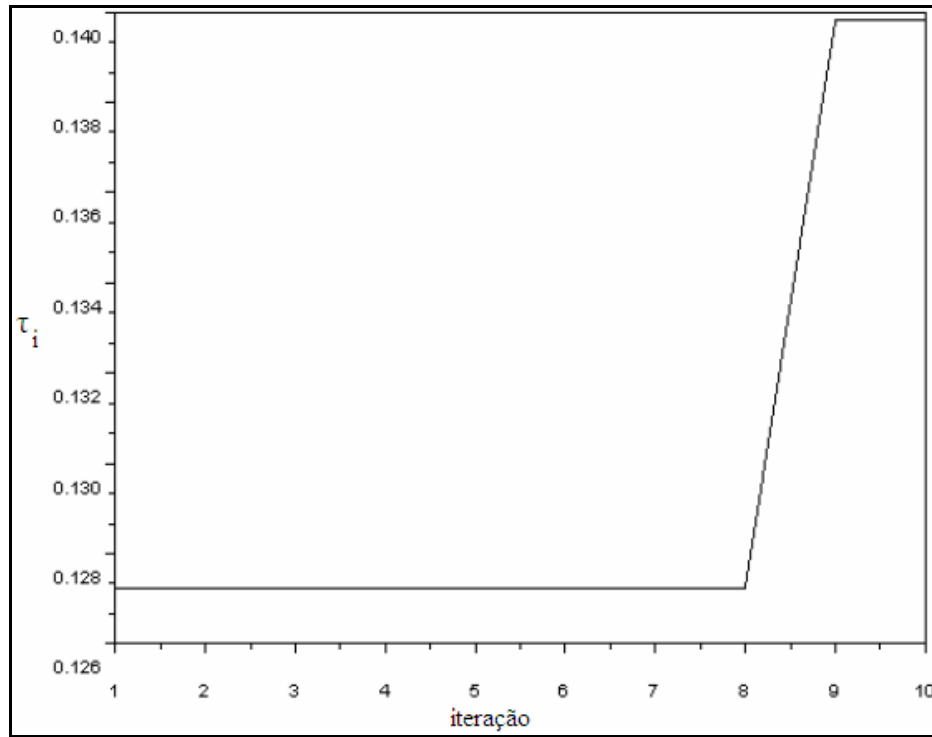


Figura 34: Evolução da solução para o caso 2

Caso 3: ($k_E=1$, $k_d=3$, $Num_{ag}=50$, $\rho=0,99$, $E_{bit}=50\text{nJ/bit}$, número de iterações=10)

O próximo teste realizado foi variar os pesos de otimização k_E e k_d , para observar o poder de controle do resultado final com a variação dos pesos. Aqui foi aumentado o peso da distância para um valor 3 vezes maior do que o do peso para a energia. O resultado obtido, com $\tau_i=0,2294$ é mostrado na Figura 35.

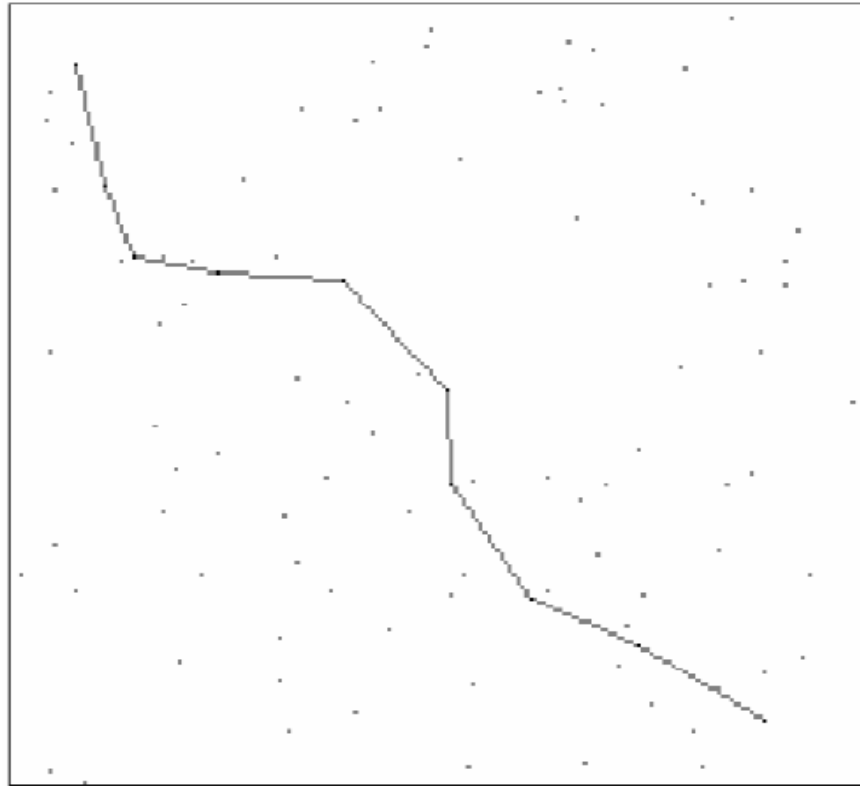


Figura 35: Melhor rota para o caso 4 - 10 nós, nó mais fraco: 12,7 J

Aqui se observa que foi obtida uma sensível redução no número de nós da rota, devido a uma maior influência da constante de peso para a distância. Porém, o nó mais fraco da rota tem um nível energia mais baixa do que no caso anterior, o que ilustra a necessidade de se manter um compromisso entre os vários objetivos do problema multiobjetivo de acordo com as necessidades da aplicação. A evolução da solução é mostrada na Figura 36.

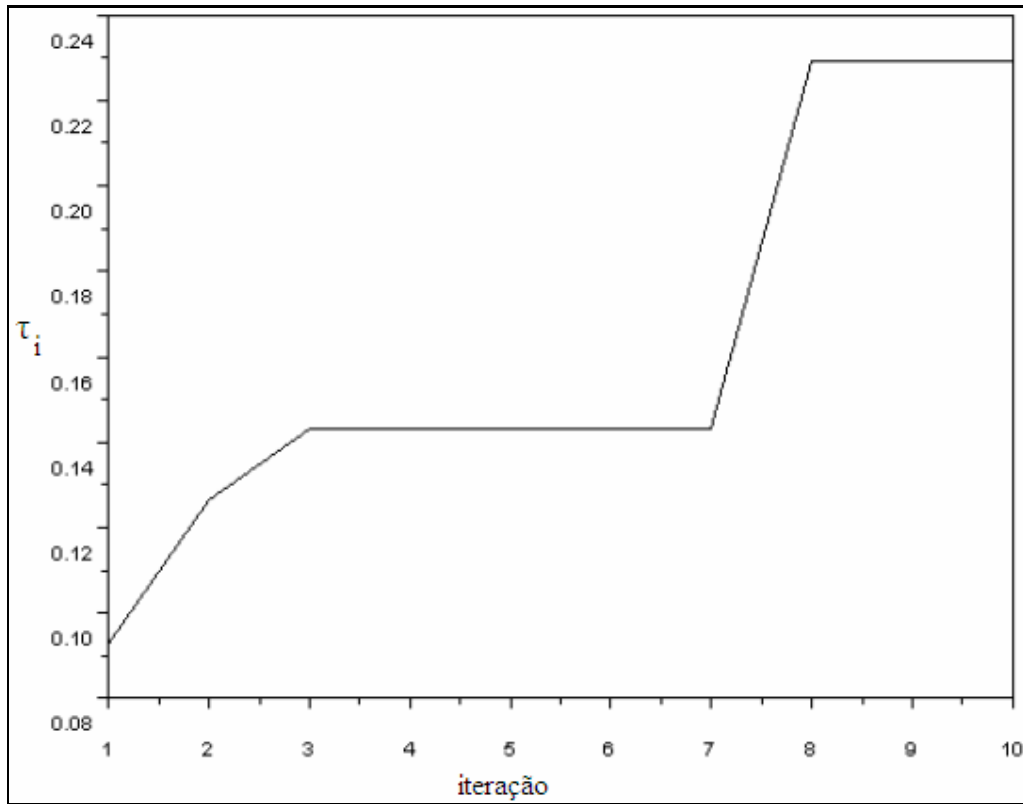


Figura 36: Evolução da solução para o caso 4

O próximo teste realizado foi o aumento do peso para a energia, para se observar o efeito contrário do realizado no último experimento:

Caso 4: ($k_E=3$, $k_d=1$, $Num_{ag}=50$, $\rho=0,99$, $E_{bit}=50\text{nJ/bit}$, número de iterações=30)

Foi aumentado aqui também o número de iterações do experimento, conseqüentemente aumentando o número de agentes e o potencial de exploração de melhores soluções. O melhor resultado com $\tau_i=0,4275$ é mostrado na Figura 37.

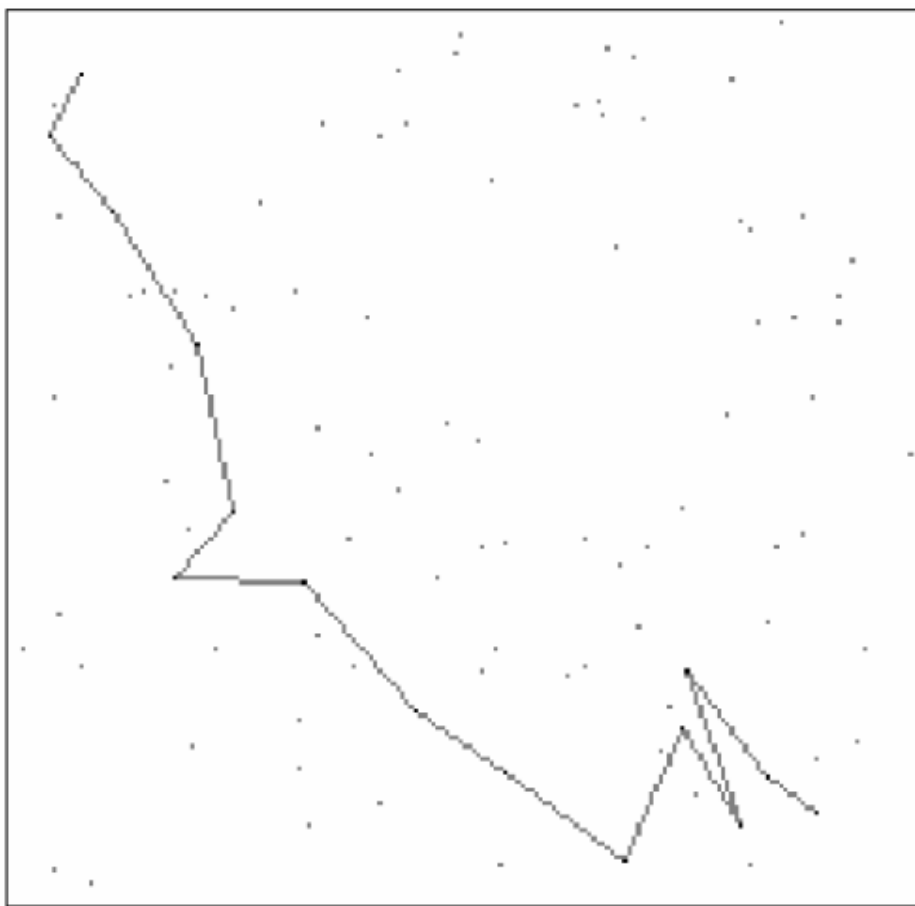


Figura 37: Melhor rota para o caso 5 - 15 nós, nó mais fraco: 21,02 J

Com o aumento do número de iterações, foi possível obter um valor melhor de energia do que nas outras tentativas. O número de nós também foi satisfatório, em comparação com os outros experimentos. É necessário um grande número de experimentos para poder calibrar as constantes do problema de forma a obter uma solução ótima ou próxima da ótima. Com o objetivo de fazer uma avaliação mais global, onde os resultados representam as médias de rotas considerando todos os nós da rede como origem e variando-se sistematicamente os pesos da função multiobjetivo, o sistema foi submetido à metodologia da avaliação estática por Fronteira de Pareto, a qual será detalhada na próxima seção.

5.3.2 Avaliação estática por Fronteira de Pareto

A avaliação por Fronteira de Pareto, descrita na seção 4.2 visa fazer uma análise do comportamento médio das soluções de um método multiobjetivo. No caso considerado neste experimento, a tentativa de obtenção da Fronteira de Pareto foi realizada através da mudança sistemática do grau de importância de cada objetivo. Aqui, ao contrário da avaliação por fonte simples, todos os nós lançam agentes de busca, de forma que o método obtém rotas de cada um dos nós da rede até o sorvedouro, formando uma árvore de roteamento. Para esta avaliação, foram realizadas simulações com camadas de enlace e física não ideais, ao contrário dos testes realizados anteriormente.

Para um problema de dois objetivos, como o proposto neste trabalho, a metodologia consiste nos seguintes passos:

- a) É definido um conjunto de N cenários de simulação possíveis gerados aleatoriamente. Os cenários devem ser diferentes entre si, de modo a avaliar o método frente a diferentes condições iniciais e minimizar o efeito da aleatoriedade na otimização. No caso do problema proposto, são utilizados 5 cenários. Cada cenário foi definido como uma rede com 40 nós, com posições aleatórias uniformemente distribuídas em uma área de $15 \times 15 \text{m}^2$. As condições iniciais de energia de cada nó também são aleatórias e distribuídas uniformemente no intervalo $[0;80]$.
- b) Todo o conjunto de cenários é simulado com valores fixos dos pesos de otimização para os objetivos. Cada conjunto de valores k_E e k_d corresponde a um conjunto de simulações $S_i, i=1, \dots, 11$. Assim, no caso do problema proposto, são fixados os valores de k_E e k_d para cada conjunto S_i e todos os 5 cenários são simulados. Ao fim do processo, são coletados as médias (μ) e os desvios padrão (σ) de todas as soluções encontradas nos quesitos número de nós das rotas e energia do nó mais fraco de cada rota. Neste caso, além dos diferentes cenários, as médias dizem respeito ao cálculo realizado considerando-se diferentes pares (nó origem – origem sorvedouro). Isto é necessário uma vez que o algoritmo é executado simultaneamente para cada nó da rede sendo considerado como origem e várias rotas diferentes são produzidas dentro do mesmo cenário.

c) Os pesos da função multiobjetivo (equação 9) são alterados e o passo b) é repetido. A variação dos pesos deve ser realizada da seguinte forma: Inicialmente, um dos pesos deve estar com o valor máximo e o outro zerado. Depois de realizadas as simulações de todos os cenários, aumenta-se um dos pesos e diminui-se o outro na mesma proporção (ver as 3 primeiras colunas na Tabela 12 e Tabela 13). Os resultados obtidos nessa metodologia são apresentados na Tabela 12 para o tamanho das rotas e na Tabela 13 para a energia das rotas.

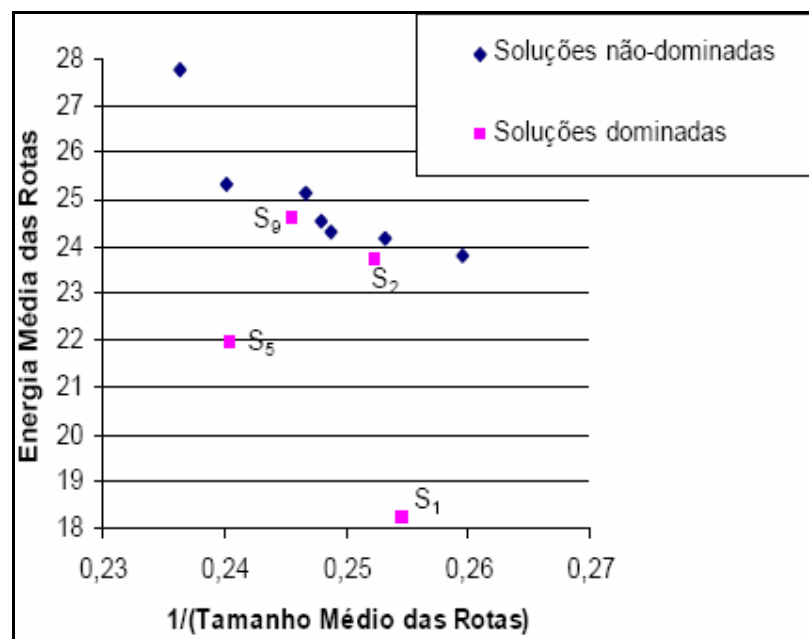
Tabela 12: Média da distância (número de nós)

Sim	Parâmetros		Estatísticas	
	k_d	k_E	μ_T	σ_T
S ₁	1	0	3,92748	0,664249
S ₂	0,9	0,1	3,96188	0,646106
S ₃	0,8	0,2	3,85248	0,721291
S ₄	0,7	0,3	4,0534	0,579183
S ₅	0,6	0,4	4,15822	0,626688
S ₆	0,5	0,5	4,01972	1,172152
S ₇	0,4	0,6	4,03316	0,902785
S ₈	0,3	0,7	4,16388	0,720419
S ₉	0,2	0,8	4,07206	1,037308
S ₁₀	0,1	0,9	4,22982	1,371122
S ₁₁	0	1	3,95006	0,606892

Tabela 13: Média da energia do nó mais fraco das rotas

Sim	Parâmetros		Estatísticas	
	k_d	k_E	μ_E	σ_E
S ₁	1	0	18,24842	4,141885
S ₂	0,9	0,1	23,72492	7,092395
S ₃	0,8	0,2	23,82658	4,241624
S ₄	0,7	0,3	25,13446	7,315885
S ₅	0,6	0,4	21,98104	9,340743
S ₆	0,5	0,5	24,3069	6,43924
S ₇	0,4	0,6	24,52654	6,245897
S ₈	0,3	0,7	25,31456	4,928004
S ₉	0,2	0,8	24,56924	7,353717
S ₁₀	0,1	0,9	27,77444	5,393451
S ₁₁	0	1	24,19538	4,05454

d) A Figura 38 apresenta a representação dos pares $(1/(\mu_T(S_i)), \mu_E(S_i))$. Assim, cada ponto do eixo corresponde a valores proporcionais às funções-objetivo com uma combinação diferente dos pesos de otimização.

**Figura 38:** Fronteira de Pareto das soluções médias obtidas

Por se tratar de um problema multiobjetivo (com dois objetivos neste caso), a escolha de qual é o melhor conjunto de simulações (solução do problema) é subjetiva e normalmente é deixada a cargo do usuário, dependendo de suas necessidades momentâneas (priorizar um dos objetivos, por exemplo, ou encontrar o melhor balanço).

A análise do gráfico mostrado na Figura 38 permite salientar que a maioria das soluções encontradas é não-dominada, ou seja, supera todas as outras soluções em pelo menos um objetivo, exceção para o conjunto {S1, S2, S5 e S9}. Entretanto, nos casos de S2, e S9, os pontos se encontram bem próximos à fronteira.

A Figura 39 mostra um exemplo das árvores de roteamento obtidas durante a resolução do problema para um cenário gerado aleatoriamente. Na figura, as rotas geradas aparecem conectando cada um dos nós da rede ao nó sorvedouro (SINK, na figura), através dos caminhos encontrados pelo método. As setas indicam da direção de propagação das mensagens, que posteriormente transmitirão dados observados por cada nó ao nó sorvedouro. Como resultado qualitativo, é possível observar visualmente que as rotas obtidas possuem poucos nós.

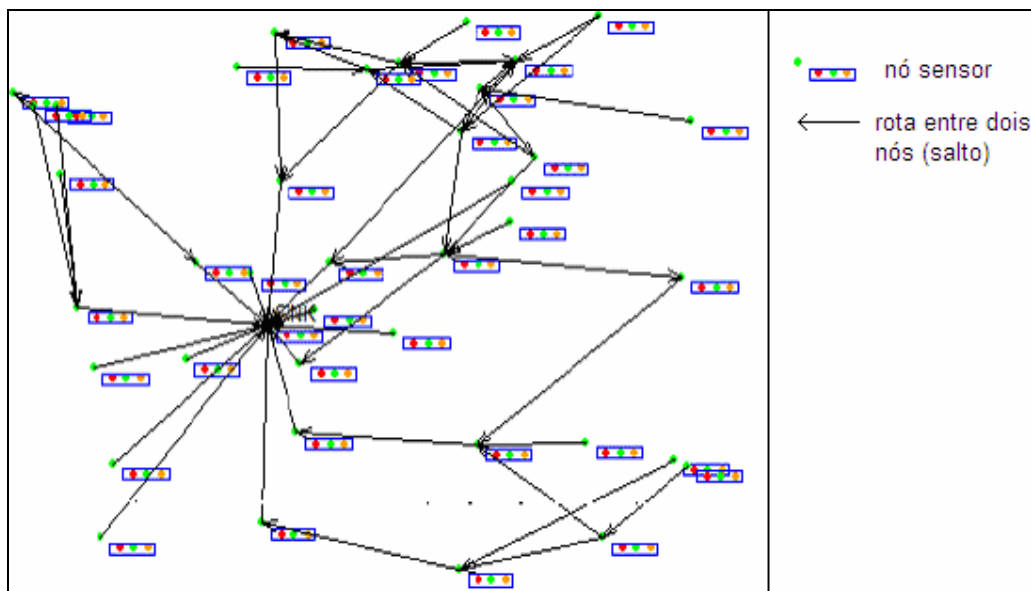


Figura 39: Exemplo de conjunto de rotas obtido em uma simulação

Tendo mostrado que o método é eficiente em termos de otimização multiobjetivo, foi possível testar seu funcionamento dinâmico, alternando ciclos de otimização e dados, e ainda compará-lo com outros protocolos em diversas métricas definidas pela literatura de RSSF. Os resultados são apresentados na próxima seção.

5.3.3 Avaliação Dinâmica

A avaliação dinâmica tem como objetivo verificar o funcionamento do protocolo proposto em regime permanente, alternando ciclos de estabelecimento de rotas e envio de dados da aplicação. Para a avaliação dinâmica, os valores numéricos dos parâmetros do *AntSensor* definidos na Tabela 9 são mostrados na Tabela 14.

Tabela 14: Valores numéricos dos parâmetros para análise dinâmica

Parâmetro	Notação	Valor
Número de Agentes (formigas)	Num_{ag}	20 agentes
Fator de evaporação do feromônio	ρ	0,99
Dimensões da rede	$L \times L$	15x15 m ²
Número de nós da rede	N	Variável (várias simulações)
Número de mensagens <i>hello</i> enviadas no início da fase de otimização	Num_{hello}	4 mensagens
Energia inicial de cada nó	E_{max}	5 mJ (milijoules)
Alcance do Rádio	A	6 m
Coefficiente do método GRASP	α	1 ou 0,5
Peso da energia no método de otimização	k_E	1 ou 3
Peso da distância do método de otimização	k_d	1
Consumo do rádio	E_{bit}	50 nJ/bit
Duração da fase de otimização	T_O	200s
Duração da fase de dados	T_D	500s
Probabilidade de ocorrência de eventos a cada pulso de relógio na fase de dados	P_{evento}	5 %

Os valores dos parâmetros aqui apresentados foram escolhidos através da realização de testes em uma rede com 40 nós, onde os parâmetros do algoritmo *AntSensor* foram pré-ajustados de forma que todos os nós da rede conseguissem obter pelo menos uma rota até o nó sorvedouro. Nos casos em que o número de agentes é muito baixo, ou a duração da fase de otimização não é longa o suficiente, alguns nós podem não obter nenhuma rota durante o processo, ficando desconectados da rede durante a fase de dados.

É importante ressaltar também que as fases de dados e otimização ocorrem em alternância: inicialmente, a rede fica apenas em otimização durante T_O segundos, sem enviar dados da aplicação. Após esse período, T_D segundos são utilizados apenas para a transmissão de dados da aplicação, retornando a seguir para a nova descoberta de rotas em uma nova fase de otimização. Dependendo da aplicação, pode ser necessário ajustar esses intervalos, caso seja necessário enviar dados da aplicação com mais frequência.

5.3.3.1 Protocolos para Comparação

Conforme já mencionado, 3 protocolos, descritos nos capítulos 2 e 3 foram implementados no *simulador Prowler* para comparação com 3 variantes do *AntSensor*:

a) EF-Tree (FIGUEIREDO, NAKAMURA e LOUREIRO, 2004): método que não visa à otimização. Gera baixo tráfego de controle e conseqüentemente consome pouca energia, porém, não estabelece rotas ótimas, penalizando o desempenho da aplicação. O EF-Tree é apresentado aqui como uma referência de baixo consumo de energia.

b) IABR (CAMILO, CARRETO, SILVA et al, 2006): método de otimização por colônia de formigas para otimização de consumo de energia.

c) EEABR (CAMILO, CARRETO, SILVA et al, 2006): método de otimização por colônia de formigas para otimização de distância (número de nós) e energia.

d) AntSensor: método multiobjetivo proposto nesta dissertação para otimização de rotas em RSSF baseado na equação 9 com $K_d=K_e=1$.

e) AntSensor $K_e=3$: análogo ao anterior com a diferença no peso da energia que é aumentado para $K_e=3$;

f) AntSensor $K_e=3$ Grasp: análogo ao anterior com a diferença da inclusão da seleção de vizinhos inspirada em grasp (com $\alpha=0.5$).

5.3.3.2 Métricas de Comparação

As métricas de desempenho utilizadas para a comparação dos protocolos já foram discutidas com detalhes na seção 2.5. As métricas aqui avaliadas são as mais difundidas para avaliação em RSSF, como mostrado em (INTANAGONWIWAT, GOVINDAN e ESTRIN, 2000). Para recapitular, as métricas que serão avaliadas são:

a) Taxa de entrega de mensagens – percentual das mensagens de dados geradas pelos nós sensores que são entregues ao nó sorvedouro:

$$\frac{M_R}{M_T} \times 100 \quad (10)$$

onde: M_T = Total de mensagens da aplicação (eventos) geradas por todos os nós durante as fases de dados do início ao fim da vida da rede

M_R = Total de mensagens da aplicação recebidas pelo nó sorvedouro do início ao fim da vida da rede.

b) Tempo de vida da rede – dividido em duas métricas: tempo de vida até a morte do primeiro nó e tempo de vida até que a metade dos nós esteja ativa.

c) Atraso médio da rede – tempo médio que uma mensagem de dados leva para ir de um nó sensor até o nó sorvedouro.

d) Eficiência Energética – produto entre a taxa de entrega e o tempo de vida até a morte da metade dos nós. Representa o compromisso entre a economia de energia e o bom desempenho na entrega de dados da aplicação.

5.3.3.3 Cenários de simulação

Aqui cada protocolo é simulado 5 vezes em cada cenário, com variação da semente do gerador de números aleatórios a cada simulação. Cada cenário possui um número diferente de nós (50, 100, 150, 200 e 250 nós) e as posições dos nós são definidas aleatoriamente com distribuição uniforme. O objetivo de variar o número de nós é observar o comportamento de cada protocolo com o aumento da densidade de nós (nós/m²). Cada métrica é coletada ao final de cada simulação e as médias de cada uma das métricas são mostradas graficamente nas figuras 40 a 49. Os valores numéricos das estatísticas obtidas (média, desvio padrão e intervalo de confiança) são mostrados no anexo 1. Os intervalos de confiança são representados graficamente através das barras verticais presentes sobre cada um dos pontos de média nas figuras 40 a 49.

5.3.3.4 Resultados

O primeiro conjunto de resultados apresentado será uma comparação entre algumas variações do *AntSensor*, com mudança de pesos e inclusão do método GRASP com $\alpha=0,5$. Os resultados são mostrados a seguir:

a) Taxa de Entrega

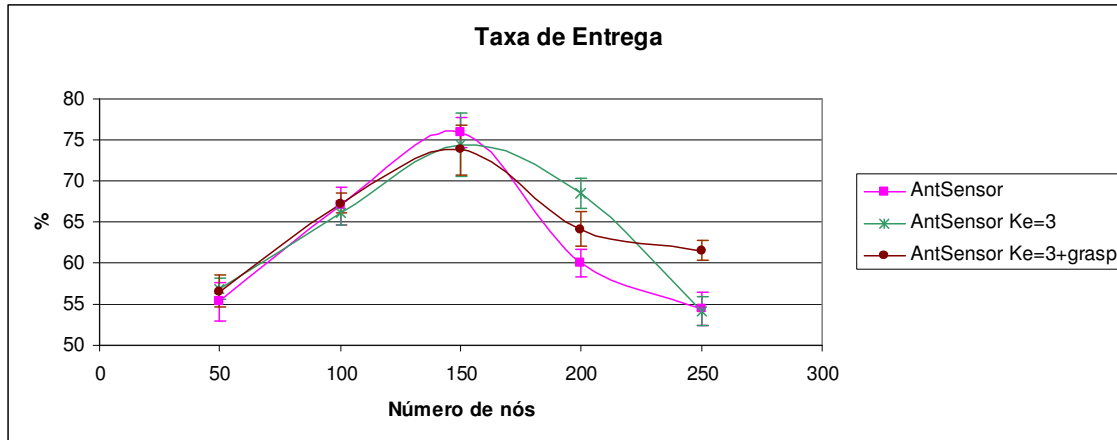


Figura 40: Taxa de Entrega do *AntSensor*

b) Atraso médio

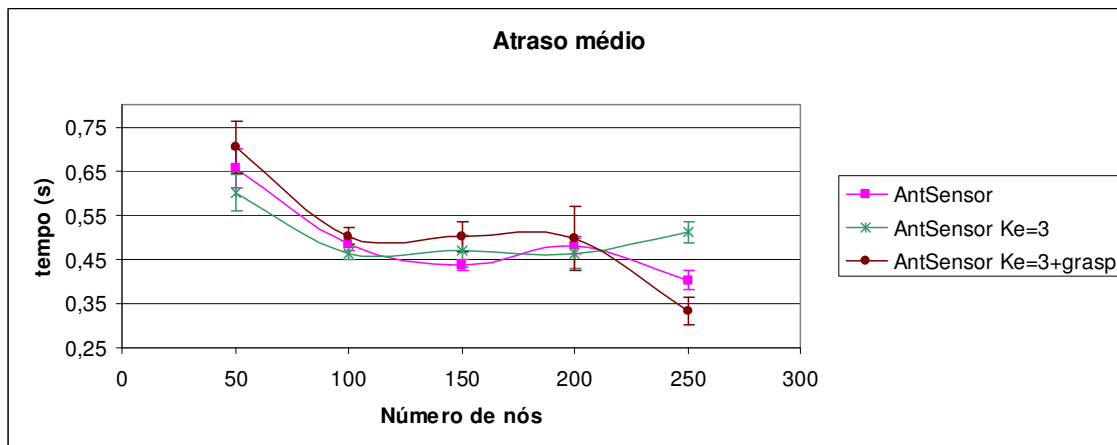


Figura 41: Atraso médio do *AntSensor*

c) Tempo de vida até a morte do primeiro nó

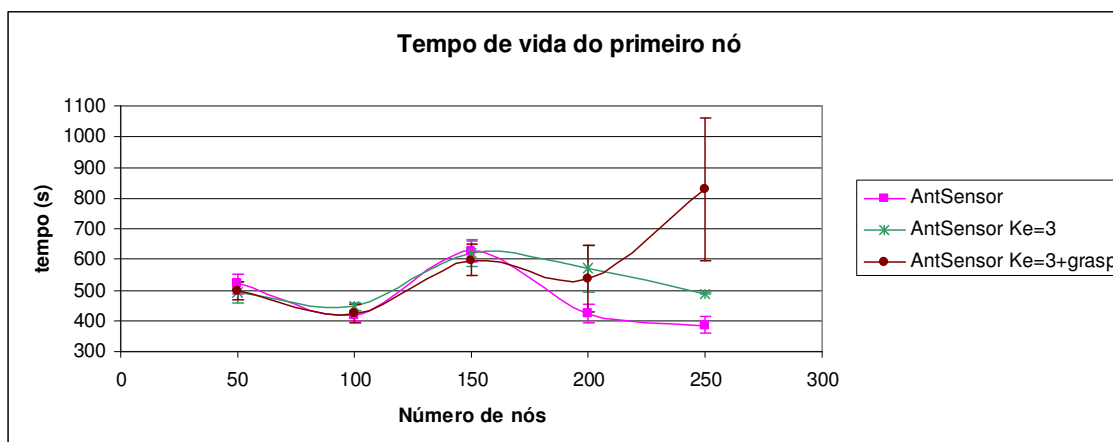


Figura 42: Tempo de vida até a morte do primeiro nó no *AntSensor*

d) Tempo de vida até que a metade dos nós esteja ativa

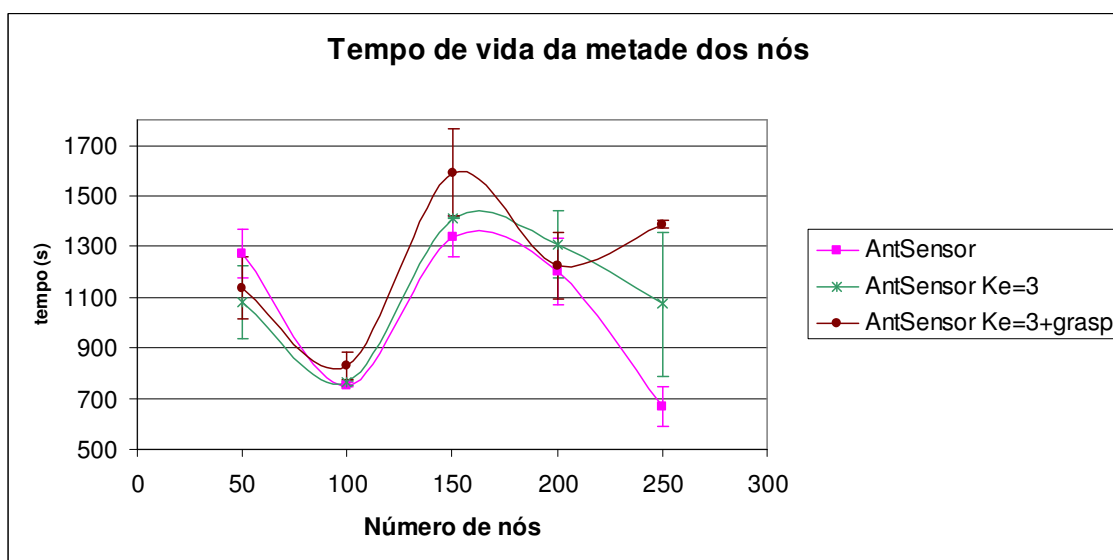
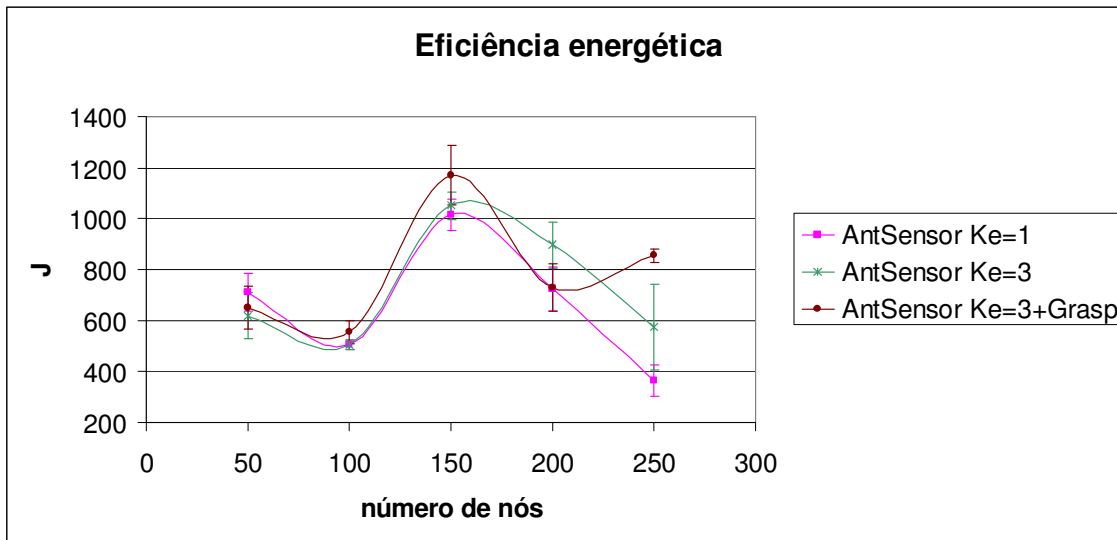


Figura 43: Tempo de vida até que metade dos nós esteja ativa no *AntSensor*

e) Eficiência Energética

**Figura 44:** Eficiência Energética do *AntSensor*

Pelos resultados aqui apresentados, verifica-se que o protocolo *AntSensor* obteve os melhores resultados com $K_e=3$ e utilização do método de seleção de vizinhos inspirado no GRASP. Conforme esperado, o aumento do peso em favor da energia proporciona um aumento no tempo de vida da rede e causa uma piora no atraso médio da rede, que está relacionado com o número de nós das rotas. Esse efeito se acentua mais com o aumento do número de nós, o que é positivo, já que o objetivo principal do *AntSensor* é a aplicação em redes de alta densidade. Como proposta para trabalhos futuros, seria interessante explorar mais essa variação de parâmetros, de forma a buscar melhores soluções.

Para a comparação com outros protocolos será utilizada a melhor versão do *AntSensor*, com $K_e=3$ e utilização do método inspirado no GRASP. Os resultados são mostrados a seguir:

a) Taxa de Entrega

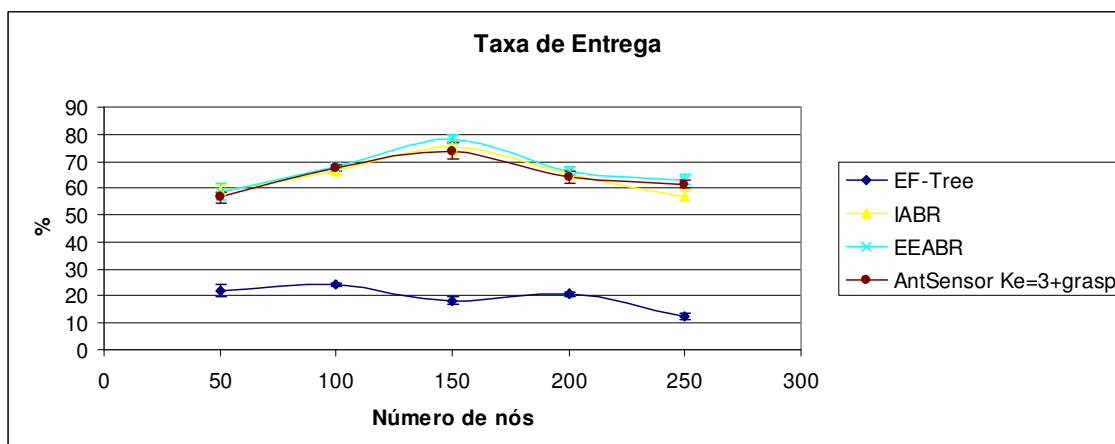


Figura 45: Comparação das taxas de entrega

b) Atraso médio

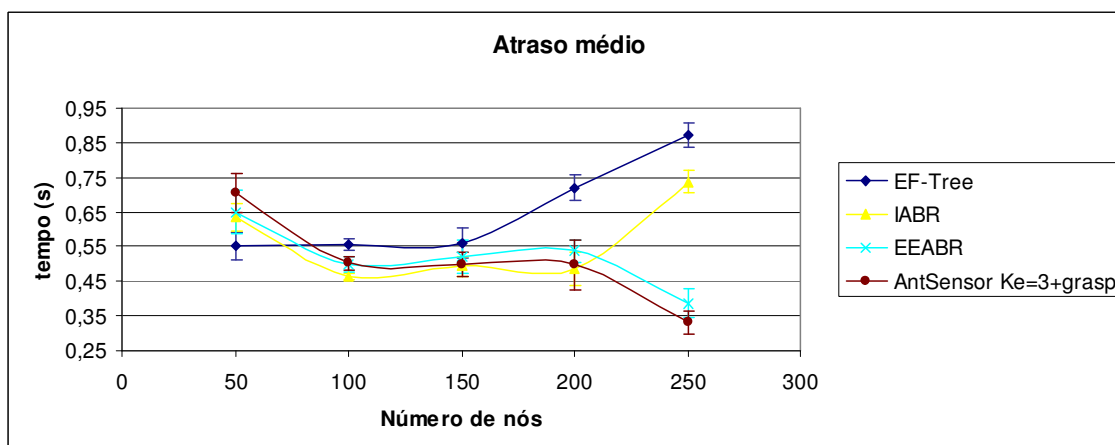


Figura 46: Comparação dos Atrasos Médios

c) Tempo de vida até a morte do primeiro nó

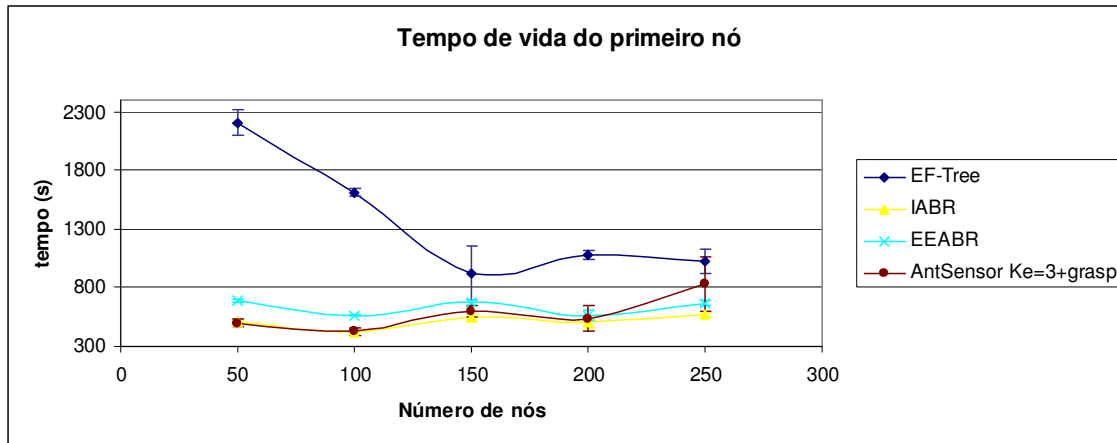


Figura 47: Comparação do tempo de vida do primeiro nó

d) Tempo de vida até que a metade dos nós esteja ativa

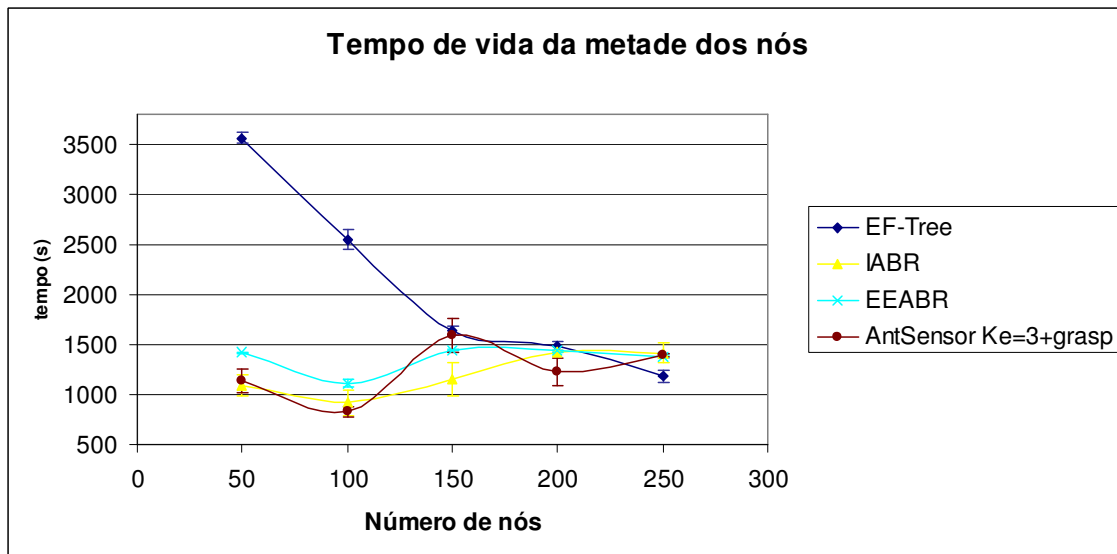


Figura 48: Comparação do tempo de vida da metade dos nós

e) Eficiência energética

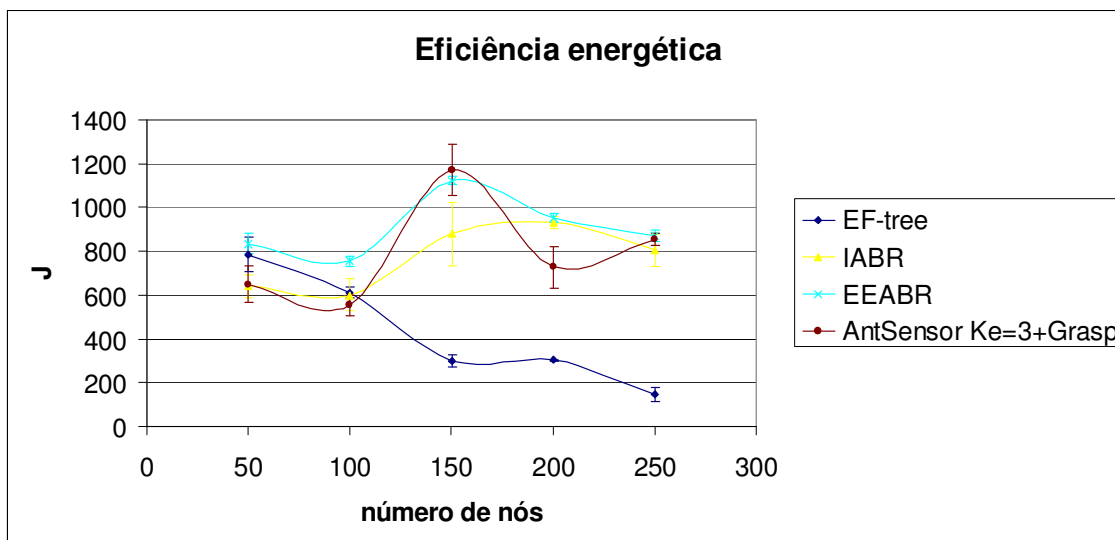


Figura 49: Comparação da eficiência energética

5.4 Análise dos Resultados

Do ponto de vista multiobjetivo, foi verificado que o protocolo *AntSensor* é capaz de controlar a qualidade das rotas obtidas, com base na variação dos pesos de otimização, de acordo com o objetivo proposto. A avaliação por Fronteira de Pareto mostrou que o protocolo é capaz de gerar soluções não dominadas, o que é muito importante para qualquer aplicação multiobjetivo. É possível alterar diversos parâmetros do problema (pesos da função multiobjetivo, número de agentes, duração das fases de otimização e dados) a fim de obter soluções otimizadas para cada tipo de aplicação desejada e de acordo com o cenário de interesse. O protocolo é aplicável a redes com qualquer número de nós, sem qualquer alteração do algoritmo descrito no capítulo 4, mantendo a mesma complexidade computacional de execução independentemente do número de nós da rede.

Com relação às métricas de desempenho avaliadas, é possível observar que: a taxa de entrega dos protocolos baseados em formigas é bastante superior (entre 55 e 80%) à do protocolo EF-Tree (Entre 10 e 25%). Isto se deve ao fato de que o EF-Tree não é um protocolo de otimização, ou seja, estabelece as rotas da maneira mais simples possível, sem se preocupar com a eficiência das rotas obtidas, de forma que os caminhos de roteamento são mais longos e não são eficientes, ocasionando mais perdas e colisões na rede. A taxa de entrega muito baixa pode inviabilizar o uso do EF-Tree em muitas aplicações.

Entre os protocolos baseados em formigas, o protocolo EEABR foi o que obteve o maior tempo de vida dos nós na maior parte dos casos, pelo fato de que além de incluir otimização de energia na criação das tabelas de feromônio o EEABR também transmite mensagens de controle com menor tamanho, reduzindo o consumo do rádio em relação ao IABR e ao *AntSensor*.

Na comparação com outros protocolos, observa-se que o *AntSensor* consegue obter resultados satisfatórios, em comparação com outros métodos e até mesmo superá-los em alguns casos. É importante ressaltar aqui que o EF-Tree é um protocolo muito simples, que consome pouca energia devido ao fato de gerar um baixo tráfego de controle. Em contrapartida obtém uma taxa de entrega de pacotes bastante baixa, comprometendo o desempenho da aplicação. É visível que os protocolos baseados em formigas conseguem manter bom desempenho mesmo com o aumento do número de nós, enquanto o EF-Tree perde bastante qualidade, com o crescimento acentuado do atraso e queda elevada do tempo de vida e eficiência energética.

É importante destacar que as curvas dos métodos baseados em ACO apresentam um comportamento oscilatório. Portanto, no futuro, testes com maior número de nós serão necessários para melhor observar a tendência dos resultados. Os intervalos de confiança mostrados nos gráficos nos permitem concluir que, apesar de haver cruzamento das curvas em alguns pontos, o comportamento de cada um dos protocolos é diferente, com a variação do número de nós.

O próximo capítulo apresenta as conclusões gerais do trabalho, com algumas sugestões de trabalhos futuros.

6 CONCLUSÕES E TRABALHOS FUTUROS

6.1 Conclusões

O trabalho realizado apresentou o *AntSensor*, um protocolo multiobjetivo que pode ser aplicado na área de RSSF, com o objetivo de otimizar o roteamento em termos de tamanho das rotas e energia. Dependendo da aplicação e das características da rede, o protocolo apresentado pode obter resultados melhores do que outros já apresentados na literatura.

O fato de o *AntSensor* incluir pesos de otimização para os diferentes objetivos, o que normalmente não ocorre em outros protocolos da literatura, representa uma maior flexibilidade para calibrar o protocolo de forma a enfrentar diferentes condições da rede e ser aplicado em um maior número de situações possíveis. Outros objetivos podem ser incluídos, modificando-se a equação (9) de modo a incluir um novo objetivo e um peso de otimização para o mesmo. O atraso médio é um fator que pode ser otimizado desta maneira, por exemplo.

A avaliação multiobjetivo ilustrou a dificuldade em se obter um compromisso entre dois objetivos que em algumas situações podem ser conflitantes. A metodologia aqui apresentada pode ser utilizada para obter um grande número de soluções e avaliar qual delas é a melhor para ser aplicada a um determinado caso prático em RSSF.

De forma geral, os objetivos de apresentação e validação do protocolo foram atingidos, ficando como contribuição um estudo comparativo com 3 outros protocolos da literatura, a ser utilizado em trabalhos futuros na área.

6.2 Trabalhos Futuros

Como sugestões de trabalhos futuros podem ser destacados os seguintes itens:

a) Análise dinâmica do *AntSensor* com uma maior variação dos pesos de otimização e do coeficiente α do método GRASP. Os coeficientes e pesos podem inclusive ser variáveis no tempo, com o objetivo de tentar obter melhores resultados que os já apresentados;

b) Estudo da variação do número de agentes: neste trabalho o número de agentes gerados por cada nó é fixo em 20 agentes. Um estudo interessante seria variar este número e observar o comportamento dinâmico do sistema. Se for possível reduzir o número de agentes sem degradar muito o desempenho, o consumo de energia será reduzido;

c) Inclusão de outros objetivos: a expressão (9) pode ser modificada para incluir outros objetivos, como atraso médio, taxa de entrega, etc.

d) Aplicação de técnicas de outros protocolos: outras técnicas apresentadas neste trabalho podem ser aplicadas na tentativa de melhorar o desempenho do *AntSensor*. A redução do tamanho das mensagens, proposta pelo EEABR é uma sugestão interessante e a agregação de dados, que ocorre no Difusão Direcionada pode ser também explorada.

e) Inclusão de uma análise multiobjetivo no funcionamento dinâmico do protocolo: a equação (9) gera um valor numérico que é utilizado para incrementar as tabelas de feromônio dos nós. Este valor é único e não leva em consideração a contribuição de cada objetivo na solução final. Se fosse guardada uma tabela de feromônio relativa a cada objetivo, seria possível propor uma análise de caráter multiobjetivo no momento do roteamento dos agentes, inclusive utilizado critérios de dominância das soluções na tomada de decisão.

f) Implementação do protocolo *AntSensor* no ns-2: o ns-2 – *Network Simulator* é o simulador mais difundido na comunidade de pesquisa em redes. Seria interessante implementar o *AntSensor* no ns-2 de modo a obter um código que pudesse ser mais facilmente distribuído para a comunidade de redes e comparado de forma mais prática a outros protocolos que já estão implementados para este simulador.

ANEXO 1 - ESTATÍSTICAS DOS TESTES DE AVALIAÇÃO DINÂMICA

a) Taxa de entrega

a.1) Média (μ)

Número de nós	Ef-tree	AntSensor Ke=Kd=1	IABR	EEABR	AntSensor Ke=3, Kd=1	AntSensor Ke=3, Kd=1 + Grasp
50	22,046	55,308	59,29968	58,732	56,87128	56,562
100	24,05	66,948	66,168	67,9786	66,08632	67,305
150	18,15	75,88	75,182	77,936	74,38333	73,8325
200	20,62464	59,956	65,264	66,544	68,56333	64,092
250	12,41	54,42	56,92	63,25	54,145	61,545

a.2) Desvio padrão (σ)

Número de nós	EF-tree	AntSensor Ke=Kd=1	IABR	EEABR	AntSensor Ke=3, Kd=1	AntSensor Ke=3, Kd=1 + Grasp
50	4,693461	5,220495	3,896258	6,999951	2,962169	4,179033
100	1,482279	5,130626	3,071314	1,498079	3,245819	2,78151
150	2,894892	4,065378	5,587389	3,724887	6,714688	6,061383
200	2,00342	3,771953	4,211731	3,211313	3,796858	4,734244
250	2,768513	4,547113	4,191673	3,858557	2,439518	1,675843

a.3) Intervalo de confiança de 68,3%³ (erro padrão – $\sigma / \sqrt{n}$)

Número de nós	EF-tree	AntSensor Ke=Kd=1	IABR	EEABR	AntSensor Ke=3, Kd=1	AntSensor Ke=3, Kd=1 + Grasp
50	$\pm 2,09898$	$\pm 2,334676$	$\pm 1,74246$	$\pm 3,130473$	$\pm 1,324722$	$\pm 1,868921$
100	$\pm 0,662895$	$\pm 2,294486$	$\pm 1,373533$	$\pm 0,669961$	$\pm 1,451574$	$\pm 1,135547$
150	$\pm 1,294635$	$\pm 1,818092$	$\pm 2,498756$	$\pm 1,66582$	$\pm 3,876727$	$\pm 3,030691$
200	$\pm 0,895957$	$\pm 1,686869$	$\pm 1,883543$	$\pm 1,436143$	$\pm 1,898429$	$\pm 2,117218$
250	$\pm 1,238117$	$\pm 2,033531$	$\pm 1,874573$	$\pm 1,725599$	$\pm 1,725$	$\pm 1,185$

³ O intervalo de confiança aqui apresentado representa a segurança de que se forem calculadas várias médias de simulações diferentes de cada grandeza, 68,3% das médias estarão contidas no intervalo ($\mu \pm \sigma / \sqrt{n}$). Considerando-se que foram realizadas poucas simulações, resultando num pequeno número de amostras (apenas 5) optou-se por um intervalo de confiança de 68,3% com o objetivo de se definir um nível aceitável de dispersão para os valores das tabelas. Para obter uma confiança maior, seria necessário tomar mais amostras de simulação de cada uma das métricas de interesse.

b) Atraso Médio

b.1) Média (μ)

Número de nós	EF-tree	AntSensor Ke=Kd=1	IABR	EEABR	AntSensor Ke=3, Kd=1	AntSensor Ke=3, Kd=1 + Grasp
50	0,55028	0,65646	0,63492	0,649444	0,60026	0,7054
100	0,554688	0,483554	0,463294	0,49818	0,46364	0,502583
150	0,55978	0,43666	0,49348	0,52128	0,4685	0,501125
200	0,719364	0,4816	0,486548	0,53704	0,464567	0,49836
250	0,8703	0,40166	0,7373	0,3878	0,5109	0,331

b.2) Desvio padrão (σ)

Número de nós	EF-tree	AntSensor Ke=Kd=1	IABR	EEABR	AntSensor Ke=3, Kd=1	AntSensor Ke=3, Kd=1 + Grasp
50	0,087129	0,100391	0,090556	0,144181	0,093727	0,130773
100	0,038231	0,02991	0,021039	0,049511	0,030567	0,041695
150	0,096029	0,028231	0,064488	0,110665	0,062872	0,070226
200	0,080966	0,041251	0,103593	0,0713	0,059652	0,162045
250	0,075589	0,049946	0,069919	0,093914	0,035072	0,043841

b.3) Intervalo de confiança de 68,3% (erro padrão – $\sigma / \sqrt{n}$)

Número de nós	EF-tree	AntSensor Ke=Kd=1	IABR	EEABR	AntSensor Ke=3, Kd=1	AntSensor Ke=3, Kd=1 + Grasp
50	$\pm 0,038965$	$\pm 0,044896$	$\pm 0,040498$	$\pm 0,06448$	$\pm 0,041916$	$\pm 0,058483$
100	$\pm 0,017097$	$\pm 0,013376$	$\pm 0,009409$	$\pm 0,022142$	$\pm 0,01367$	$\pm 0,018647$
150	$\pm 0,042946$	$\pm 0,012625$	$\pm 0,02884$	$\pm 0,049491$	$\pm 0,036299$	$\pm 0,035113$
200	$\pm 0,036209$	$\pm 0,018448$	$\pm 0,046328$	$\pm 0,031886$	$\pm 0,03444$	$\pm 0,072469$
250	$\pm 0,033804$	$\pm 0,022337$	$\pm 0,031269$	$\pm 0,042$	$\pm 0,0248$	$\pm 0,031$

c) Tempo de vida do primeiro nó

c.1) Média (μ)

Número de nós	EF-tree	AntSensor Ke=Kd=1	IABR	EEABR	AntSensor Ke=3, Kd=1	AntSensor Ke=3, Kd=1 + Grasp
50	2207,842	521,694	513,776	690,026	491,48	497,192
100	1614,238	416,474	415,356	565,674	446,53	424,9233
150	917,9221	625,132	546,648	671,108	623,22	598,1125
200	1076,316	422,892	501,188	560,902	569,5667	537,71
250	1024,1	386,02	568,95	669,01	489,78	829,57

c.2) Desvio padrão (σ)

Número de nós	EF-tree	AntSensor Ke=Kd=1	IABR	EEABR	AntSensor Ke=3, Kd=1	AntSensor Ke=3, Kd=1 + Grasp
50	248,827	72,19557	28,83531	27,87777	79,06386	67,43298
100	82,35576	40,54783	9,878465	15,95264	26,52278	74,06242
150	519,969	73,64226	66,67226	47,92362	77,07387	105,3394
200	77,18708	65,89604	96,64581	99,64033	130,8903	245,2489
250	232,0847	63,07042	50,50796	47,84859	3,917372	328,7198

c.3) Intervalo de confiança de 68,3% (erro padrão – $\sigma / \sqrt{n}$)

Número de nós	EF-tree	AntSensor Ke=Kd=1	IABR	EEABR	AntSensor Ke=3, Kd=1	AntSensor Ke=3, Kd=1 + Grasp
50	$\pm 111,2788$	$\pm 32,28684$	$\pm 12,89554$	$\pm 12,46732$	$\pm 35,35843$	$\pm 30,15694$
100	$\pm 36,83062$	$\pm 18,13354$	$\pm 4,417784$	$\pm 7,134235$	$\pm 11,86135$	$\pm 30,23586$
150	$\pm 232,5372$	$\pm 32,93382$	$\pm 29,81674$	$\pm 21,4321$	$\pm 44,49862$	$\pm 52,66972$
200	$\pm 34,51911$	$\pm 29,4696$	$\pm 43,22132$	$\pm 44,56051$	$\pm 75,56952$	$\pm 109,6786$
250	$\pm 103,7914$	$\pm 28,20595$	$\pm 22,58785$	$\pm 21,39854$	$\pm 2,77$	$\pm 232,44$

d) Tempo de vida da metade dos nós

d.1) Média (μ)

Número de nós	EF-tree	AntSensor Ke=Kd=1	IABR	EEABR	AntSensor Ke=3, Kd=1	AntSensor Ke=3, Kd=1 + Grasp
50	3563,222	1273,42	1089,16	1420,868	1080,154	1136,262
100	2546,238	753,04	916,374	1113,054	760,654	829,44
150	1631,646	1341,332	1154,256	1441,366	1413,223	1593,18
200	1478,46	1198,618	1426,778	1432,344	1308,827	1224,156
250	1187,03	668,56	1415,32	1375,79	1073,325	1388,665

d.2) Desvio padrão (σ)

Número de nós	EF-tree	AntSensor Ke=Kd=1	IABR	EEABR	AntSensor Ke=3, Kd=1	AntSensor Ke=3, Kd=1 + Grasp
50	128,7191	218,0021	231,2268	11,74035	320,7164	273,0623
100	225,0323	18,95225	281,5082	72,1802	24,39155	126,3837
150	94,98335	182,6745	370,1275	16,10096	7,183602	337,0415
200	104,7737	295,08	12,46319	13,23308	228,9592	297,4104
250	138,3771	178,6772	223,8314	28,31365	494,3878	23,20017

d.3) Intervalo de confiança de 68,3% (erro padrão – $\sigma / \sqrt{n}$)

Número de nós	EF-tree	AntSensor Ke=Kd=1	IABR	EEABR	AntSensor Ke=3, Kd=1	AntSensor Ke=3, Kd=1 + Grasp
50	$\pm 57,56493$	$\pm 97,4935$	$\pm 103,4078$	$\pm 5,250443$	$\pm 143,4288$	$\pm 122,1172$
100	$\pm 100,6375$	$\pm 8,475704$	$\pm 125,8943$	$\pm 32,27997$	$\pm 10,90823$	$\pm 51,59591$
150	$\pm 42,47784$	$\pm 81,6945$	$\pm 165,526$	$\pm 7,200568$	$\pm 4,147454$	$\pm 168,5207$
200	$\pm 46,85621$	$\pm 131,9638$	$\pm 5,573709$	$\pm 5,918015$	$\pm 132,1897$	$\pm 133,006$
250	$\pm 61,88412$	$\pm 79,90687$	$\pm 100,1005$	$\pm 12,66225$	$\pm 285,435$	$\pm 13,39463$

e) Eficiência Energética

e.1) Média (μ)

Número de nós	EF-tree	AntSensor Ke=Kd=1	IABR	EEABR	AntSensor Ke=3, Kd=1	AntSensor Ke=3, Kd=1 + Grasp
50	786,6086	709,6892	641,9562	834,7216	618,1994	649,2371
100	611,4494	504,2183	601,6054	756,8162	503,2767	554,2741
150	297,6428	1015,874	880,5627	1122,967	1051,138	1172,334
200	303,2953	723,2466	931,1531	953,1181	896,5923	728,6395
250	147,3104	363,8304	805,6001	870,1872	575,1215	854,8483

e.2) Desvio padrão (σ)

Número de nós	EF-tree	AntSensor Ke=Kd=1	IABR	EEABR	AntSensor Ke=3, Kd=1	AntSensor Ke=3, Kd=1 + Grasp
50	176,4685	166,6725	114,6572	102,3046	203,0457	188,8211
100	55,96544	41,08031	161,6633	55,42272	40,26355	103,8613
150	59,73116	135,7313	324,9219	44,17296	94,2208	234,1671
200	8,376214	193,2896	60,15434	46,54921	160,6728	188,6904
250	75,13534	134,1934	165,3492	62,11237	241,5023	37,55039

e.3) Intervalo de confiança de 68,3% (erro padrão – $\sigma / \sqrt{n}$)

Número de nós	EF-tree	AntSensor Ke=Kd=1	IABR	EEABR	AntSensor Ke=3, Kd=1	AntSensor Ke=3, Kd=1 + Grasp
50	$\pm 78,91913$	$\pm 74,53822$	$\pm 51,27628$	$\pm 45,75201$	$\pm 90,80482$	$\pm 84,44338$
100	$\pm 25,02851$	$\pm 18,37167$	$\pm 72,29802$	$\pm 24,78579$	$\pm 18,00641$	$\pm 46,44817$
150	$\pm 26,71259$	$\pm 60,7009$	$\pm 145,3095$	$\pm 19,75475$	$\pm 54,39841$	$\pm 117,0835$
200	$\pm 3,745957$	$\pm 86,44173$	$\pm 26,90184$	$\pm 20,81744$	$\pm 92,76447$	$\pm 94,3452$
250	$\pm 33,60155$	$\pm 60,01313$	$\pm 73,94641$	$\pm 27,7775$	$\pm 170,7679$	$\pm 26,55214$

REFERÊNCIAS BIBLIOGRÁFICAS

- BARROS, F. A. R. Redes de Sensores sem Fio. 2008. Disponível em: http://www.gta.ufrj.br/seminarios/semin2002_1/flavio/. Acessado em 26/02/2008.
- CROSSBOW, Crossbow Technology. 2008. Disponível em: <http://xbow.com>. Acessado em 25/02/2008.
- FONSECA, F. L. Redes de Sensores Sem Fio. 2008. Disponível em: <http://www.gta.ufrj.br/~rezende/cursos/eel879/trabalhos/wsn/index.htm>. Acessado em 26/02/2008.
- GOVINDAN, R. Wireless Sensor Networks, A Tutorial. 2008. Disponível em: <http://www.comsoc.org/freetutorials/nsc>. Acessado em 26/03/2008.
- IETF. Wireless Mobile Ad-Hoc Networks (manet) charter. 2008. Disponível em: <http://www.ietf.org/html.charters/manet-charter.html>. Acessado em 22/05/2008.
- INTEL. Sensor Nets / RFID. 2008. Disponível em: http://developer.intel.com/research/exploratory/wireless_sensors.htm. Acessado em 26/02/2008.
- WISL. Wireless Information System Lab, UCSD. Disponível em http://circuit.ucsd.edu/~curts/wisl/nav_projects_sampling.htm. Acessado em 30/05/2008.
- AKYILDIZ, I. F., SU, W., SANKARASUBRAMANIAM, Y., CAYIRCI, E. A survey on sensor networks. **IEEE Communications Magazine**, v. 40, n. 8, p. 102-114, 2002.
- APPLEGATE, D. F., BIXBY, R. E., CHVÁTAL, V., COOK, W. J. The Traveling Salesman Problem: A Computational Study. **Princeton University Press**, 2006.
- CAMILO, T., CARRETO, C. SILVA, J.S., BOAVIDA, F. An energy-efficient ant-based routing algorithm for wireless sensor networks. **ANTS Workshop**. p. 49-59, 2006.
- CHANKONG, V., HYMES, Y. Y. Multiobjective Decision Making – Theory and Methodology. **Elsevier**, 1983.
- DI CARO, G., DORIGO, M. *AntNet*: A mobile agents approach to adaptive routing. **Technical Report IRIDIA**, Université Libre de Bruxelles, 1997.

- DI CARO, G., DORIGO, M. *AntNet*: Distributed stigmergic control for communication networks. **Journal of Artificial Intelligence Research**, v. 9, p. 317-365, 1998.
- DI CARO, G., DUCATELLE, F., GAMBARDELLA, L. M. *AntHocNet*: an adaptive nature-inspired algorithm for routing in mobile networks. **European Transactions on Telecommunications**, v. 16, p. 433-455, 2005.
- DORIGO, M., MANIEZZO, V., COLORNI, A. The ant system: optimization by a colony of cooperating agents. **IEEE Transactions on Systems, Man, and Cybernetics-Part B**, v. 26, n. 1, p. 1-13, 1996.
- DORIGO, M., BONABEAU, E., THERAULAZ, G. Ant algorithms and stigmergy. **Future Generation Computer Systems**, v. 16, pp. 851-871, 2000.
- FACCIOLI, R. A. Algoritmo híbrido multi-objetivo para predição de estrutura terciária de proteínas. **Dissertação de Mestrado**. Escola de Engenharia de São Carlos da Universidade de São Paulo, SP, 2007.
- FAROOQ, M. *BeeHive*: An efficient fault-tolerant routing algorithm inspired by honey bee behaviour. **ANTS**, 2004.
- FERREIRA, P. A. V. Análise Otimização Multiobjetivo – Teoria e Aplicações. **Tese de Livre Docência**. Escola politécnica da Universidade de São Paulo, SP, 1998.
- FIGUEIREDO, C. M. S., NAKAMURA, E. F., LOUREIRO, A. A. F. Protocolo adaptativo híbrido para disseminação de dados em redes de sensores sem fio auto-organizáveis. **Anais do 22o Simpósio Brasileiro de Redes de Computadores (SBRC'04)**, p. 43-56, Gramado, RS, 2004.
- GARBE, B. Análise de algoritmos de roteamento baseados em formigas. **Dissertação de Mestrado**. FEEC, UNICAMP, 1998.
- HABIB, E., CAMARA, D., LOUREIRO, A. A. F. ICA: um novo algoritmo de roteamento para redes de sensores. **Anais do 22o Simpósio Brasileiro de Redes de Computadores (SBRC'04)**, p. 147-160, Gramado, RS, 2004.

- HEINZELMAN, W. R., CHANDRAKASAN, A., BALAKRISHNAN, H. Energy-efficient communication protocol for wireless microsensor networks. **International Conference on System Sciences**. Hawaii, 2000.
- HEINZELMAN, W. R., KULIK, J., BALAKRISHNAN, H. Adaptive protocols for information dissemination in wireless sensor networks. **Proceedings of the 5th ACM/IEEE Mobicom Conference**. Seattle, WA, 1999.
- INTANAGONWIWAT, C., GOVINDAN, R., ESTRIN, D. Directed Difusion: A scalable and robust communication paradigm for sensor networks. **Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCOM '00)**. Boston, MA, 2000.
- KARP, B., KUNG, H. T. GPSR: Greedy perimeter stateless routing for wireless networks. **Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCOM '00)**. pp. 243-254, Boston, MA, 2000.
- LINDSEY, S., RAGAVENDRA, C. S. PEGASIS: Power-efficient gathering in sensor information systems. **XXII Proceedings of the IEEE Aerospace Conference**. pp. 1125-1130, Big Sky, MT, 2002.
- LUZ, G. D. Roteamento em redes de sensores. **Apostila**. Instituto de Matemática e Estatística, Universidade de São Paulo, 2004.
- NAKAMURA, E. F., FIGUEIREDO, C. M. S., LOUREIRO, A. A. F. Disseminação de dados adaptativa em redes de sensores sem fio auto-organizáveis. **XXII Simpósio Brasileiro de Redes de Computadores**. p. 29-42, Gramado, RS, 2004.
- PARPINELLI, R.S., LOPES, H.S., FREITAS, A.A. An ant colony algorithm for classification rule discovery. In: Abbas, H. A., Sarker, R. A., Newton, C. S. (Eds.) **Data Mining: A Heuristic Approach**. Hershey, PA: Idea Group Publishing, p. 190-208, 2001.
- PERKINS, C., ROYER, E. The Ad Hoc On-Demand Distance Vector (AODV) routing. **Ad Hoc Networking**; pp. 173-219. Adisson-Wesley, 1999.

- RAMASUBRAMANIAN, V., HAAS, Z. J., SIRER, E. G. SHARP: A hybrid adaptive routing protocol for mobile ad hoc networks. **Proceedings of the International Symposium of Mobile Ad Hoc Networking and Computing (MOBIHOC'03)**. Annapolis, MD, 2003.
- RAPPAPORT, T. Wireless Communications: Principles and Practice, 2nd Edition. **Prentice Hall**, 2001.
- RESENDE, M., PISTOULIS, L. S. GRASP, in Handbook of applied optimization; pp. 168-182. **Oxford University Press**, 2002.
- SCHOONDERWOERD, R., HOLLAND, O., BRUTEN, J., ROTHKRANTZ, L. Ant based load balancing in telecommunications networks. **Adaptive Behavior**. pp 169-207, 1996.
- SILVA, R. C., MUNARETTO, A., DELGADO, M. R. Otimização por colônia de formigas para o roteamento e consumo de energia em redes de sensores sem fio. **Anais do I SBIC Simpósio Brasileiro de Inteligência Computacional**. Florianópolis, SC, 2007.
- SIMON, G., VOLGYESI, P., MARÓTI, M., LÉDCZY, A. Simulation-Based optimization of communication protocols for large-scale wireless sensor networks. **IEEE Aerospace Conference**. Big Sky, MT, 2003.
- TANENBAUM, A. S. Redes de Computadores. 4 ed. Rio de Janeiro, Brasil: **Ed. Campus**, 2003.
- WHITE, T. Routing wit swarm intelligence. **Technical Report**, SCE, 1997.
- YE, W., HEIDERMAN, J., ESTRIN, D. An energy-efficient MAC protocol for wireless sensor networks. **Proceedings of the IEEE Infocom**. New York, NY: USC/Information Sciences Institute, IEEE, p. 1567-1576, 2002.
- ZHANG, Y., KUHN, L. D., FROMHERZ, M. P. J. Improvements on Ant Routing for sensor networks. **ANTS workshop**. pp. 154-165, 2004.
- ZELENY, M. Multiple Criteria Decision Making. **Mc-Graw Hill**, 1982.

RESUMO:

Redes de Sensores Sem Fio (RSSF) são um tipo especial de rede *Ad Hoc*, sem hierarquia, onde os nós não possuem conhecimento prévio de sua posição em relação aos outros nós da rede e precisam se comunicar entre si para estabelecer a estrutura de comunicação da rede. Os nós dessas redes são bastante simples e possuem recursos limitados.

Um amplo ramo de pesquisa na área de Redes de Sensores sem Fio é o estudo da otimização do consumo de energia, tendo em vista que, em muitos casos, os nós são dispostos em regiões inóspitas, onde é impossível realizar a substituição das baterias. Dessa forma, os nós possuem uma vida útil limitada, conseqüentemente limitando também a vida útil da rede como um todo.

O presente trabalho tem como objetivo apresentar uma abordagem multiobjetivo para otimização do roteamento e consumo de energia para Redes de Sensores sem Fio. Esta abordagem é baseada no algoritmo de Otimização por Colônia de Formigas – ACO – *Ant Colony Optimization*. O método ACO é uma técnica da Computação Natural que procura estudar o comportamento das formigas em busca de alimento para a otimização de processos computacionais. Serão descritos também os principais protocolos da literatura para RSSF, como forma de enfatizar as principais características necessárias para um bom protocolo de roteamento em RSSF e compará-los com o protocolo proposto.

A abordagem proposta foi testada em simulação e se mostrou eficiente de acordo com os critérios de avaliação multiobjetivo, na avaliação com apenas um nó fonte de dados e no teste de otimização multiobjetivo baseado na Fronteira de Pareto. Para comparação com três outros protocolos propostos na literatura de RSSF (EF-Tree, IABR e EEABR), foram realizadas simulações de uma aplicação real baseada em eventos com variação da densidade de nós na área de monitoramento. O protocolo proposto obteve resultados bastante satisfatórios e em alguns casos superiores aos outros em termos de taxa de entrega de mensagens, atraso e tempo de vida da rede.

PALAVRAS-CHAVE

Redes de Sensores sem Fio, Otimização por Colônia de Formigas, Multiobjetivo.

ÁREAS DO CONHECIMENTO

1.03.03.04-9	Sistemas de Informação
1.03.04.00-2	Sistemas de Computação
1.03.03.00-6	Metodologia e Técnicas da Computação
3.04.06.03-0	Sistemas de Telecomunicações

ano
2008

Nº: 479

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)