Luciano Gonda

**Manutenção de Topologia de Redes de
Sensores Sem Fio Usando Tabela de
Decisão Adaptativa**

São Paulo
2010

Luciano Gonda

Manutenção de Topologia de Redes de Sensores Sem Fio Usando Tabela de Decisão Adaptativa

Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Doutor em Engenharia Elétrica.

Área de concentração:
Sistemas Digitais

Orientador:
Prof. Livre-Docente Carlos
Eduardo Cugnasca

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 24 de julho de 2010.

Assinatura do autor _____

Assinatura do orientador _____

Ficha Catalográfica

Gonda, Luciano

Manutenção de Topologia de Redes de Sensores Sem Fio Usando Tabela de Decisão Adaptativa. São Paulo, 2010. 96 p.

Tese (Doutorado) — Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1. Redes de Sensores Sem Fio. 2. Tecnologia Adaptativa. I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais. II. Título.

DEDICATÓRIA

Dedico este trabalho ao meu filho Rodrigo que é uma benção na minha vida, a minha esposa Lucia que me “deu” esta alegria e a meus pais Jorge e Zilda (*in memoriam*) por tudo o que fizeram por mim durante toda a trajetória da minha vida.

AGRADECIMENTOS

Esta tese é o resultado de cinco anos de trabalho e com certeza só foi possível graças ao apoio de muitas pessoas, das quais gostaria de agradecer. Peço desculpas se alguém não foi lembrado, mas como o apoio vem de muitas pessoas, realmente fica difícil lembrar de todos.

Primeiramente, gostaria de agradecer à minha esposa Lucia, que nestes anos de trabalho sempre me apoiou e abdicou de minha presença em diversos momentos para que eu pudesse trabalhar e me concentrar nos estudos da melhor maneira possível. Agradeço pela compreensão nos momentos mais difíceis, pelas viagens que teve que suportar e por todo apoio, amor e carinho nestes anos de trabalho.

Aos meus pais Jorge e Zilda (*in memoriam*) que sempre incentivaram meus estudos desde o início, que sempre me ensinaram coisas boas, me deram carinho, amor, afeto e sempre me apoiaram em tudo na minha vida.

Ao meu orientador, Professor Livre-Docente Carlos Eduardo Cugnasca. Primeiramente, por ter acreditado no meu trabalho e ter aceitado me orientar mesmo com as dificuldades de trabalho e da distância. Pela paciência e pela dedicação a este trabalho. Pelas inúmeras conversas e pelas trocas de experiências e pelo incentivo que fez com que não desistisse do mesmo. Muito obrigado pelas oportunidades!

Ao Professor Livre-Docente João José Neto pelos valiosos momentos do café, nos quais sempre me apoiou e acreditou neste trabalho. Por todas as vezes que surgia uma dúvida quanto ao desenvolvimento e aquelas sábias palavras sempre de apoio faziam com que um novo ânimo surgia para o desenvolvimento do trabalho.

Aos amigos Amaury, Said, Raphael, Fábio, Leo pelos momentos de estudo e por terem me acolhido da melhor maneira possível em SP. Além disso, pelos momentos de companheirismo, descontração e de amizade que sempre vão ficar marcados.

Aos amigos Danilo, Sandro, Welton, Valmir, Adriano pelos momentos de alegria, encontros e pelas diversas vezes que apoiaram neste trabalho.

Ao amigo Sérgio, por ser a primeira pessoa a me acolher em SP e pela hosi-

talidade.

Aos amigos da UCDB Leo, Jeferson, Hemerson, Wanderlei, Naka, Mauro, Edson, Alexsandro e todos os demais professores que sempre contribuíram e incentivaram este trabalho.

Ao pessoal do LAA pelo apoio, conversas e trocas de experiências que com certeza foram fundamentais para o desenvolvimento deste trabalho.

À Universidade Católica Dom Bosco pela bolsa concedida durante o primeiro ano de trabalho.

Ao CNPQ, pelo suporte ao desenvolvimento deste trabalho por meio do Projeto "Automação de Casas de Vegetação Utilizando Redes de Sensores Sem Fio" aprovado no Edital Universal - 2006.

À professora Cristina Borba pela revisão (apesar de ter sido enviado de última hora) do *abstract* deste trabalho.

Aos membros da banca, pelas valiosas contribuições.

RESUMO

Esta tese propõe um algoritmo denominado ADAP_RSSF para manutenção de topologia de Redes de Sensores Sem Fio para Aplicações com Modelo de Entrega de Dados Híbrida, com o objetivo de garantir que eventos como falhas e inserção de novos nós não interfiram no funcionamento da rede. O algoritmo proposto utiliza uma Tabela de Decisão Adaptativa com algumas modificações, para realizar a reconstrução da topologia sempre que necessário. Além disso, é apresentada a formalização de grafos dinâmicos, baseada na definição de Dispositivos Adaptativos; eles são utilizados para mostrar graficamente as diversas mudanças que podem ocorrer na topologia da rede. A utilização de Tecnologia Adaptativa é importante para garantir que mudanças e eventos não previstos possam ser incorporados de maneira automática, sem que haja necessidade de interação com o usuário após a configuração inicial da tabela, pois nem sempre isto é possível em uma Rede de Sensores Sem Fio. Dessa forma, sempre que um ou mais eventos que possam causar mudanças na rede são detectados, é realizada uma busca na Tabela de Decisão Adaptativa e, caso não exista nenhuma regra que trate aquele evento específico, uma nova regra é criada usando a camada adaptativa da tabela. A solução proposta nesta tese mostra uma possibilidade de aplicação da Tecnologia Adaptativa em Redes de Sensores Sem Fio. Uma outra contribuição deste trabalho é a possibilidade de modificação no comportamento do algoritmo de forma que não somente os eventos e regras pré-estabelecidos sejam considerados na manutenção da topologia, de tal forma que a incorporação de novas informações sobre mudanças na topologia e funcionamento da rede seja possível sempre que necessário. Por fim, são apresentados o funcionamento do algoritmo, uma breve análise baseada em *hop counts* e considerações sobre uma possível simulação. Esta análise mostrou que o algoritmo apresenta um *hop count* inferior ou igual a algoritmos como o REMUDA e MULTI.

Palavras Chave: Redes e Comunicação de Dados; Sensor; Adaptação; Tecnologia.

ABSTRACT

An algorithm named ADAP_RSSF is proposed to solve Wireless Sensor Network for Applications with Hybrid Data Delivery Model topology maintenance problems, guaranteeing that network operation will not be interfered by node faults or insertion of new nodes. The proposed algorithm uses a modified Adaptive Decision Table to rebuild network topology as needed. Besides, based on adaptive device definition, the description of dynamic graphs is also presented to use these devices to graphically represent a Wireless Sensor Network topology. The use of Adaptive Technology is important to guarantee that unforeseeable changes and events may be incorporated with no user interaction. Thus, every time an event that may change network topology is detected, a search on the Adaptive Table is executed and, if no results are found, the adaptive layer accounts for a new rule creation. The solution described here shows the applicability of Adaptive Technology in Wireless Sensor Networks. Another contribution is the algorithm behavior modification in such a way that not only are previous events and rules used in topology rebuilding, but new rules and criteria may be incorporated as needed. The algorithm operation, a brief analysis based on hop count and a possible simulation considerations are also presented. This analysis showed that the proposed algorithm presents a hop count less or equal other algorithms, such as, REMUDA and MULTI.

Keywords: Data and Communication Network; Sensor; Adaptation; Technology

Lista de Figuras

1	Exemplo de comunicação <i>multihop</i>	8
2	Uma Taxonomia para Controle de Topologia.	16
3	Estrutura Geral de Um Dispositivo Adaptativo(TCHEMRA, 2009)	27
4	Estrutura Geral de uma TDA (TCHEMRA, 2009).	35
5	Estrutura de uma Matriz de Decisão (FULÖP, 2005).	40
6	Escala Fundamental de Saati (SAATI, 1991).	43
7	Estrutura Geral da TDAE (TCHEMRA, 2009).	45
8	Exemplo de Construção da Topologia Inicial.	52
9	Exemplo de Construção da Topologia Inicial Utilizando Lista de Possíveis Pais.	52
10	Um exemplo de configuração de regras de uma TDA.	55
11	Configuração c_0 do grafo dinâmico.	57
12	Configuração t_0 da TDA.	58
13	Regra existente a ser buscada na TDA.	58
14	Configuração c_1 do grafo dinâmico que representa a RSSF.	59
15	Falha nos nós v2 e v7 e inserção do nó v10.	59
16	Regra a ser buscada e posteriormente inserida na TDA.	60
17	Configuração t_1 da Tabela de Decisão Adaptativa.	60
18	Configuração c_2 do grafo dinâmico que representa a RSSF.	61
19	Procedimento Receive do Algoritmo ADAP_RSSF.	63
20	Funcionamento do Recebimento de Pacotes no Nó SINK.	64
21	Funcionamento do Recebimento de Pacotes no Nó MIDDLE_SINK.	64
22	Funcionamento do Recebimento de Pacotes no Nó INTERMEDI- ATE.	65

23	Funcionamento do Recebimento de Pacotes no Nó COMMON. . .	66
24	Funcionamento do Envio de Pacotes.	66
25	Especificação da TDA.	67

Lista de Tabelas

1	Exemplo de Tabela de Decisão com Entradas Limitadas.	32
2	Exemplo de Tabela de Decisão com Entrada Mista.	33
3	Exemplo de uma Tabela de Decisão com Entradas Limitadas. . .	33
4	Comparação do Algoritmo ADAP_RSSF com outros algoritmos no caso de Reconstrução Local.	68

Lista de Abreviaturas e Siglas

AHP Analytic Hierarchy Process

ATC Aggregation Time Control

CA Camada Adaptativa

EF-Tree Earlier First Tree

ELECTRE Elimination and Choice Translating Reality

FA Função Adaptativa

GAF Geographical Adaptive Fidelity

GMRE Greedy Maximum Residual Energy

LEACH Low-Energy Adaptive Clustering Hierarchy

LEPS Link Estimation and Parent Selection

MACBETH Measuring Attractiveness by a Categorical Based Evaluation
Technique

MCDA Multiple Criteria Decision Aiding

MUAT Multiple Attribute Utility Theory

NS-2 Network Simulator 2

OLAP Online Application Processing

PROMETHEE Preference Ranking Method for Enrichment Evaluation

RSSF Rede de Sensores Sem Fio

SMART Simple Multi-Attribute Rating Technique

SID Source-Initiated Dissemination

SIG Sistemas de Informação Gerencial

SPIN Sensor Protocols for Information via Negotiation

SSR Self-Selecting Routing

TA Tecnologia Adaptativa

TDA Tabela de Decisão Adaptativa

TDAE Tabela de Decisão Adaptativa Estendida

TDN Tabela de Decisão Não Adaptativa

TODIM Tomada de Decisão Interativa Multicritério

TOPSIS Technique for Order Preference by Similarity to Ideal Solution

Sumário

1	Introdução	1
1.1	Objetivos	3
1.2	Justificativa	4
1.3	Organização do Texto	5
2	Redes de Sensores Sem Fio	7
2.1	Principais Características das Redes de Sensores Sem Fio	8
2.2	Considerações sobre Comunicação em RSSF	9
2.2.1	Protocolos Centrados em Dados	10
2.2.2	Protocolos Hierárquicos	11
2.2.3	Protocolos Baseados em Localização	11
2.3	Aplicações	12
2.3.1	Monitoramento de Ambientes	12
2.3.2	Monitoramento de Pacientes	12
2.3.3	Outras Aplicações	13
2.4	Consumo de Energia	14
2.5	Construção de Topologia	15
2.5.1	Taxonomia para Controle de Topologia em Redes <i>Ad Hoc</i>	16
2.5.2	Um Modelo de Controle de Topologia para RSSF	16
2.6	Manutenção de Topologias em RSSF	18
2.7	Algoritmos para RSSF	20
2.7.1	LEACH	20
2.7.2	SPIN	20

2.7.3	Directed Diffusion	21
2.7.4	REMUDA	22
2.7.5	MULTI	23
2.8	Considerações Finais Sobre o Capítulo	23
3	Tecnologia Adaptativa	25
3.1	Dispositivos Adaptativos	25
3.2	Grafos Dinâmicos	28
3.3	Tabelas de Decisão Adaptativas	30
3.4	Tabela de Decisão Adaptativa Estendida	37
3.4.1	Decisões Multicritérios	37
3.4.2	Descrição da Tabela de Decisão Adaptativa Estendida . . .	43
3.5	Considerações sobre o Capítulo	48
4	Algoritmo para Manutenção de Topologia Utilizando Tabela de Decisão Adaptativa	50
4.1	Definição do Problema de Manutenção de Topologia	50
4.2	Aspectos Gerais do Algoritmo Proposto	51
4.2.1	Construção da Topologia Inicial	51
4.2.2	Reconstrução da Topologia	53
4.2.3	Funcionamento do Algoritmo	56
4.3	Algoritmo Proposto	62
4.3.1	Especificação do Algoritmo	63
4.3.2	Análise do Comportamento do Algoritmo Proposto ADAP_RSSF	64
4.3.3	Algumas Considerações sobre o Algoritmo	68
4.4	Considerações sobre a Simulação	70
4.4.1	Implementação do Agente	74
4.4.2	Alterações no NS-2	75
4.5	Considerações sobre o Capítulo	75

5	Conclusões	77
5.1	Contribuições	78
5.2	Trabalhos Futuros	79
5.3	Considerações Finais	81
	Referências	83
	Apêndices	92
A	Pseudo-Código	92

1 Introdução

Devido à evolução de diversas áreas, em especial comunicação sem fio, microprocessadores e dispositivos eletromecânicos, a tecnologia de Redes de Sensores Sem Fio (RSSF) se tornou viável, possibilitando que pequenos dispositivos, dispostos em uma área a ser monitorada, pudessem reproduzir, por meio de sensoramento, processamento e comunicação com outros nós, fenômenos físicos com mais acurácia (CULLER; ESTRIN; SRIVASTAVA, 2004). Uma RSSF é composta por diversos módulos distribuídos, denominados nós sensores, que possuem capacidade de sensoramento, processamento e comunicação (AKYILDZ et al., 2002). Diversas aplicações para RSSF estão surgindo, em áreas como medicina, agricultura, controle de ambientes, militar, entre outras (BARONTI et al., 2007).

Como os nós sensores possuem dimensões pequenas, em geral, apresentam limitação em relação à capacidade de processamento, armazenamento, de energia e de alcance do seu rádio de comunicação. Conseqüentemente, os nós sensores devem ser agrupados para executarem tarefas de maneira colaborativa, pois individualmente não são muito eficazes para executar o sensoramento de uma região.

Além disso, uma das características essenciais em uma RSSF é a capacidade de auto-organização dos nós sensores com a finalidade de estabelecer e/ou manter a comunicação da rede. Dessa forma, ao serem espalhados em grandes quantidades em uma área de monitoramento, os nós sensores devem se organizar automaticamente, para que seja estabelecida a comunicação entre os diversos dispositivos da rede (BARONTI et al., 2007). É importante ressaltar que a auto-organização é necessária, não somente no instante em que a rede está sendo criada, mas também na manutenção da rede, devido a possíveis mudanças em decorrência de falhas e inserção de novos nós sensores.

Com a finalidade de manter o funcionamento da rede e economizar recursos, especialmente a energia, a utilização de mecanismos para manutenção e prevenção a falhas são extremamente importantes em protocolos de comunicação para RSSF. Dessa forma, diversos mecanismos e protocolos têm sido propostos, tais como,

SPIN (HEINZELMAN; KULIK; BALAKRISHNAN, 1999), Directed Diffusion (INTANAGONWIWAT; GOVINDAN; ESTRIN, 2000), LEACH (HEINZELMAN; CHANDRAKASAN; BALAKRISHNAN, 2000), MULTI (FIGUEIREDO; NAKAMURA; LOUREIRO, 2004), REMUDA (SUN; YAN; BI, 2006), SSR (BABBITT et al., 2008), protocolos baseados na aplicação (AMATO; CARUSO; CHESSA, 2009) e outros.

Muitos dos algoritmos propostos foram desenvolvidos inicialmente para serem utilizados em qualquer RSSF, independente da aplicação ou do tipo de tráfego da rede. Entretanto, de acordo com Akkaya e Younis (2005), o comportamento e o desempenho dos algoritmos de roteamento podem ser diferentes de acordo com a aplicação e do tipos de tráfego gerado na rede. Além disso, Ramasubramanian, Haas e Sirer (2003) afirmam que um desafio em redes *ad hoc* de um modo geral é o desenvolvimento de protocolos que adaptam seu comportamento a mudanças que possam ocorrer na rede. Dessa forma, algoritmos que se adaptam ao comportamento da aplicação passaram a ser desenvolvidos como o SPIN (HEINZELMAN; KULIK; BALAKRISHNAN, 1999), o LEACH (HEINZELMAN; CHANDRAKASAN; BALAKRISHNAN, 2000), MULTI (FIGUEIREDO; NAKAMURA; LOUREIRO, 2004) e outros. Entretanto, protocolos como o SPIN e o LEACH dependem da aplicação modelar seu funcionamento, por meio da criação de mecanismos específicos como formato de dados próprios na aplicação. Já o MULTI apenas alternam a utilização de dois protocolos já estabelecidos

Devido a estas restrições diversos mecanismos como escolha de caminhos, descoberta de vizinhos, tolerância a falhas, roteamento e controle de topologia têm surgido para melhorar o funcionamento de mecanismos de comunicação para RSSF com o objetivo de maximizar o tempo de vida da rede (MAO; HOU, 2007; DRESSLER, 2008; MOUSTAPHA; SELMIC, 2008; ANASTASI et al., 2009; KOHVAKKA et al., 2009; HE; LEE; GUAN, 2009; MARKHAM; TRIGONI, 2010; MIN et al., 2010; LUNG; ZHOU, 2010; GHICA et al., 2010).

Devido a comportamentos diferentes dos protocolos de comunicação, passou-se a classificá-los de acordo com o tipo de tráfego na rede e também de acordo com o funcionamento da aplicação (TILAK; ABU-GHAZALEH; HEINZELMAN, 2002; AKKAYA; YOUNIS, 2005). As aplicações possuem basicamente dados contínuos, dados solicitados pela aplicação quando necessário (baseados em eventos ou no observador) e dados híbridos, quando mais de um tipo é possível. Do ponto de vista do fluxo de dados, o mesmo pode ser por meio de *flooding*, *unicast* ou *multicast*.

Neste sentido, para mecanismos que funcionem de maneira híbrida são ne-

cessários mecanismos e funcionamentos que se configurem automaticamente de acordo com as ocorrências e eventos da RSSF no momento. Uma das alternativas neste sentido pode ser a Tecnologia Adaptativa, cujos dispositivos adaptativos são capazes de mudar seu comportamento em resposta a estímulos de entrada sem a necessidade de ações de agentes externos (NETO, 2001). É importante observar que estas mudanças são possíveis após a configuração das regras iniciais de tais dispositivos.

O primeiro dispositivo e o mais conhecido definido foi o Autômato Adaptativo proposto por Neto (2001) e uma das principais aplicações de TA é o processamento de linguagens, como apresentado em (RICCHETTI, 2005; MATSUNO, 2006; MORAES, 2006; DOI, 2007; FREITAS, 2008; SHIBATA, 2008; PELEGRINI, 2009; JUNIOR, 2009; CHAER, 2010).

Entretanto, devido a sua capacidade de processamento e a possibilidade de mudança de comportamento de maneira automática, a TA vem sendo cada vez mais utilizada em outras áreas como Biotecnologia (PISTORI; SOUZA, 2010), Robótica (SOUSA, 2006), Reconhecimento de Padrões (PEDRUZZI; JR.; NETO, 2009; TCHEMRA; CAMARGO, 2009; GANZELI et al., 2010; CAMARGO; RAUNHEITTE, 2010), Tomadas de Decisão em Empresas (TCHEMRA, 2009), Tráfego de Veículos (KOMORI et al., 2009; FERREIRA; AFFONSO; SASSI, 2010), dentre outras.

Na área de RSSFs, em muitas aplicações não é possível que se tenha acesso à rede mesmo quando ocorram mudanças e a necessidade de mecanismos automáticos é inerente ao comportamento da rede, o que faz com que a utilização de TA passe a se tornar uma alternativa no mapeamento do comportamento da rede.

Assim, sempre que mudanças nos eventos e no comportamento da RSSF ocorreram, decisões podem ser tomados no sentido de manter a conectividade e evitar que a comunicação na rede apresente falhas. Dentre os dispositivos de tomada de decisão podem ser citadas as Tabelas de Decisão Adaptativas (TDA) e Tabelas de Decisão Adaptativas Estendidas (TDAE), propostas por Neto (2001) e Tchemra (2009), respectivamente.

1.1 Objetivos

Este trabalho de pesquisa tem como objetivo principal o desenvolvimento de um algoritmo para roteamento e manutenção da comunicação em uma Rede de Sensores Sem fio (RSSF) com tipo de tráfego híbrido que possa se adaptar ao

comportamento da rede por meio da utilização de Tabela de Decisão Adaptativa (TDA) e Grafos Dinâmicos, visando manter a conectividade da rede mesmo que falhas ou eventos não previstos ocorram.

A ideia principal do trabalho é utilizar uma Tabela de Decisão Adaptativa (TDA) para garantir que as mudanças na rede, possam ser detectadas e corrigidas, além de possibilitar que situações não previstas possam ser tratadas de maneira adaptativa através de criação de novas regras, sem que haja intervenção do usuário. É importante ressaltar que as regras iniciais da TDA devem ser configuradas pelo usuário e que o algoritmo não contempla mecanismos de detecção de falhas, nem mecanismos para modelagem do consumo de energia da RSSF.

Ao contrário da maioria dos protocolos de comunicação dinâmicos, além de permitir mudanças na topologia da rede, por meio de funcionamento já conhecido, este trabalho propõe um algoritmo que permite que mudanças e eventos não previstos possam ser tratados por meio de inserção de novas regras utilizando Tabelas de Decisão Adaptativas.

Para alcançar tais objetivos foi necessário o estudo sobre os principais aspectos relacionados à RSSF, bem como sua arquitetura de comunicação, protocolos existentes, possíveis aplicações, formas de comunicação, com a identificação dos principais fatores que podem influenciar no funcionamento e, especialmente, na comunicação entre os diversos dispositivos que compõem uma RSSF.

Além disso, como o trabalho foi baseado na aplicação da Tecnologia Adaptativa (TA), também se fez necessário o estudo detalhado dos dispositivos adaptativos, conhecendo bem o seu funcionamento, possíveis operações e aplicabilidade.

1.2 Justificativa

A tecnologia de Redes de Sensores Sem Fio foi apontada como uma das dez tecnologias emergentes que podem mudar o modo de vida das pessoas, incluindo a computação, medicina, e outras áreas, segundo a revista MIT Enterprize Technology Review (MIT, 2007). Neste sentido, a cada ano que passa, surgem novas aplicações (ARDIZZONE et al., 2005; FERNÁNDEZ-LUQUE et al., 2006; AZIZ et al., 2007; WARK et al., 2007; ZERGER et al., 2010) que utilizam RSSF para obter resultados com maior acurácia e, em alguns casos, resultados que antes era impossíveis ou muito difíceis de serem obtidos devido à dificuldade de acesso ao local a ser monitorado.

Entretanto, devido ao tamanho reduzido dos nós sensores, existem limitações

de processamento, armazenamento e comunicação. Além disso, na transmissão de dados em uma RSSF, ao contrário do que acontece nas redes tradicionais, além de se preocupar com o envio de informações, os protocolos de comunicação devem levar em consideração também o consumo de energia, já que na maioria das vezes os nós sensores são alimentados por baterias, que possuem carga limitada (CULLER; ESTRIN; SRIVASTAVA, 2004).

Outra característica que deve ser levada em consideração em uma RSSF é a capacidade de auto-organização. Dessa forma, ao serem espalhados em grandes quantidades em uma área de monitoramento, os nós sensores devem se organizar automaticamente, para que seja estabelecida a comunicação entre os diversos dispositivos da rede (BARONTI et al., 2007). É importante ressaltar que a auto-organização é necessária, não somente no instante em que a rede está sendo criada, mas também na manutenção da rede, devido a possíveis mudanças em decorrência de falhas e inserção de novos nós sensores.

Por estas razões, o desenvolvimento de algoritmos para comunicação em RSSF é um dos aspectos de bastante relevância para a área de RSSF. Dentre os principais problemas relacionados à comunicação, podem ser citados: manutenção de topologia em RSSF, cobertura da área a ser monitorada e redução de falhas, dentre outros. Neste sentido, a adoção de mecanismos para manutenção e prevenção a falhas são extremamente importantes em algoritmos de comunicação para RSSF. Como a topologia de uma RSSF costuma ser dinâmica e pode ser modelada em uma árvore ou grafo, a Tecnologia Adaptativa se mostra bastante favorável para ser aplicada, possibilitando que as mudanças ocorridas na rede possam ser mapeadas em dispositivos adaptativos, conforme proposta apresentada em (GONDA et al., 2009; GONDA; CUGNASCA; NETO, 2010).

1.3 Organização do Texto

O restante do texto está organizado da seguinte forma. No Capítulo 2, são apresentados os aspectos relacionados à tecnologia de Redes de Sensores Sem Fio, bem como características relacionadas ao controle e manutenção da topologia; no Capítulo 3, são mostradas as definições de dispositivos adaptativos, grafos dinâmicos, tabela de decisão adaptativa e tabela de decisão adaptativa estendida; a integração entre as áreas de Redes de Sensores Sem Fio e Tecnologia Adaptativa é mostrada no Capítulo 4, no qual é proposto um algoritmo para manutenção de topologia de Redes de Sensores Sem Fio usando Tabela de Decisão Adaptativa Estendida, com a finalidade de garantir que novas regras critérios possam ser

incorporados sempre que eventos não previstos ocorram. Por fim, no Capítulo 5 são apresentadas as contribuições desta tese e propostas de continuidade do trabalho.

2 Redes de Sensores Sem Fio

Conforme apresentado no Capítulo 1, o uso de Redes de Sensores Sem Fio (RSSF) está se tornando cada vez mais freqüente em diversas aplicações e os avanços tecnológicos têm permitido que dispositivos cada vez menores com capacidade de sensoriamento, processamento e comunicação possam ser construídos com a finalidade de monitoramento.

Em geral, os recursos dos nós sensores são limitados, ou seja, os nós sensores possuem restrição no processamento e no armazenamento, além de possuírem largura de banda restrita na comunicação. Conseqüentemente, os nós sensores devem ser agrupados para executarem tarefas de maneira colaborativa, pois individualmente não são muito eficazes. Dessa forma, podem ser obtidas diversas informações a respeito de fenômenos físicos dos ambientes a serem monitorados. Ao serem despejados em grandes quantidades em uma área de monitoramento, os nós sensores podem se organizar automaticamente, para que seja formada uma rede do tipo *ad hoc* para que seja possível a comunicação entre os diversos dispositivos da rede (BARONTI et al., 2007). A comunicação entre os nós sensores e usuários remotos é realizada por meio de um nó sensor especial, denominado de sorvedouro (*sink node*). Assim, um usuário remoto pode enviar comandos e receber dados da RSSF por meio deste nó.

Este capítulo está organizado da seguinte forma: a Seção 2.1 apresenta inicialmente características gerais relacionadas às RSSFs; na Seções 2.3 e 2.4 são apresentadas algumas aplicações nas quais vem se destacando o uso de RSSF e uma breve descrição do problema de consumo de energia, respectivamente. Na Seção 2.7 são apresentados alguns algoritmos para roteamento em RSSF. Por fim, na Seção 2.5 são apresentados aspectos e algumas abordagens relacionadas ao controle de topologia.

2.1 Principais Características das Redes de Sensores Sem Fio

As RSSFs são redes compostas de uma grande quantidade de dispositivos conectados por meio de uma comunicação sem fio do tipo *ad hoc*, ou seja, não necessitam da existência de uma infra-estrutura de comunicação previamente estabelecida. Em uma rede *ad hoc*, os dispositivos trocam dados diretamente entre si, ao contrário do que acontece nas redes de dados tradicionais, como a Internet, por exemplo, na qual dispositivos dedicados de roteamento devem ser utilizados para garantir a comunicação.

Entretanto, o que diferencia uma RSSF de uma rede móvel *ad hoc* é que os elementos computacionais que compõem a rede móvel provêem um suporte para comunicação entre os diversos dispositivos, que podem executar tarefas distintas, enquanto em uma RSSF a execução de tarefas depende de todos os nós sensores, que atuam de maneira colaborativa na execução de cada tarefa, sendo que todos os nós são alocados para executarem a mesma tarefa (LOUREIRO et al., 2003).

Para que isso seja possível, as RSSFs utilizam, em geral, uma comunicação *multihop*, que prevê que as informações possam passar por um ou mais nós intermediários até chegarem ao destino. Este tipo de comunicação é apresentado na Figura 1.

Figura 1: Exemplo de comunicação *multihop*.

Além disso, na maioria das aplicações, os nós sensores devem funcionar por longos períodos de tempo e, como em geral a energia dos nós sensores advém de pequenas baterias, existe limitação na disponibilidade da mesma, o que dificulta também o desenvolvimento de protocolos de comunicação para RSSF, que devem se preocupar não somente com a transferência de dados, mas também com a minimização do consumo de energia ao executar operações de envio e recebimento de dados. Além disso, a forma de comunicação e o envio de dados em uma

RSSF, ao contrário do que acontece nas demais redes de comunicação, depende muito da aplicação. Em geral, as aplicações necessitam de uma grande quantidade de nós sensores e, como a manutenção e configuração manual de muitos dispositivos pequenos é muito trabalhosa, as RSSFs devem possuir capacidade de auto-organização, além de proverem mecanismos de gerenciamento da rede como um todo, e não apenas de nós individuais.

Logo, os fatores citados acima fazem com que os protocolos de comunicação e algoritmos distribuídos utilizados em redes tradicionais e nas demais redes sem fio *ad hoc* não possam ser aplicados às RSSFs, pois estas necessitam de algoritmos e protocolo de comunicação especiais (LOUREIRO et al., 2003). Dentre as razões para tal restrição podem ser citados (AKKAYA; YOUNIS, 2005): não é possível definir um mecanismo de endereçamento global para todas as RSSFs como ocorre na Internet; ao contrário do que acontece nas aplicações de redes tradicionais, quase todas as aplicações que utilizam RSSFs necessitam que o fluxo dos dados coletados tenha múltiplas fontes de origem e um único nó destino (nó sorvedouro); há bastante redundância nos dados coletados, já que muitos sensores apresentarão os mesmos resultados para as mesmas grandezas e esta redundância pode ser aproveitada pelos protocolos de comunicação.

2.2 Considerações sobre Comunicação em RSSF

Apesar do desenvolvimento de algoritmos de comunicação para RSSF, em especial algoritmos para roteamento, possuir diversos desafios, muitos novos protocolos estão sendo propostos para tal finalidade (AKKAYA; YOUNIS, 2005). Além dos protocolos, diversas classificações e aspectos são levados em consideração. Nesta seção serão apresentados alguns aspectos e classificações dos algoritmos de comunicação para RSSF.

Do ponto de vista da aplicação o modelo de entrega de dados em uma RSSF pode ser classificado em (TILAK; ABU-GHAZALEH; HEINZELMAN, 2002):

- Contínuo: os sensores enviam dados continuamente a uma taxa previamente estabelecida;
- Baseado em Eventos: os sensores enviam dados somente se algum evento ocorrer na rede. Neste modelo, o usuário está interessado em monitorar um fenômeno ou um conjunto de fenômenos específicos;
- Baseado no Observador: os sensores enviam dados em resposta a uma con-

sulta realizada pelo usuário na rede;

- Híbrido: este modelo pode conter a combinação de dois ou mais tipos.

Em relação ao fluxo de dados dos nós sensores até o nó sorvedouro, os modelos de entrega de dados são classificados de maneira diferente, e levam em consideração a maneira como o roteamento dos pacotes é realizado na rede. Entretanto, é importante observar que cada um dos modelos citados acima pode se encaixar em um dos seguintes modelos (TILAK; ABU-GHAZALEH; HEINZELMAN, 2002):

- *Flooding*: neste modelo um nó sensor envia, por meio de *broadcast*, os dados para cada um de seus vizinhos, que por sua vez propagam esta informação por *broadcast*, até que os dados cheguem ao nó sorvedouro. Esta abordagem não é afetada por alterações na topologia da rede, porém, o *overhead* de comunicação é alto devido ao número de mensagens de *broadcast*. Um das formas de reduzir este *overhead* é utilizar a agregação de dados (HEINZELMAN; KULIK; BALAKRISHNAN, 1999; INTANAGONWIWAT; GOVINDAN; ESTRIN, 2000).
- Unicast: neste modelo os dados são enviados diretamente de um nó sensor para outro (*multihop*), até que cheguem ao nó sorvedouro.
- Multicast: neste modelo os sensores formam grupos e os dados são enviados através dos grupos formados. Dessa forma, sempre que um usuário deseja saber uma informação da rede, é possível consultar informações em um sensor do grupo e obter as informações necessárias.

Uma outra classificação, apresentada por Akkaya e Younis (2005), utiliza como parâmetro a camada de rede e a forma como os nós sensores são organizados na rede. Assim, os protocolos podem ser classificados em: Protocolos Centrados em Dados, Protocolos Hierárquicos e Protocolos Baseados em Localização.

2.2.1 Protocolos Centrados em Dados

Neste tipo de protocolo assume-se que não é possível a manutenção de um mecanismo de endereçamento global na rede e que todos os sensores são capazes de fazer o sensoramento e enviar os seus dados até o nó sorvedouro. Dessa forma, sempre que uma informação é desejada, o nó sorvedouro faz uma consulta aos sensores de uma determinada região e aguarda pelas informações (AKKAYA;

YOUNIS, 2005). Dentre os protocolos que utilizam esta abordagem podem ser citados o SPIN (HEINZELMAN; KULIK; BALAKRISHNAN, 1999), Directed Diffusion (INTANAGONWIWAT; GOVINDAN; ESTRIN, 2000) e outros que utilizam o mesmo conceito ou variações, tais como os propostos em (BRAGINSKY; ESTRIN, 2002; CHU; HAUSSECKER; ZHAO, 2002).

2.2.2 Protocolos Hierárquicos

Uma outra abordagem utilizada em algoritmos de roteamento para RSSF é a utilização de *clusters*. O objetivo principal de utilizar *clusters* é executar a agregação e fusão de dados a fim de reduzir o número de mensagens enviadas ao nó sorvedouro e, conseqüentemente, reduzir o consumo de energia na rede (AKKAYA; YOUNIS, 2005). Uma das vantagens desta abordagem é que ela permite uma maior escalabilidade em relação a redes com um único sorvedouro quando o número de nós sensores e a área a ser monitorada é muito grande.

A formação de um *cluster*, em geral, leva em consideração a energia disponível no nó sensor ou a proximidade do nó com o nó cabeça do *cluster* (*cluster head*). Um dos primeiros protocolos hierárquico desenvolvido para RSSFs foi o LEACH (HEINZELMAN; CHANDRAKASAN; BALAKRISHNAN, 2000), que serviu de base para inúmeros outros protocolos hierárquicos, tais como APTEEN (MANJESHWAR; AGRAWAL, 2002), PEGASIS (LINDSEY; RAGHAVENDRA, 2002). Além disso, protocolos não baseados no LEACH e que também utilizam *clusters* foram desenvolvidos, tais como (SUBRAMANIAN et al., 2000; LINDSEY; RAGHAVENDRA; SIVALINGAM, 2001).

2.2.3 Protocolos Baseados em Localização

Em uma RSSF, a localização dos nós sensores pode ser importante para determinar a distância entre os nós, para que o consumo de energia possa ser estimado e possa auxiliar no roteamento de dados com a finalidade de economia de energia na comunicação. Outra vantagem de se saber a localização dos nós sensores é a possibilidade de executar consultas na rede em locais específicos (AKKAYA; YOUNIS, 2005). Dentre os protocolos que utilizam esta abordagem podem ser citados MECN (RODOPLU; MENG, 1999), SMECN (LI; HALPERN, 2001), GAF (XU; HEIDEMANN; ESTRIN, 2001) e GEAR (YU; GOVINDAN; ESTRIN, 2001).

2.3 Aplicações

Devido à evolução e constante desenvolvimento das RSSFs, o conjunto de aplicações nas quais ela tem sido utilizadas tem crescido continuamente, sendo possível encontrar novas aplicações em diversas áreas, tais como: medicina, agrícola, controle de ambientes, militar, segurança, dentre outras.

2.3.1 Monitoramento de Ambientes

A área de controle de ambientes, é uma das áreas na qual as RSSFs têm sido largamente utilizadas, devido à necessidade de monitoramento de tais locais.

Em Potter et al. (2005) é apresentada uma aplicação de monitoramento de ambientes hostis, de difícil acesso, e que não possibilitam ou permitem o acesso humano, pois a área a ser monitorada é de preservação ambiental. Nestas aplicação, é possível que os pesquisadores obtenham amostras e dados a distância do ambiente em taxas e distâncias que não eram possíveis sem a tecnologia de RSSF.

Uma outra aplicação que necessita de uma RSSF é apresentada em (ARDIZZONE et al., 2005), na qual um sítio arqueológico importante da Itália é monitorado para obter informações do comportamento dos visitantes, com a finalidade de preservar o sítio contra danos ou até mesmo acidentes. A necessidade de se utilizar RSSFs existe devido à grande área a ser monitorada, o que torna impraticável a utilização de veículos ou de sensores tradicionais. Nesta aplicação as RSSFs são utilizadas em conjunto com câmeras que capturam as imagens e as transmitem por meio da RSSF.

2.3.2 Monitoramento de Pacientes

Na área médica, existem diversas aplicações no monitoramento de pacientes. Em Aziz et al. (2007) é apresentado um sistema de monitoramento de pacientes em processo de recuperação pós-operatória de cirurgias abdominais. A idéia é permitir que os pacientes em casa tenham sinais vitais monitorados, com a finalidade de evitar a ocorrência de feridas, abscessos, má nutrição e outros problemas. O sistema apresentado é utilizado de maneira pervasiva, evitando danos ao paciente.

Em Bisiani et al. (2009) é apresentada uma proposta de uma abordagem lógica para cuidados com a saúde em um ambiente *home care* com auxílio de uma RSSF. A proposta realiza o monitoramento constante e pervasivo do paciente, com o objetivo de prover uma maior autonomia e melhorar a sua qualidade de vida. O

sistema lógico é responsável pelo processamento das informações coletadas por meio da RSSF e verificação da situação do paciente.

Uma outra aplicação na área de saúde é o e-Diab, que consiste em um sistema de monitoramento, assistência e educação de pacientes com Diabetes (FERNÁNDEZ-LUQUE et al., 2006). O sistema é composto por um nó central (PDA ou Celular) para a transmissão das informações médicas. Os sensores de glicose são interligados ao nó central por meio de Zigbee ou Bluetooth e a comunicação entre o nó central e o servidor é realizada por meio de GPRS/GSM.

2.3.3 Outras Aplicações

Galetto e Pralio (2010) apresentam um sistema denominado de Mobile Spatial coordinate Measuring System (MScMS), que tem como objetivo realizar medição de objetos médios e grandes. O sistema apresentado é composto por três componentes principais: uma RSSF distribuída pela área de trabalho, um dispositivo portátil para executar as medições, e um sistema responsável por processar os dados coletados, determinar a geometria e características do objeto, e representá-lo em um ambiente virtual. O trabalho apresenta ainda uma estratégia de posicionamento dos dispositivos para que a aplicação funcione adequadamente.

Em Bai, Atiquzzaman e Lilja (2004) é apresentada uma proposta de utilização de RSSFs no monitoramento de motores de aeronaves. Neste trabalho são descritas a arquitetura de cada nó sensor, que é alimentado por energia do próprio avião, e estratégias de roteamento e posicionamento dos nós sensores.

Em Camilli et al. (2007) é apresentada uma aplicação simulada para Agricultura de Precisão, na qual os sensores são utilizados para coleta de dados, e vários métodos foram analisados para verificação da aplicação. Além disso, são apresentados aspectos tecnológicos e de codificação da aplicação que poderiam ser melhorados, com a finalidade de melhorar os resultados em aplicações que envolvem Agricultura de Precisão.

Em Choi et al. (2004) é apresentada a aplicação de RSSF utilizando Bluetooth para detecção de invasão em prédios. Nesta aplicação existem diversos nós sensores e nós de retransmissão. Todos os nós sensores estão conectados por meio dos nós retransmissores ao nó de controle, que está diretamente ligado a um sistema de controle local, que está conectado a um sistema de segurança global. Sempre que um nó detecta um evento, o mesmo comunica a sua ocorrência ao nó de controle, por meio dos nós retransmissores.

2.4 Consumo de Energia

Uma das principais restrições das RSSF e que em sido fruto de diversas pesquisas é o consumo de energia. Neste sentido, protocolos de comunicação que se preocupam com esse consumo têm sido propostos e desenvolvidos, com o intuito de reduzi-lo e aumentar o tempo de vida da rede (MARGI, 2006). Além da restrição de energia, em diversas aplicações os nós sensores são alimentados por baterias que não podem ser trocadas ou recarregadas (LANDSIEDEL; WEHRLE, 2005; ANASTASI et al., 2009).

Apesar do tipo de nó influenciar no consumo de energia, como mostrado em Raghunathan et al. (2002), as seguintes características, em geral, são válidas (ANASTASI et al., 2009):

- o gasto com transmissão e recepção de dados é bem maior que o gasto com processamento de informações;
- se o rádio estiver executando as operações de recepção e transmissão ou ligado no modo inativo o consumo de energia pode ser elevado, enquanto no modo *sleep* esse consumo é reduzido. Logo, o rádio deve ser colocado em modo *sleep* sempre que possível;
- dependendo da aplicação, as operações de sensoriamento podem ser consideradas como uma importante fonte de consumo de energia, portanto, a redução no consumo de energia para tais operações deve ser considerada.

De acordo com Anastasi et al. (2009), as técnicas para redução do consumo de energia podem ser divididas em:

- *Duty cycling*: esta técnica é focada no sistema de comunicação e pode ser dividida em técnicas de controle de topologia e de gerenciamento de energia. Dentre as técnicas de controle de topologia a decisão de ligar e desligar nós sensores pode ser baseada na localização ou na conectividade da rede. Basicamente, esta técnica consiste em colocar os nós sensores em modo *sleep* sempre que nenhuma comunicação é necessária. Assim, os nós sensores alternam entre os modos ativo e *sleep*. Esta alternância é denominada de *duty cycling*, e a fração de tempo que os nós ficam ativos durante seu tempo de vida é denominado de *duty cycle*. Portanto, algoritmos para o escalonamento de modos de operação dos nós sensores devem ser utilizados para determinar em qual instante de tempo haverá a transição de um modo

de operação para outro. Dentre os protocolos que adotam esta técnica podem ser citados o *Geographical Adaptive Fidelity* (GAF) (XU; HEIDEMANN; ESTRIN, 2001) e o Span (CHEN et al., 2002);

- Abordagem Guiada por Dados (*data-driven approach*): esta técnica consiste em manter a acurácia do sensoriamento em um nível aceitável, mesmo reduzindo a quantidade de dados sensorizados. Para alcançar uma redução considera-se que o sensoriamento pode impactar no consumo de energia do nó de duas formas: amostras desnecessárias (informações redundantes são obtidas e enviadas na rede) e consumo de energia do sensoriamento (a frequência do sensoriamento é elevada). Dentre os protocolos que utilizam esta técnica pode ser citado o SAF (TULONE; MADDEN, 2006);
- Mobilidade: em alguns casos a mobilidade pode auxiliar na redução do consumo de energia. Em redes nas quais os nós são estáticos, alguns nós tendem a consumir mais energia com comunicação do que outros por fazerem parte de um caminho até o nó sorvedouro. Uma possibilidade é permitir que nós móveis façam a captação de dados dos sensores estáticos, evitando que nós intermediários tenham sua energia dissipada com a propagação de informações. Quando não há a possibilidade de utilizar nós móveis (por exemplo, devido ao seu custo) é possível ainda anexar nós estáticos em entidades que podem se movimentar, como veículos ou animais. Dentre os protocolos que utilizam esta abordagem estão o *Greedy Maximum Residual Energy* (GMRE) (BASAGNI et al., 2008), o *Two-Tier Data Dissemination* (TTDD) (LUO et al., 2005).

2.5 Construção de Topologia

A topologia em uma RSSF, ao contrário do que acontece nas redes de computadores tradicionais, pode ser alterada dinamicamente, devido a diversos fatores, como falha de nós, inserção de novos nós e alterações no comportamento da rede. Neste sentido, mecanismos de controle de topologia em uma RSSF são importantes principalmente para aumentar o tempo de vida da rede ou aumentar a capacidade da rede para possíveis adições de novos nós, sem que a conectividade da rede seja afetada (SANTI, 2005; BARONTI et al., 2007). Em geral, a topologia em uma RSSF é determinada por um grafo $G = (V, E)$, no qual V é o conjunto de todos os nós da rede e E é o conjunto de todas as arestas $(v_i, v_j) \in E \subseteq V^2$. Entretanto, como nem todos os nós precisam estar ativos, em geral o que se utiliza

para representar a rede é um sub-grafo $T = (V_T, E_T)$ de G , sendo que $V_T \subseteq V$ e $E_T \subseteq E$ (BARONTI et al., 2007).

2.5.1 Taxonomia para Controle de Topologia em Redes *Ad Hoc*

O controle de topologia é um problema bastante comum em redes *ad hoc* móveis e em RSSFs. Dessa forma, diversas abordagens, considerando diversos aspectos da transmissão de dados, são levados em consideração, como apresentado na Figura 2.

Figura 2: Uma Taxonomia para Controle de Topologia.

Na taxonomia apresentada na Figura 2, proposta por (SANTI, 2005), o modelo mais simples é o de controle de topologia homogênea, no qual todos os nós possuem o mesmo alcance de transmissão. Neste caso, é importante determinar o valor mínimo do alcance r para garantir o funcionamento da rede, denominado de Alcance de Transmissão Crítico (ATC). No controle de topologia não homogêneo (heterogêneo), cada nó transmissor pode ter um alcance diferente e, de acordo com o tipo de informação utilizado para construir a topologia, podem existir três categorias. O controle baseado em localização assume que a localização dos nós é conhecida, o que nem sempre é possível em uma RSSF. No controle baseado em direção, apesar de a posição dos nós ser desconhecida, cada um dos nós é capaz de determinar a direção dos seus vizinhos. Por fim, na abordagem baseada em Vizinhança, os nós conhecem a identificação (ID) de seus vizinhos.

2.5.2 Um Modelo de Controle de Topologia para RSSF

Para que o controle de topologia seja eficiente, podem ser utilizadas diversas métricas, tais como conectividade, consumo de energia, vazão da rede, *overhead* da rede, dentre outras. Embora existam diversas métricas, a mais importante delas é a conectividade, pois a mesma garante que as informações sejam trocadas

entre os nós da rede (SANTI, 2005).

Dada uma RSSF, pode-se chamar de r_{max} o alcance máximo da transmissão, e a função $RA : V \rightarrow (0, r_{max}]$ como a função de atribuição de alcance. O parâmetro r_{max} depende do rádio a ser utilizado que, em geral, é o mesmo utilizado em todos os nós sensores (SANTI, 2005).

Segundo Santi (2005) apud Rapport (2002), dados dois nós sensores i e j espalhados em um ambiente a ser monitorado, para que um nó i possa transmitir dados de maneira correta para um nó j , a potência p_i do sinal deve obedecer à inequação 2.1:

$$p_i \geq \beta \delta_{i,j}^\alpha \quad (2.1)$$

na qual $\alpha \geq 2$ é o gradiente de potência da distância, e $\beta \geq 1$ é o parâmetro de qualidade da transmissão e $\delta_{i,j}$ é a distância Euclidiana dos nós. O valor de β geralmente é configurado para 1, e o parâmetro α em condições ideais (sem nenhuma interferência) é configurado para 2. Entretanto, um valor entre 2 e 6 é aceitável e o valor geralmente utilizado é 4. É importante ressaltar que a inequação eq. (2.1) que define p_i considera apenas o consumo de energia no envio de dados, e desconsidera a presença de ruídos, refração, reflexão. Apesar disso, este valor é aceitável para redes *ad hoc*, como uma RSSF. Dessa forma, pode-se definir o consumo de energia para uma função de atribuição RA , como

$$c(RA) = \sum_{i \in N} (RA(i))^\alpha. \quad (2.2)$$

Entretanto, além de saber o consumo toda a rede, é importante determinar o *Aggregation Time Control* (ATC) da rede, que em uma RSSF deve ser o menor valor possível para garantir conectividade da rede. Este fator é importante para definir questões como o número mínimo de nós necessários para monitorar uma determinada região $R = [0, l]^d$, $d = 1, 2, 3$. Para determinar o ATC, é importante obter informações sobre o local no qual os nós sensores serão despejados para monitoramento. Caso as condições sejam bem conhecidas e não haja obstáculos, o ATC pode ser determinado como é o comprimento da maior aresta (maior distância Euclidiana entre dois nós) da árvore geradora mínima do grafo. Porém, na maioria dos casos, a posição dos nós pode não ser fácil de ser determinada e, nestes casos, a ATC é aproximadamente $l\sqrt{d}$, segundo Santi (2005).

2.6 Manutenção de Topologias em RSSF

A manutenção de topologia em RSSF consiste na montagem da topologia inicial e na reconstrução da topologia sempre que necessário. Para a construção de topologias iniciais, diversas propostas foram elaboradas. Dentre as diversas topologias, a topologia em árvore (que pode ser representada por um grafo direcionado), também conhecida como *routing tree* é muito utilizada, pois além de ser simples é eficiente. Em (ZHOU; KRISHNAMACHARI, 2003) é apresentada uma avaliação de quatro estratégias para a construção de uma topologia em árvore. A primeira delas é denominada de *Earliest-First*, que consiste na escolha do primeiro nó candidato, é escolhido como pai. Na segunda abordagem, denominada de *Nearest-First* um nó seleciona o candidato mais próximo como seu pai na árvore. Outras duas abordagens selecionam o nó pai aleatoriamente. A primeira delas, escolhe um dos nós entre todos os possíveis pais com igual probabilidade para todos. A segunda abordagem, atribui pesos a todos os possíveis pais de acordo com o número de vizinhos, para garantir um balanceamento no número de filhos. Quanto maior o número de vizinhos, menor o peso do possível pai.

Para garantir tolerância a falhas que podem ocorrer em RSSFs é necessário que mecanismos de controle de topologia sejam utilizados. A reconstrução de topologia em uma RSSF pode ser executada ou necessária devido a diversos fatores, como falha de nós, balanceamento de energia ou agregação de dados. Por exemplo, sempre que um nó que é responsável por roteamento de dados falha, os nós que dependem dele para o envio de dados deixam de se comunicar com a rede e, portanto, uma reconstrução de topologia é necessária para que os nós desconectados sejam reconectados, caso seja possível (NAKAMURA, 2007).

Para realizar a reconstrução da topologia, as seguintes abordagens são possíveis (NAKAMURA, 2007):

- Abordagem baseada no nó sorvedouro (*sink*): nesta abordagem o nó sorvedouro é responsável por detectar a necessidade de reconstrução da topologia. Como a reconstrução é global, toda a topologia da RSSF é reconstruída; entretanto, esta abordagem é mais custosa do ponto de vista de comunicação, mas, pode levar à obtenção de melhores topologias, pois o processo de reconstrução envolve todos os nós da rede.
- Abordagem baseada em um nó fonte: nesta abordagem os nós fonte, ou seja, os nós que obtêm dados de sensoriamento para envio ao nó sorvedouro são responsáveis por detectar quando é necessária uma reconstrução de to-

pológia. Neste caso, apenas uma parte dos nós é envolvida na reconstrução e, conseqüentemente, é realizada uma reconstrução parcial da topologia da RSSF. O custo associado a esta abordagem é inferior, entretanto, podem ocorrer falhas na reconstrução, pois a mesma pode não abranger todos os nós necessários.

- Abordagem mista: esta abordagem consiste na combinação dos dois métodos anteriores. Dessa forma, a Abordagem Baseada em um Nó Fonte é utilizada para realizar reconstruções locais da topologia, enquanto periodicamente reconstruções globais são executadas para reduzir problemas devido às reconstruções locais que possam ocorrer.

Um dos problemas relacionado ao Controle de Topologia é detectar em qual momento é necessário realizar uma reconstrução. Em redes nas quais o tráfego é constante, é possível medir o número de pacotes enviados em um determinado intervalo de tempo, ou seja, determinar o nível de tráfego da rede, que deve se manter constante, desde que não haja falha ou inserção de novos nós. Dessa forma, é possível verificar que se nenhum novo nó foi adicionado à rede e falhas ocorreram, o número de pacotes em um determinado intervalo de tempo t_i deve ser menor do que o número de pacotes do intervalo t_{i-1} . Além disso, se novos nós foram adicionados à rede e não ocorreram falhas, o número de pacotes do intervalo de tempo t_i deve ser maior que o número de pacotes do intervalo t_{i-1} .

É importante observar que o impacto no número de pacotes de um intervalo de tempos depende do tipo de nó que falhou. Caso um nó responsável por roteamento falhe, a redução no tráfego da rede, em geral, é maior, pois este nó também realiza a transmissão de diversos outros nós. Se um nó folha da árvore falha, a redução do tráfego se restringe aos pacotes gerados por este nó. A primeira situação é denominada de falha de roteamento e a segunda é denominada de falha periférica (neste caso não é necessária reconstrução de topologia).

Uma outra alternativa é utilizar o envio de pacotes periódicos¹ de controle para verificar se houve falhas de nós na rede. Em (WANG; KUO, 2003) é apresentada uma proposta que propaga as informações coletadas sobre as falhas detectadas por toda a rede. O problema desta abordagem é que a mesma aumenta o número de transmissões realizadas na rede.

¹o envio periódico de pacotes é denominado de *heartbeat*

2.7 Algoritmos para RSSF

Com o avanço em diversas áreas e o funcionamento não uniforme da maioria das aplicações, diversos algoritmos de comunicação foram desenvolvidos para RSSFs. Uma consideração importante é que a maioria dos algoritmos de comunicação desenvolvidos para RSSFs tem como objetivos a troca de informações entre os dispositivos e o consumo reduzido de energia, para prolongar o tempo de vida da rede.

2.7.1 LEACH

O protocolo LEACH (*Low-Energy Adaptive Clustering Hierarchy*) é um dos protocolos hierárquicos mais populares para RSSFs (HEINZELMAN; CHANDRAKASAN; BALAKRISHNAN, 2000). A idéia básica deste protocolo é a formação de *clusters* baseada na força do sinal recebido, e utilizar os cabeças dos *clusters* (*cluster heads*) locais para realizar roteamento até o nó sorvedouro. Dessa forma, apenas os *cluster heads* serão responsáveis por transmitir as informações dos nós sensores.

Todo o processamento dos dados, como agregação e fusão é realizado pelos *cluster heads*. Para realizar um balanceamento de energia, os *cluster heads* são trocados periodicamente de maneira aleatória. O protocolo LEACH é distribuído, não necessita que haja uma visão global da rede, e cada nó transmite diretamente para o nó sorvedouro ou para o *cluster head*. Entretanto, este protocolo não pode ser aplicado para RSSFs espalhadas em regiões muito grandes, e a troca periódica do *cluster head* gera um *overhead* a mais na comunicação (AKKAYA; YOUNIS, 2005).

2.7.2 SPIN

O protocolo SPIN (*Sensor Protocols for Information via Negotiation*) foi proposto por (HEINZELMAN; KULIK; BALAKRISHNAN, 1999), tratando-se de um protocolo centrado em dados, cuja idéia principal é nomear os dados usando descritores de alto nível (ou meta-dados). Antes da transmissão, meta-dados são trocados entre os nós sensores por meio de um mecanismo de anúncio. Cada um dos nós, ao receber um novo dado, propaga-os aos seus vizinhos e os vizinhos que não possuem tal informação devem solicitá-la por meio do envio de uma mensagem de requisição.

Esta negociação inicial faz com que o SPIN não necessite enviar os dados por

meio de *flooding*, reduzindo problemas como recebimento de informações redundantes, sobreposição de áreas de monitoramento, dentre outros (AKKAYA; YOUNIS, 2005). O protocolo considera que o formato dos meta-dados será definido pela aplicação; porém três mensagens são definidas por ele:

- ADV: mensagem destinada a divulgação de um meta-dado.
- REQ: mensagem para a requisição de um dado específico.
- DATA: mensagem para o envio de dados.

Uma das vantagens do SPIN é que as mudanças na topologia são sempre locais, já que é necessário saber apenas qual é o próximo nó. Entretanto, o mecanismo de anúncio de dados não pode garantir a entrega de dados. Também, se os nós sensores interessados em dados estão distantes do nó fonte, e os nós sensores entre os nós fonte e destino não estão interessados em algum dado, estes dados não chegarão ao nó interessado. Este protocolo não é interessante para aplicações que necessitam de confiabilidade na entrega dos dados em intervalos de tempo regulares, como detecção de intrusão (AKKAYA; YOUNIS, 2005).

2.7.3 Directed Diffusion

O protocolo Directed Diffusion (INTANAGONWIWAT; GOVINDAN; ESTRIN, 2000) é um algoritmo centrado em dados cuja idéia principal é difundir os dados pelos nós sensores, utilizando um esquema de nomes para os dados. O objetivo de utilizar um esquema de nomes na aplicação é fazer com que ela não tenha que se preocupar com as operações de roteamento.

O Directed Diffusion sugere o uso de pares atributo-valor para os dados e utiliza estes pares para realizar as consultas aos nós sensores. Antes de realizar uma consulta, é criada uma lista de pares atributo-valor, denominada de interesse, que é propagada pelo sorvedouro a seus vizinhos. Cada um dos nós pode armazenar em *cache* o interesse para uso posterior (AKKAYA; YOUNIS, 2005). Além da lista de pares atributo-valor, um interesse armazena também um *link* para um vizinho do qual o interesse foi recebido, denominado de gradiente, que é composto por taxa de transmissão, duração e tempo de expiração.

A utilização dos gradientes e dos pares são responsáveis pela criação de caminhos entre o nó sorvedouro e os nós fonte. Este processo pode gerar diversos caminhos e o nó sorvedouro é responsável por escolher um deles por meio de reforços, enviando o mesmo interesse pelo caminho selecionado. Caso um caminho

entre um nó sensor e o nó sorvedouro falhe, outro caminho pode ser selecionado pelo reenvio do interesse.

Uma das vantagens do Directed Diffusion é que não há a necessidade de manter um controle de topologia global da rede. Porém, ele é ineficiente em relação a aplicações que precisam de dados contínuos, como monitoramento de ambientes. Assim como no SPIN, os esquemas de dados são dependentes da aplicação.

2.7.4 REMUDA

Remuda é um protocolo de comunicação para RSSF dividido em três etapas: construção da topologia, envio de dados e manutenção da topologia (SUN; YAN; BI, 2006). A topologia utilizada é a de árvore, e o critério principal para a sua montagem é a qualidade do sinal, seguida da distância até o nó sorvedouro (*sink node*) e, por fim, o nó com menor identificação (*Node ID*). Neste protocolo, inicialmente é montada uma topologia temporária; posteriormente são enviados pacotes de controle aos nós para que a topologia da rede se torne estável (árvore geradora). Sempre que uma mudança na topologia ou na qualidade do sinal ocorre, é necessário que haja uma reconfiguração de parte ou de todos os nós, que prevê a eventual escolha de um novo pai para cada nó na árvore.

Para a determinação da qualidade do sinal uma mensagem de verificação de sinal é enviada ao pai periodicamente por cada nó filho. Caso existam mensagens de comunicação, a própria mensagem de dados é utilizada para verificar a comunicação entre um nó e seu pai. A vantagem de utilizar esta abordagem é que ela reduz o número de mensagens de verificação na rede. Entretanto, sempre que um nó verifica que a comunicação com seu pai apresenta problemas, é necessário enviar uma mensagem do tipo *Help*, que inclui sua identificação. Ao receber uma resposta, o nó inicializa um contador de tempo e, quando o contador expira, o nó escolhe como seu pai um dentre todos os nós que responderam à mensagem *Help*.

Para a troca de dados entre os nós sensores existem dois tipos de mensagens: dados urgentes e dados periódicos. Ao receber um dado urgente, um nó deve encaminhá-la imediatamente ao seu pai, para que a mensagem chegue ao nó sorvedouro o mais rápido possível. Quando um dado periódico é recebido, o mesmo é agregado com os dados do próprio nó, para reduzir o número de mensagens da rede.

Com o intuito de testar o protocolo, foram realizados experimentos em uma

plataforma de hardware criada pelos autores da pesquisa, baseada em uma RSSF desenvolvida pela empresa Crossbow (CROSSBOW, 2006), que possui como sistema operacional o TinyOS (TINYOS, 2006). Os resultados experimentais deste trabalho relatados mostram que ele é mais eficiente que o algoritmo *Link Estimation Selection* (LEPS) que é utilizado nos nós da Crossbow (TINYOS, 2006), apesar de possuir a desvantagem de ter o contador de saltos (*hop count*) mais elevado em alguns casos, o que gera um maior número de nós gastando energia durante a transmissão.

2.7.5 MULTI

O mecanismo denominado de Multi corresponde à utilização alternada de dois outros protocolos: *Source-Initiated Dissemination* (SID) e *Earliest First Tree* (EF-Tree). O protocolo SID é utilizado em RSSFs nas quais as aplicações são baseadas em eventos e o tráfego na rede é reduzido. Dessa forma, toda vez que um nó sensor precisa enviar informações coletadas em decorrência de algum evento, deve ser criada uma rota deste nó até o nó sorvedouro (*sink*). Para a construção do caminho, o nó possui informações a serem transmitidas e difunde uma mensagem de *broadcast* na rede. Esta mensagem é propagada pela rede até que chegue ao nó sorvedouro e permita que seja determinada uma rota para o envio de dados.

O protocolo EF-Tree mantém uma estrutura de comunicação em árvore, que é construída a partir do nó sorvedouro, que envia uma mensagem de *broadcast* para a construção da topologia, propagada pelos demais nós da rede. Cada um dos nós sensores da rede seleciona o primeiro nó do qual recebeu a informação de controle, como nó pai na árvore de roteamento. Periodicamente, este algoritmo é executado para a manutenção da rede. O algoritmo Multi proposto em Figueiredo, Nakamura e Loureiro (2004) utiliza alternadamente os algoritmos SID e EF-Tree, de acordo com o estado da rede. Inicialmente, o algoritmo SID é escolhido para transmissão de dados até o nó *sink*. Porém, caso o tráfego de dados na rede se torne muito intenso, de acordo com parâmetros a serem configurados, o algoritmo Multi passa a utilizar o protocolo EF-Tree.

2.8 Considerações Finais Sobre o Capítulo

Este capítulo descreveu brevemente aspectos de importância relacionados às RSSFs. Inicialmente, foram descritas suas principais características, aspectos relacionados à comunicação, ao consumo de energia e controle de topologia, devido à

sua importância no contexto deste trabalho. Algumas áreas de aplicação também são descritas. Por fim, alguns algoritmos de roteamento para RSSF são apresentados.

3 Tecnologia Adaptativa

A Tecnologia Adaptativa permite que dispositivos com estruturas fixas, como autômatos, grafos, tabelas de decisão, entre outros, possam modificar suas estruturas de maneira automática e sem interferência externa, para melhor se adaptar à resolução de um determinado problema.

Este capítulo apresenta alguns dispositivos adaptativos e está organizado da seguinte forma: a Seção 3.1 apresenta o conceito de um dispositivo adaptativo e sua formulação geral; na Seção 3.2 é definido formalmente um grafo dinâmico como um dispositivo adaptativo e mostra uma modificação para que possa ser aplicado em RSSF; as Seções 3.3 e 3.4 mostra a definição de Tabela de Decisão Adaptativa e Tabela de Decisão Adaptativa Estendida.

3.1 Dispositivos Adaptativos

Um dispositivo formal é dito adaptativo sempre que o seu comportamento muda dinamicamente, em resposta a estímulos de entrada, sem a interferência de agentes externos, incluindo seus usuários.

Em sua formulação geral, os dispositivos adaptativos são caracterizados por meio de suas camadas conceituais: o *formalismo subjacente* (ou dispositivo subjacente) e o *mecanismo adaptativo* (ou camada adaptativa) (NETO, 2001). O formalismo subjacente é representado por um dispositivo guiado por regras (formalismo convencional, não-adaptativo) e opera, movendo-se de uma configuração para outra, sucessivamente, em resposta a estímulos recebidos em sua entrada. Tais movimentos são definidos por um conjunto de regras que mapeiam cada possível configuração em uma configuração seguinte correspondente.

O mecanismo adaptativo responde pelas mudanças incrementais no comportamento do dispositivo subjacente. As propriedades de auto-modificação são representadas pelas funções e ações adaptativas, capazes de alterar o conjunto de regras do dispositivo subjacente, determinando assim, conseqüentemente, seu

comportamento independente de decisões externas a partir de então.

Para isso, a cada uma de suas regras de formação, podem ser associadas ações adaptativas, que têm a função de tornar o dispositivo auto-modificável. Em outras palavras, as eventuais mudanças no comportamento do dispositivo devem ser conhecidas, em qualquer fase do funcionamento em que sejam efetuadas.

Portanto, um dispositivo adaptativo deve ser capaz de detectar todas as situações que possam provocar alterações e reagir adequadamente, impondo as alterações correspondentes ao comportamento do dispositivo. O comportamento desta classe de dispositivos formais baseia-se no funcionamento de um formalismo subjacente (não-adaptativo), descrito por algum conjunto finito de regras de formação.

Um dispositivo adaptativo pode ser obtido por ações adaptativas inerentes às regras do formalismo subjacente, de modo a que, sempre que uma regra é aplicada, a ação adaptativa associada a ela é executada, provocando as mudanças correspondentes sobre o conjunto de regras do formalismo subjacente (não-adaptativo).

De acordo com Neto (2001), um dispositivo adaptativo baseado em um formalismo subjacente (não-adaptativo) pode ser definido, de forma geral, como uma 5-upla $ND = (C, NR, S, c_0, A, NA)$, na qual:

- ND : um dispositivo guiado por regras definidos em um conjunto de regras NR ;
- C : conjunto de todas as possíveis configurações, no qual c_0 é a configuração inicial;
- S : o conjunto (finito) de todos os possíveis eventos que podem ser estímulos de entradas válidos, no qual $\varepsilon \in S$;
- ε denota “vazio” e representa o elemento nulo do conjunto que pertence;
- $A \subseteq C$ (*resp.* $F = C - A$): subconjunto de configurações aceitáveis (respectivamente, F é o conjunto das configurações não aceitáveis).
- NA : conjunto finito ($\varepsilon \in NA$) de todos os possíveis símbolos de saída produzidos por ND , como resultado da aplicação de regras de NR .
- NR : conjunto de regras que definem ND por meio da relação $NR \subseteq C \times S \times C \times NA$. Cada regra $r \in NR$ possui o formato $r = (c_i, s, c_j, z)$, na qual $s \in S$ é um estímulo de entrada, r muda a configuração atual de c_i para c_j , consumindo s e tendo como saída $z \in NA$ como efeito.

Além dos conjuntos que definem o dispositivo adaptativo, tem-se também o fluxo de estímulos de entrada $w = w_1 w_2 \dots w_n$, $k = 1, \dots, n$, com $k \in S - \{\varepsilon\}$.

A estrutura geral de um dispositivo adaptativo é apresentada na Figura 3.

Figura 3: Estrutura Geral de Um Dispositivo Adaptativo(TCHEMRA, 2009)

Na Figura 3, o dispositivo subjacente é não-adaptativo e possui um conjunto finito e estático de regras R . De acordo com Tchemra (2009), para representar este conjunto de regras em um algoritmo, utilizam-se estruturas de controle do tipo *se-então*, que podem levar às seguintes situações:

- caso uma única regra de R satisfaça as condições desejadas, o dispositivo executa a ação determinada pela regra (operação determinística);
- caso mais de uma regra de R satisfaça as condições desejadas, o dispositivo executa as diversas possibilidades em paralelo (operação não determinística);
- caso nenhuma regra de R satisfaça as condições desejadas, o dispositivo comunica uma situação de erro e encerra sua operação.

Para que o dispositivo se torne adaptativo, é necessário adicionar funções adaptativas que são associadas a regras do formalismo subjacente. Toda vez que uma função adaptativa é chamada, ações adaptativas são tomadas com o objetivo de alterar o conjunto de regras R do dispositivo. Estas ações podem ser dos seguintes tipos (TCHEMRA, 2009):

- Ação adaptativa elementar de consulta $AD?$: executa uma busca no conjunto de regras do dispositivo para encontrar aquelas que satisfaçam as condições procuradas;

- Ação adaptativa elementar de inclusão $AD+$: executa a adição de uma nova regra ao dispositivo, baseando-se nas condições oferecidas;
- Ação adaptativa elementar de exclusão $AD-$: executa a eliminação de uma regra do conjunto de regras, baseando-se nas condições fornecidas.

Uma observação importante é que uma mesma função adaptativa pode ser associada a mais de uma ação adaptativa e, caso isto ocorra, as ações de consulta $AD?$ e de exclusão $AD-$ são executadas antes da inclusão $AD+$ da mesma função. Além disso, somente duas ações adaptativas podem ser associadas a uma regra R selecionada do formalismo subjacente:

- uma ação adaptativa a ser tomada antes da execução da regra;
- outra ação adaptativa a ser tomada após a execução da regra.

É importante ressaltar que, apesar do Autômato Adaptativo ser o primeiro dispositivo adaptativo a ser apresentado, existem diversos dispositivos convencionais (tais como árvores de decisão, tabela de decisão e outros) que podem se tornar auto-modificáveis por meio da adição de uma camada adaptativa (ações adaptativas) (TCHEMRA, 2009). Em Neto (2007) é apresentada a evolução da Tecnologia Adaptativa e também uma breve descrição de diversos dispositivos adaptativos. Além disso, Pistori (2003) apresentada com mais detalhes a evolução desta tecnologia.

3.2 Grafos Dinâmicos

Na maioria das aplicações de algoritmos em grafos, os grafos são dispositivos estáveis que sofrem pouca ou nenhuma mudança. Entretanto, variações menos estáveis desses dispositivos formais, que possam sofrer alterações em sua estrutura, tais como inserção e remoção de arestas e vértices, vêm sendo alvo de estudos, demonstrando o crescente interesse no desenvolvimento de aplicações que façam uso de formalismos dinâmicos (DEMETRESCU; FINOCCHI; ITALIANO, 2003; DEMETRESCU; ITALIANO, 2006; CHAN; YANG, 2009; DÍAZ; MITSCHKE; PÉREZ-GIMÉNEZ, 2009). Nesta seção, definem-se os grafos dinâmicos com o objetivo de utilizá-los como ferramentas para a modelagem de soluções para problemas relacionados às RSSFs.

Baseado na definição de Neto (2001) apresentada na Seção 3.1, pode-se definir um dispositivo adaptativo que usa um grafo como formalismo subjacente (não-

adaptativo). Embora o dispositivo formal apresentado na Seção 3.1 seja uma 5-upla, um grafo dinâmico para modelar o comportamento de uma RSSF não precisa de todos os conjuntos e símbolos. O conjunto A (que define aceitação) não é utilizado, pois os estímulos e eventos que podem ocorrer na RSSF são infinitos e não há final de entrada, ou seja, a o tamanho da cadeia de estímulos w é infinito. Outros problemas, como controle de máquinas adaptativo também possuem o mesmo comportamento. Além disso, o símbolo vazio ε também não é utilizado no caso de RSSFs, pois não há mudança de comportamento e da topologia da RSSF sem que ocorra um evento na rede.

Portanto, um dispositivo adaptativo que utiliza um grafo direcionado como formalismo subjacente pode ser definido como uma 4-upla

$$G = (C, NR, S, c_0) \quad (3.1)$$

Nesta 4-upla tem-se:

- G : corresponde ao grafo que modela a RSSF em questão;
- C : conjunto de possíveis configurações do grafo que modela a RSSF. c_0 é sua configuração inicial, ou seja, a configuração que é montada inicialmente com base nas regras de conexão da RSSF;
- NR : conjunto de regras que definem o grafo ND pela relação $C \times S \times C$. Neste caso, as regras $r \in NR$ têm a forma $r = (c_i, s, c_j)$, indicando que, em resposta a um estímulo de entrada $s \in S$, a regra r altera a configuração corrente do grafo c_i para configuração c_j . Neste caso, a mudança de configuração representa uma alteração na topologia da RSSF;
- S : conjunto de todos os possíveis eventos válidos e considerados como estímulos de entrada para G . Neste caso, ε também é desconsiderado, pois o estímulo “vazio” não provoca alterações na RSSF. É importante observar que eventos como *timeout* serão tratados como falhas de um nó.

Seja uma topologia em árvore de uma RSSF, um grafo dinâmico G que representa esta topologia tem as seguintes propriedades:

- $V = \{v_1, v_2, \dots, v_n\}$ é o n -upla ordenada de nós da rede e o nó sorvedouro é v_1 ;
- um vértice v_j é pai de um vértice v_i , $1 \leq j \leq n$ e $2 \leq i \leq n$, se e somente

se, $(v_i, v_j) \in E$, na qual E é o conjunto de arestas do grafo (conexões entre nós da rede);

- O nó sorvedouro não possui pai, ou seja, $(v_1, v_i) \notin E$, $2 \leq i \leq n$;
- Cada nó possui apenas um único pai, ou seja, se $(v_i, v_j) \in E$ e $(v_i, v_k) \in E$, então $v_j = v_k$, $2 \leq i \leq n$ e $1 \leq j \leq n$.

O mecanismo adaptativo é composto pelas operações que promovem a modificação da topologia da rede, por meio de atualizações no grafo que representa a rede. Entre os prováveis fatores que podem influenciar estas alterações estão a falha de um nó, o desligamento temporário de um nó (economia de energia), a inclusão de novos nós, os ajustes para melhorar o desempenho da rede (nós com pouca energia diminuem sua capacidade de comunicação), dentre outros. Para realizar a alteração na topologia e nas regras do grafo, serão utilizadas Tabelas de Decisão Adaptativas, descritas na Seção 3.3.

Uma atualização em um grafo pode ser definida como qualquer operação de inserção ou remoção de arestas ou vértices do grafo ou mudança associada aos vértices ou arestas, tais como custo ou cor de vértices.

3.3 Tabelas de Decisão Adaptativas

As Tabelas de Decisão Adaptativas podem ser consideradas uma versão adaptativa das Tabelas de Decisão Tradicionais, que são utilizadas nas mais diversas áreas com o objetivo de auxiliar na descrição de problemas complexos. Segundo Tchemra (2009) apud Hugges, Shank e Stein (1968), as Tabelas de Decisão mostrarem-se também mais eficientes do que outras formas de expressão, tais como narrativas, escritas, diagramas e outras. Neste sentido, é necessário descrever um pouco o funcionamento das Tabelas de Decisão Tradicionais para depois descrever o funcionamento de Tabelas de Decisão Adaptativas.

As tabelas de decisão podem ser utilizadas tanto para documentação de informações (tabelas inertes), como meio de análise de modelos de decisão e seus resultados (tabelas ativas). Quando utilizadas como tabelas ativas, estas podem ser utilizadas como alternativa para problemas compostos por condições, ações e regras (TCHEMRA, 2009):

- as condições são variáveis cujos valores devem ser avaliados para a tomada de decisão;

- as ações representam o conjunto de operações a serem executados condicionadas pelas respostas das condições;
- regras representam o conjunto de situações que são verificadas em resposta às condições para a execução das ações.

Segundo Tchemra (2009) apud Hugghe, Shank e Stein (1968), Gildersleeve (1970), Neto (2001) uma tabela de decisão é tipicamente dividida em quatro partes:

- Linha das condições: corresponde a uma variável do problema que será avaliada no processo de decisão e cujo valor está presente na lista de valores das condições;
- Possíveis valores das condições: neste item são apresentados todos os possíveis valores que as condições podem assumir;
- Linha das ações: conjunto de procedimentos ou operações a serem executadas de acordo com as condições;
- Ações a serem tomadas: as ações a serem tomadas após a avaliação das regras.

Formalmente, Tchemra (2009) define uma Tabela de Decisão como uma tripla $TD = (C, A, R)$, na qual:

- C é o conjunto de condições que representam as variáveis do problema. Se $|C| = n$, então, $C = \{C_i, 1 \leq i \leq n\}$.
- A é a relação de ações que podem ser executadas para solucionar o problema. $|A| = m$, então, $C = \{A_i, 1 \leq i \leq m\}$.
- R é o conjunto de regras. Cada regra $R_i, 1 \leq i \leq k$, é composta pelo valor de alguma condição $C_i \in C$ combinado com alguma ação $A_i \in A$ e possui o formato $R_k = (c_{i,k}, a_{j,k})$, sendo que:
 - $c_{i,k}$ um valor que a condição C_i pode assumir na regra;
 - $a_{j,k}$ uma indicação que a ação A_j será executada na regra.

Cada regra R_k pode ser interpretada como estruturas condicionais do tipo “Se *condição* então *ações a serem executadas*”, na qual cada *condição* pode ser uma composta por operadores lógicos *and* e/ou *or*.

Uma observação importante é que cada *condição* C_i , em geral, assume valores binários *falso* (F) e *verdadeiro* (V) ou *Sim* (S) e *Não* (N) para uma determinada regra R_k . Isto indica que a condição C_i é verificada para a regra R_k . Quando uma determinada condição é indiferente para uma determinada regra, a mesma é deixada em *branco* (b). Quando uma determinada ação A_j deve ser executada devido a uma determinada regra R_k , a mesma deve ser sinalizada, em geral através de um X, para demonstrar que a ação em questão está sendo executada em razão da regra ser satisfeita. Caso uma ação não seja executada em função de uma determinada regra, utiliza-se *branco*(b) ou traço (-).

Quando uma Tabela de Decisão possui apenas valores de condições “S”, “N” ou brancos(b) e valores de ações iguais a “X ” ou branco, as mesmas são denominadas de Tabelas de Decisão com entradas limitadas. Na Tabela 1 é mostrado um exemplo de uma Tabela de Decisão com entrada limitada, na qual um firewall de uma rede deve tomar decisões de acordo com o tipo de pacote da rede.

Tabela 1: Exemplo de Tabela de Decisão com Entradas Limitadas.

Endereço IP de Origem Confiável	S	N	S	S	N	S
Endereço IP de Destino Confiável	S	S	N	N	N	S
Porta de Destino Confiável	S	S	N	S	S	N
Liberar Pacote	X			X		X
Encaminhar Pacote para Análise			X			

Entretanto, existem alguns casos nos quais as variáveis das Tabelas de Decisão apresentam valores que correspondem a intervalos, denominadas de entradas estendidas e entradas limitadas. Segundo Tchemra (2009) apud Gildersleeve (1970), Hugges, Shank e Stein (1968), este tipo de tabela é conhecida como Tabela de Decisão com Entrada Mista. A única restrição neste tipo de tabela é que cada linha só pode conter entradas de um tipo apenas. A Tabela 2, mostra um exemplo de uma Tabela de Decisão com Entradas Mistadas. Neste Exemplo, um usuário deseja comprar um produto precisa escolher entre diversos sites e lojas.

Entretanto, para facilitar a análise, as Tabelas de Decisão com Entradas Mistadas devem ser evitadas e transformadas em uma Tabela de decisão com Entradas Limitadas.

A Tabela 3 mostra a Tabela 2 transformada em uma Tabela de Decisão com Entradas Limitadas.

É importante ressaltar que em alguns casos, as ações a serem tomadas devem obedecer a alguma ordem. Neste caso, ao invés das ações a serem tomadas serem

Tabela 2: Exemplo de Tabela de Decisão com Entrada Mista.

Livro Disponível na Loja	S	N	N	S	N
Prazo de Entrega (dias)	≤ 10	≤ 10	≤ 10	≤ 10	> 10
Parcelas	≤ 3	≤ 3	≤ 3	≤ 3	4 - 12
Dinheiro Disponível	N	S	N	N	S
Comprar na Loja				X	
Comprar na Internet		X			X
Não Comprar	X		X		

Tabela 3: Exemplo de uma Tabela de Decisão com Entradas Limitadas.

Livro Disponível na Loja	S	N	N	S	N
Prazo de Entrega de até 10 dias	N	S	S	S	N
Entrega Superior a 10 dias	N	N	N	N	S
Pagamento Parcelado em Até 3 Vezes	S	S	N	S	N
Pagamento Parcelado de 4 a 12 Vezes	N	N	S	N	S
Dinheiro disponível para Pagamento	N	S	N	N	S
Comprar na Loja				X	
Comprar na Internet		X			X
Não Comprar		X		X	

marcadas com um “X”, as mesmas são numeradas indicando a ordem da execução das regras, sendo que a primeira ação é a de número 1. Como este trabalho apenas utiliza a idéia de Tabela de Decisão, não serão descritas todas suas características da mesma. Caso sejam necessárias mais informações, estas podem ser obtidas em Tchémra (2009).

Segundo Tchémra (2009), uma Tabela de Decisão convencional ou não-adaptativa (*TDN*) pode ser definida de maneira mais detalhada como a 6-upla:

$$TDN = (CT, NRT, CV, t_0, AT, CRA)$$

, na qual:

- CT é o conjunto de todas as possíveis configurações da tabela de decisão;
- $NRT \subseteq CT \times CV \times CT \times CRA$ é o conjunto de regras de decisão da tabela;
- CV é o conjunto finito de valores válidos das condições do problema inseridos na tabela, incluindo ε (vazio);
- $t_0 \in CT$ é a configuração inicial da tabela de decisão;
- $AT \in CT$ é o subconjunto de configurações da tabela de decisão aceitas durante o processo ($CT - AT$ são as configurações rejeitadas);

- CRA é o conjunto finito dos possíveis resultados obtidos pela execução das ações quando da aplicação das regras NRT , com $\varepsilon \in CRA$.

As regras $rt \in NRT$ são da forma $rt = (t_i, cv, t_j, r)$, na qual dado o valor de uma condição $cv \in CV$, a regra rt gera a resposta $r \in CRA$ e a tabela de decisão muda sua configuração de t_i para t_j . É importante observar que assim como descrito na Seção 3.2, no caso das RSSFs o conjunto de aceitação AT não é utilizado, pois o conjunto de estímulos de entrada em uma RSSF é infinita. Além disso, o símbolo ε também não é utilizado, pois não há mudanças na RSSF sem que haja estímulos válidos. Dessa forma, pode-se definir uma TDN que modela o comportamento de uma RSSF como uma 5-upla $TDN_{rssf} = (CT, NRT, CV, t_0, CRA)$.

Após esta breve descrição sobre Tabelas de Decisão, será descrito, a seguir, o funcionamento das Tabelas de Decisão Adaptativas, que foram definidas por Neto (2001), utilizando-a para simular um autômato adaptativo para reconhecer sentenças de linguagens dependentes do contexto.

A Tabela de Decisão Adaptativa (TDA) proposta por Neto (2001), utiliza como dispositivo subjacente uma Tabela de Decisão Convencional e uma camada adaptativa composta por linhas que correspondem às funções adaptativas. Dessa forma, as colunas da TDA correspondem às regras da mesma.

Assim, uma TDA que representa uma RSSF pode ser definida como $TDA = (TDN_{rssf}, CA)$, na qual TDN_{rssf} é a Tabela de Decisão Aplicada à RSSF e CA é o mecanismo adaptativo, que é definido pelo conjunto das funções adaptativas FA , cada uma declarada pela sêxtupla $FA = (nome, P, V, G, BA, AD, AA)$, na qual (TCHEMRA, 2009):

- *nome*: é uma identificação usada para referenciar a função;
- *P*: é composto pelo conjunto de parâmetros formais da função;
- *V*: é o conjunto de variáveis utilizadas para armazenar valores resultantes da aplicação da função adaptativa. Este parâmetro é opcional;
- *G*: é uma lista de geradores usados para referenciar novos valores, quando a função adaptativa é ativada Este parâmetro é opcional;
- *BA (Before Action)*: indica a chamada de ação adaptativa anterior (opcional);

- *AD*: representa o corpo da função adaptativa e é composta por um conjunto de ações adaptativas elementares de consulta (*AD?*), inclusão (*AD+*) e exclusão (*AD-*) que modificam o conjunto de regras corrente da tabela de decisão;
- *AA* (*After Action*): indica a chamada de ação adaptativa posterior (opcional).

É importante ressaltar que cada ação adaptativa executada em uma TDA corresponde a uma chamada de uma FA, através da atribuição de parâmetros formais.

Para melhor ilustrar uma TDA, a Figura 4 apresenta a estrutura geral de uma TDA.

	linha de cabeçalho (tag)	colunas das funções adaptativas	colunas das regras
tabela de decisão não-adaptativa	linhas das condições	declaração das funções adaptativas	valores das condições
	linhas das ações		ações a serem aplicadas
funções adaptativas	linhas das funções adaptativas		ações adaptativas a serem executadas

Figura 4: Estrutura Geral de uma TDA (TCHEMRA, 2009).

Cada um dos itens da Figura 4 que fazem parte da TDA são descritos a seguir (TCHEMRA, 2009; NETO, 2001):

- linha de cabeçalho (tag) – linha na qual são identificados os tipos de cada coluna que compõe a tabela.
 - H: indica a coluna do cabeçalho de especificação de uma função adaptativa; em cada célula desta coluna, junto às linhas das funções adaptativas, deve ser definido um dos caracteres:
 - * B (*before*): indica que a função deve ser executada antes da regra não-adaptativa subjacente;

- * A (*after*): indica que a função será executada depois da regra não-adaptativa subjacente;
 - * P: indica que a linha é um parâmetro de uma função adaptativa;
 - * V: indica que a linha corresponde a uma variável;
 - * G: indica um gerador.
- ? + –: representam, respectivamente, as colunas das ações adaptativas elementares de consulta, inclusão e exclusão de regras na tabela de decisão subjacente; em cada coluna desse tipo, as células podem conter:
- * nas células das condições, valores específicos que serão testados;
 - * nas células das ações, devem ser assinaladas as ações que serão tomadas;
 - * nas células das ações adaptativas, indicam-se aquelas ações adaptativas a serem executadas, bem como, seus parâmetros, suas variáveis e seus geradores correspondentes.
- S (*start*): indica a coluna da regra de início da execução da TDA; nela são representadas as condições iniciais e ações iniciais para a operação da tabela;
- R: indica que a coluna corresponde a uma regra normal, ou seja, é uma regra da tabela de decisão não-adaptativa, cujas células podem conter, além dos valores das condições e das indicações das ações subjacentes, que devem ser executadas, as sinalizações para as chamadas das ações adaptativas, com a passagem dos parâmetros, variáveis e geradores correspondentes; geralmente, as colunas de regras não-adaptativas são indicadas por números inteiros e seqüenciais;
- E: corresponde à última coluna da tabela, servindo como delimitador do conjunto de regras.
- linhas das condições: cada linha de condição corresponde à uma variável do problema;
 - valores das condições: em suas células, são indicados os valores das condições, podendo ser entradas limitadas (S , N ou branco (b)) ou entradas estendidas;
 - linhas das ações: cada linha representa uma ação ou procedimento, que pode ser executado em resposta ao conjunto de condições combinadas; nestas ações, pode-se incluir os comandos de chamada a tabelas que executam outras tabelas de decisão não-adaptativas ou adaptativas;

- ações a serem aplicadas: devem ser marcadas as células das ações que serão executadas quando as regras são avaliadas;
- linhas de funções adaptativas: nestas linhas são definidos os elementos das funções adaptativas (nomes das funções adaptativas, parâmetros, variáveis e geradores);
- ações adaptativas a serem executadas: em suas células assinalam-se aquelas que serão chamadas antes ou depois da aplicação da regra, assim como, indicam-se os valores dos parâmetros, variáveis e geradores necessários à sua execução.

3.4 Tabela de Decisão Adaptativa Estendida

A Tabela de Decisão Adaptativa Estendida (TDAE) é baseada na TDA apresentada na Seção 3.3, que é um dispositivo adaptativo dirigido por regras que utiliza uma Tabela de Decisão Convencional ou não adaptativa como dispositivo subjacente. A TDAE foi definida por Tchemra (2009) para ser utilizada em procedimentos de tomada de decisão que envolvem multicritérios simultaneamente. Para um melhor entendimento sobre o funcionamento de uma TDAE é necessário descrever sobre multicritérios. Dessa forma, esta seção descreve de maneira sucinta aspectos relacionados à Teoria da Decisão, aspectos relacionados a Decisões Multicritério e, por fim, a TDAE.

3.4.1 Decisões Multicritérios

Antes de falar de decisões multicritérios, é necessário fazer uma breve descrição sobre Teoria da Decisão de um modo geral, com a finalidade de possibilitar um melhor entendimento sobre o assunto. Primeiramente, segundo Tchemra (2009) apud Raiffa (1977), a Teoria da Decisão envolve processos de decisão que auxiliam pessoas a realizarem melhores escolhas, por meio de métodos de análise, utilizando informações coletadas e disponíveis. Existem inúmeras aplicações que envolvem tomada de decisão, tais como, planejamento estratégico de empresas, análise de riscos, investimentos, controle de tráfego e outros. Neste trabalho, o processo de tomada de decisão envolve a reconstrução da topologia de uma RSSF.

Uma decisão, não importando sua complexidade, envolve a escolha entre duas ou mais alternativas propostas, sendo que o número pode ser ilimitado nos casos mais complexos (CERTO, 2000). A alternativa escolhida tem como finalidade a

solução de um determinado problema. Segundo Almeida, Gomes e Gomes (2002), para resolver um determinado problema é necessário saber as causas que geraram o problema e as restrições envolvidas. No caso das RSSF, falhas de nós por falta de energia, destruição dos nós, desligamentos temporários são as principais causas dos problemas de controle de topologia. Entretanto, as restrições de energia dos nós sensores são mais importantes para que a reconstrução possa ser realizada da melhor maneira possível.

Em problemas relacionados a investimentos, planejamento estratégico e questões que envolvem negócios, em geral existe a figura do decisor, que é responsável pela tomada de decisão e pode ser representado por uma pessoa ou um grupo de pessoas (ALMEIDA; GOMES; GOMES, 2002). Entretanto, no comportamento de uma RSSF, devido à sua característica de auto-organização e a dificuldade de acesso em alguns casos, o processo decisório deve ser realizado de forma automática.

Segundo Tchemra (2009) apud Thomaz (2000), Pomerol e Adam (2007), o processo de tomada de decisão envolve a escolha de alternativas com base no valor de preferência daquele que toma as decisões. Dessa forma, não basta saber quais alternativas são possíveis, mas também escolher a que melhor se adapta ao problema em questão. Entretanto, nem sempre é possível conhecer todas as alternativas disponíveis e em casos de problemas mais complexos, uma decisão considerada ótima pode não ser alcançada. Isso ocorre muitas vezes porque nem sempre as pessoas apresentam conhecimento restrito em relação às decisões a serem tomadas. Portanto, para que a decisão possa ser considerada confiável é necessário que se leve em consideração tanto os elementos objetivos quanto os subjetivos.

Entretanto, é importante observar que a parte subjetiva, inerente ao decisor, não deve ser responsável pela escolha no processo de tomada de decisão, pois isto pode fazer com que alguma alternativa seja preferida em relação a outra (RAIFFA, 1977). Além disso, nem sempre uma boa decisão em um determinado momento pode ser considerada boa sempre, pois os fatores e as circunstâncias podem ser modificados, mudando totalmente a viabilidade de uma decisão tomada. Para minimizar a subjetividade e garantir que uma boa decisão seja tomada, a utilização de modelos e métodos de tomada de decisão é fundamental (RAIFFA, 1977; RAGSDALE, 2001).

De acordo com Tchemra (2009) apud Binder (1994), Pidd (1998), Balestrin (2002), a utilização de ferramentas computacionais na tomada de decisão é fundamental em todo o processo de tomada de decisão que vai desde a coleta de

informações sobre o problema, criação de modelos para análise e exploração das alternativas, escolha de uma alternativa, até o monitoramento da solução escolhida. O monitoramento é uma fase importante, pois permite a realimentação do processo e, conseqüentemente, um aprimoramento na tomada de decisão. Para que isso seja possível, estes recursos computacionais que geralmente consistem em um sistema de informação, devem manter bases de dados com informações sobre os problemas que estão envolvidos na tomada de decisão.

É importante ressaltar que o nível de tomada de decisão depende do tipo de informação envolvida, que pode ser (O'BRIEN, 2004):

- Decisão estruturada: situações levam a um processo de decisão já conhecido.
- Decisão semi-estruturada: neste nível de decisão parte do processo é conhecido, porém, não o suficiente para a tomada da decisão. Este nível é mais comum em situações que envolvem a tomada de decisão.
- Decisão desestruturada: neste nível, nenhum procedimento é especificado ou conhecido.

Do ponto de vista computacional, existem ferramentas capazes de trabalhar com todos os níveis de tomada de decisão. No caso de tomadas de decisão estruturadas, geralmente são utilizados Sistemas de Informação Gerencial (SIG) ou OLAP (*Online Application Processing*). Para tratar problemas semi-estruturados ou desestruturados, utilizam-se técnicas que permitem a resolução de problemas mais complexos, tais como Sistemas de Apoio à Decisão (SAD) ou técnicas de Inteligência Artificial (Redes Neurais, Mineração de dados, Algoritmos genéticos, e outros). Maiores detalhes sobre estes métodos podem ser encontrados em Tchemra (2009).

Após uma breve revisão sobre conceitos que envolvem Tomada de Decisão, serão descritos a seguir conceitos que envolvem Métodos Multicritério de Apoio à Decisão. Estes métodos permitem a compreensão e solução de problemas de decisão com número finito de critérios e alternativas, conferindo-lhes consistência e confiabilidade nos julgamentos do decisor (FULÖP, 2005). De acordo com a forma de formulação do problema métodos multicritério podem ser classificados em (ROY, 1996; FULÖP, 2005; TCHEMRA, 2009):

- Métodos Multicritério de Escolha: permite que apenas uma alternativa seja selecionada de um subconjunto obtido do conjunto de alternativas disponíveis;

- Métodos Multicritério de Classificação: estes métodos consistem em distribuir as alternativas em categorias, de acordo com um valor associado à alternativa, obtido por funções de utilidade;
- Métodos Multicritério de Ordenação: estes métodos ordenam as alternativas, de acordo com o nível de importância atribuído pelo decisor, baseado no problema.

Os métodos multicritério utilizam uma matriz de decisão contendo as alternativas, os critérios e seus respectivos pesos (TCHEMRA, 2009). Dessa forma, dado um problema com m critérios, n alternativas, o conjunto de critérios $C = \{c_1, c_2, \dots, c_m\}$ e o conjunto de alternativas $A = \{a_1, a_2, \dots, a_n\}$, a matriz de decisão pode ser representada de acordo com a Figura 5.

		ax_1	.	.	ax_j	.	.	ax_p
		a_1	.	.	a_j	.	.	a_p
w_1	c_1	$z_{1,1}$	.	.	$z_{1,j}$	.	.	$z_{1,p}$
.	.	.	.	.	.	.	.	.
w_i	c_i	$z_{i,1}$	.	.	$z_{i,j}$	.	.	$z_{i,p}$
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
w_m	c_m	$z_{m,1}$	.	.	$z_{m,j}$	.	.	$z_{m,p}$

Figura 5: Estrutura de uma Matriz de Decisão (FULÖP, 2005).

Além dos critérios e das alternativas, é necessária a atribuição de pesos aos critérios do problema, representados por um vetor $w = (w_1, w_2, \dots, w_m)^T$. Cada peso w_i do vetor w corresponde ao peso atribuído ao critério c_i , com $1 \leq i \leq m$, $c_i \in C$ e $w_i \in w$. A atribuição dos pesos pode variar de um método para outro, porém, em todos os métodos estes pesos devem estar relacionados à relevância dos critérios no problema. Após a atribuição dos pesos, é necessário determinar o desempenho de cada alternativa para resolução do problema. Assim, é construída uma matriz de pontuações $Z = (Z_{i,j})_{m \times n}^T$, na qual $Z_{i,j}$ representa o desempenho da alternativa a_j em relação ao critério c_i , na qual $1 \leq j \leq n$ e $1 \leq i \leq m$ (FULÖP, 2005).

Por meio da matriz de pontuações Z e do vetor w , é possível que cada método associe valores globais para cada uma das alternativas $a_j \in A$ para a tomada

de decisão. Estes valores são determinados pelos métodos multicritério e formam um vetor $ax = (ax_1, ax_2, \dots, ax_n)^T$, no qual ax_j é associado à alternativa a_j (TCHEMRA, 2009).

Independente da sua classificação, os métodos multicritério são provenientes da Escola Americana ou da Escola Européia, que se baseiam na Teoria da Utilidade Multiatributo e na Sobreclassificação, respectivamente (TCHEMRA, 2009).

Os métodos baseados na Teoria de Utilidade Multiatributo são conhecidos como métodos multicritério MUAT (*Multiple Attribute Utility Theory*). Segundo esta teoria, pode-se aplicar funções de utilidade ao modelo matemático do problema de decisão, pelas quais se refletem a utilidade e preferências do decisor para cada critério em relação às alternativas possíveis (TCHEMRA, 2009). Dentre os métodos MUAT, podem ser destacados o AHP (*Analytic Hierarchy Process*), SMART (*Simple Multi-Attribute Rating Technique*) e TODIM (Tomada de Decisão Interativa Multicritério).

Para determinar o desempenho ax_j de cada alternativa a_j , estes os métodos MUAT levam em consideração funções de utilidade para atribuir pesos w_i aos critérios c_i e obter às pontuações $Z_{i,j}$. De acordo com Tchemra (2009) apud Keeney e Raiffa (1976), as alternativas com valor de utilidade mais elevados, em geral, resultam em uma alternativa com melhor desempenho, ou seja, estes métodos em geral indicam qual a melhor alternativa.

Os métodos baseados em Sobreclassificação (*Outranking*) são também conhecidos como métodos MCDA (*Multiple Criteria Decision Aiding*) e consistem no estabelecimento de relações entre pares de alternativas por meio de comparações e têm como objetivo ou classificar as alternativas em categorias, ou ordenar as alternativas, ou obter a melhor alternativa, segundo os níveis de preferência do decisor. Dentre os métodos de Sobreclassificação podem ser citados ELECTRE (*Elimination and Choice Translating Reality*), PROMETHEE (*Preference Ranking Method for Enrichment Evaluation*), TOPSIS (*Technique for Order Preference by Similarity to Ideal Solution*) e MACBETH (*Measuring Attractiveness by a Categorical Based Evaluation Technique*).

Nos métodos MCDA as alternativas a_j são categorizadas utilizando-se os valores w_i e $z_{i,j}$ são usados para determinar a categorização de cada alternativa a_j , que consiste em determinar a preferência de uma alternativa a_i em relação à alternativa a_j , ou o nível de concordância, ou discordância, de uma em relação à outra (TCHEMRA, 2009). Dessa forma, estes métodos determinam um subconjunto de alternativas viáveis, que podem ser utilizadas pelo decisor para reduzir

o tamanho do conjunto inicial A ou para tomar alguma decisão.

Uma observação importante é que as preferências de escolha que envolvem duas alternativas a_i e a_j são do tipo binária. Mais detalhes sobre preferências de escolha podem ser encontrados em Lewis e Papadimitriou (1998), Araya, Carignano e Gomes (2004), Tchemra (2009). Como este trabalho utiliza uma Tabela de Decisão Adaptativa Estendida baseada na proposta apresentada em Tchemra (2009) na qual o método AHP é utilizado como Método de Decisão Multicritério, uma maior ênfase será dada apenas a este método.

Método AHP (*Analytic Hierarchy Process*)

O método AHP foi desenvolvido por Thomas L. Saaty e, conforme dito anteriormente, é um método multicritério do tipo MUAT. Este método consiste na decomposição hierárquica do problema é dividido em três fases principais (TCHEMRA, 2009):

- Estruturação do problema (Primeira Fase): esta fase do método consiste na representação das partes em níveis hierárquicos. No nível mais alto da estrutura, é representado o objetivo da decisão, seguido pelos níveis de critérios e subcritérios, caso existam, e finalizando com o nível das alternativas, mostrando as relações entre os elementos.
- Julgamentos comparativos (Segunda Fase): nesta fase, dentro de cada nível, os critérios são julgados aos pares, de forma que o decisor deve definir uma escala de valores e julgar a importância de um critério c_i em relação a um critério c_j . Com base nestes critérios, é montada uma matriz E de julgamentos, que representa a intensidade de preferência de um decisor por um critério em relação a outro. Dessa forma, $E_{i,j}$ representa a intensidade de preferência do critério c_i em relação ao critério c_j . Além disso, tem-se que $E_{j,i} = \frac{1}{E_{i,j}}$ é o valor de comparação entre os critérios c_j e c_i e $E_{i,i} = 1$, com $1 \leq i, j \leq m$ e m é o número de critérios existentes (SAATI, 1991; SAATI, 1994).

Segundo Tchemra (2009), como em geral os critérios são descritos de forma verbal Saati (1991) propôs uma escala, denominada de Escala Fundamental de Saati que possibilita traduzir em valores quantitativos as preferências do decisor, conforme apresentado na Figura 6. A partir da matriz E , é gerado o vetor de pesos dos critérios w .

Em seguida, é necessário verificar se os julgamentos são consistentes e aceitáveis que, de acordo com Saati (1994), é baseado na razão de con-

sistência CR . Se $CR \leq 0,10$ o valor de julgamento é aceitável, caso contrário, o decisor deve rever seus julgamentos. Em Tchemra (2009), são apresentados os detalhes do cálculo de CR .

Caso CR seja aceitável, é necessário determinar a matriz de desempenho das alternativas em relação aos critérios. Esta matriz, denominada de Z é calculada analogamente ao julgamentos dos critérios, de maneira que as preferências de cada alternativa $a_i \in A$ em relação a cada critério deve ser $c_j \in C$, com $1 \leq i \leq n$ e $1 \leq j \leq m$ e n e m são a quantidade de alternativas e de critérios, respectivamente. Assim, $z_{i,j}$ é o nível de preferência da alternativa a_i em relação ao critério a_j . Por fim, é possível calcular o valor global de cada alternativa ax_j da seguinte maneira:

$$ax = Z.w \quad (3.2)$$

- Análise das prioridades (Terceira Fase): esta fase consiste em verificar a alternativa com maior prioridade e recomendá-la como melhor escolha.

Escala	Significado
1	Importância ou preferência igual: os critérios são igualmente importantes
3	Importância ou preferência moderada por um em relação ao outro: um critério é um pouco mais importante que o outro
5	Importância ou preferência forte ou essencial: o julgamento favorece fortemente um critério em relação a outro
7	Importância ou preferência muito forte ou demonstrada: um critério é muito fortemente importante em relação a outro
9	Extrema importância ou preferência: um critério é extremamente preferível a outro
2,4,6,8	Valores usados em julgamentos intermediários

Figura 6: Escala Fundamental de Saati (SAATI, 1991).

3.4.2 Descrição da Tabela de Decisão Adaptativa Estendida

A Tabela de Decisão Adaptativa Estendida (TDAE) foi proposta por Tchemra (2009) baseada na formulação da TDA apresentada na Seção 3.3 e tem por objetivo apoiar processos decisórios multicritérios semi-estruturados, possibilitando ao decisor a análise do conjunto de regras, interação com o sistema, resultando na obtenção de soluções viáveis para o problema. Além disso, a TDAE possui um conjunto de técnicas projetadas com o objetivo de suportar processos decisórios, tais como (TCHEMRA, 2009):

- permitir a modelagem de problemas de decisão semi-estruturados;
- permitir uma melhor compreensão do problema;
- proporcionar ao decisor condições para expressar seus julgamentos a cerca das informações do problema de decisão;
- permitir analisar os critérios impostos às regras do problema e suas combinações, além de apresentar alternativas de solução para situações não previstas anteriormente;
- incorporar algoritmos clássicos de métodos multicritério;
- usar técnicas adaptativas para que o sistema se automodifique, tanto na sua estrutura, como no seu comportamento, de forma autônoma, e gere respostas para o processo de tomada de decisão.

Formalmente, a TDAE proposta por Tchemra (2009) pode ser definida como uma tripla $TDAE = (TDA, FM, M)$, na qual:

- TDA : é a Tabela de Decisão Adaptativa convencional descrita na Seção 3.3;
- FM : é um conjunto de funções auxiliares que estabelece uma comunicação entre a camada adaptativa e o restante da tabela. Estas funções devem ser executadas sempre antes da aplicação da regra que a referencia, pois a TDAE depende dos valores calculados por elas.
- M : é o método multicritério utilizado. Apesar do método AHP (descrito na Seção 3.4.1) ter sido escolhido, a especificação de M permite que outros métodos de decisão multicritérios possam ser aplicados, de acordo com a preferência do decisor.

Para um melhor entendimento deste formalismo, a estrutura geral da TDAE é apresentada na Figura 7.

A implementação das funções do conjunto FM é definida externamente e sua operação não é explicitada na tabela. Por outro lado, a operação das chamadas dessas funções ocorre de maneira similar ao de uma chamada de função adaptativa, exceto pelo fato de que as funções em FM devem ser executadas em primeiro lugar.

			AÇÕES ADAPTATIVAS				REGRAS					
			...	$\mathcal{AD}_i$	...		r_1	...	r_j	...	r_s	E
TABELA DE DECISÃO CONVENCIONAL	CRITÉRIOS	c_1					Valores dos critérios d_{ij}					
		c_2										
		$\vdots$										
		c_i										
		$\vdots$										
		c_m										
	ALTERNATIVAS	a_1					Ações a serem aplicadas x_{kj}					
		a_2										
		$\vdots$										
		a_k										
		$\vdots$										
		a_p										
	FUNÇÕES	FM ₁					Funções auxiliares a serem chamadas					
		...										
		FM _q										
CAMADA ADAPTATIVA	FUNÇÕES ADAPTATIVAS	$\mathcal{FAD}_1$					Ações adaptativas a serem executadas					
		$\mathcal{FAD}_2$										
		$\vdots$					Referências a $\mathcal{FAD}_s$					
		$\mathcal{FAD}_s$										
		$\mathcal{FAD}_{sf}$										

conjunto de ações adaptativas elementares que compõem $\mathcal{FAD}_s$

Figura 7: Estrutura Geral da TDAE (TCHEMRA, 2009).

A TDA segue o mesmo formalismo definido por Neto (2001) e apresentado na Seção 3.3, na qual $TDA = (TDN, CA)$. Assim, dado um conjunto de critérios $C = \{c_1, c_2, \dots, c_m\}$ e o conjunto de alternativas do problema $A = \{a_1, a_2, \dots, a_n\}$, ou seja, A corresponde ao conjunto finito dos possíveis resultados das ações após a aplicação das regras NRT . Assim, pode-se definir $TDN = (CT, NRT, CV, t_0, AT, A)$, na qual (TCHEMRA, 2009):

- CT é o conjunto de todas as configurações possíveis da tabela de decisão. Uma configuração significa a imagem da tabela em cada instante, mostrando todos os critérios, as alternativas, as regras e as combinações válidas de condições contidas no conjunto de regras;
- NRT é o conjunto finito de regras de decisão. Cada regra $r_j \in NRT$, $1 \leq j \leq q$ é do tipo $r_j = (d_{i,j}, x_{k,j})$, na qual:
 - $d_{i,j}$: representa um valor binário para o critério $c_i \in C$ na regra r_j ;
 - $x_{k,j}$: é um valor binário que sinaliza a alternativa $a_k \in A$ para a regra r_j ;
- CV é o conjunto finito dos valores $d_{i,j}$ válidos dos critérios c_i ;

- $t_0 \in CT$ é a configuração inicial da tabela de decisão;
- $AT \subseteq CT$ é o subconjunto de configurações aceitas da tabela;
- A é o conjunto finito das alternativas do problema;

A camada adaptativa CA é associada ao conjunto de regras NRT e é definido como $CA = (FA, RA)$, na qual:

- FA : é o conjunto de funções adaptativas FAD_s , $1 \leq s \leq nf$, na qual cada função adaptativa FAD quando instanciada e chamada com os respectivos argumentos, executa as ações adaptativas que a compõe;
- RA : é o conjunto de regras composto pelas regras da tabela de decisão subjacente e pelas chamadas adaptativas.

A estrutura geral de cada função adaptativa $FAD \in FA$ é representada por uma óctupla $FAD = (NF, P, V, G, BA, AD, AA)$, que representam:

- NF : nome ou identificação da função;
- P : conjunto de parâmetros, no qual $P = \{p_i, 1 \leq i \leq np\}$, $|P| = np$;
- V : conjunto de variáveis, no qual $V = \{v_i, 1 \leq i \leq nv\}$, $|V| = nv$;
- G : conjunto de geradores, no qual $G = \{g_i, 1 \leq i \leq ng\}$, $|G| = ng$;
- BA : é uma indicação de ação adaptativa anterior (opcional); $BA \in AD$;
- AD : representa o corpo da função adaptativa e é composto por um conjunto de ações adaptativas elementares de consulta (?), inclusão(+) e exclusão(-) que modificam o conjunto corrente de regras da tabela de decisão subjacente. Dessa forma, AD pode ser representado na forma $AD = \{AC, AE, AI\}$;
- AA : é uma indicação de ação adaptativa posterior (opcional); $AA \in AD$

O conjunto de regras RA da camada adaptativa CA é composto por regras $ra_j = (BA, r_j, AA)$, $1 \leq j \leq r_j$ e $AA, BA \in AD$. Estas regras são provenientes da associação de ações adaptativas AD com as regras da tabela subjacente, na qual:

- BA : indica uma ação adaptativa a ser executada antes da aplicação da regra r_j (opcional);

- r_j : regra da tabela de decisão subjacente associada à ação adaptativa;
- BA : indica uma ação adaptativa a ser executada depois da aplicação da regra r_j (opcional).

É importante observar que se $r_j = ra_j$, ou seja, BA e AA não estão presentes em ra_j , significa que nenhuma ação adaptativa está associada a r_j e que até duas ações adaptativas podem ser associadas a cada regra r_j (uma BA e uma AA).

Segundo Tchémra (2009), o formalismo da TDAE possibilita que para cada função adaptativa FAD seja associada uma lista de ações adaptativas elementares AD , que indicam as modificações a serem impostas ao conjunto de regras, de acordo com o método M adotado para o processo decisório de problemas de decisão específicos. Além disso, a aplicação da TDAE pode ser vista sob dois aspectos:

- a TDAE faz o papel de decisor, no qual a solução do problema de decisão é apresentada pelo próprio dispositivo, ao final do processo decisório;
- a TDAE pode ser utilizada como um instrumento para consulta de decisões parciais, deixando que a decisão final seja tomada pelo decisor ou pelo especialista do particular problema de decisão; nesse sentido, o algoritmo pode apresentar um conjunto de possíveis soluções viáveis para o problema, segundo o método multicritério adotado.

No caso da problema de reconstrução da topologia em RSSF, como em geral não há a possibilidade da existência de uma pessoa atuar como decisor, além da necessidade de auto-configuração citada no Capítulo 2, a TDAE deve ser utilizada como um decisor e determinar qual a melhor configuração.

Para uma melhor compreensão do funcionamento da TDAE, a seguir será descrito o funcionamento da mesma.

No início de sua operação, a TDAE apresenta uma configuração inicial t_0 e funciona como uma tabela de decisão convencional, ou seja, executa a busca das regras procuradas pelo processo decisório, de acordo com o método multicritério escolhido. À medida que situações não previstas surgem, a Camada Adaptativa da TDAE atua, gerando uma nova configuração da tabela, através da inclusão de informações na tabela. Nesta etapa, os critérios e alternativas, são submetidos ao método multicritério adotado para julgamento, que gera um resultado ou categoriza as regras iniciais (TCHEMRA, 2009).

Deve-se observar que os pesos dos critérios e as preferências das alternativas também são usados pelas funções auxiliares de FM para determinar situações ainda não consideradas na tabela de decisão adaptativa.

Nas operações de consulta das regras da tabela de decisão adaptativa TDA , pode ocorrer de existirem ou não regras que satisfaçam os critérios especificados. Caso haja ocorrência do primeiro caso, a regra em questão determinará qual alternativa deverá ser utilizada. Caso contrário, uma função adaptativa é associada à regra especificada, e ações adaptativas são executadas, modificando a tabela de decisão adaptativa (TCHEMRA, 2009).

O algoritmo adotado para a tomada de decisão TDAE foi dividido em três módulos principais:

- Construção da Tabela de Decisão Convencional (TDN): esta etapa consiste basicamente na configuração dos critérios e alternativas.
- Criação da Matriz de Decisão: esta etapa cria a matriz de decisão baseada no julgamento do decisor. Os valores da matriz representam os pesos e preferências dos critérios e alternativas.
- Montagem da Camada Adaptativa e uso da TDAE para consultas: nesta etapa são declaradas as funções adaptativas, definição dos corpos das funções adaptativas e definições das ações adaptativas elementares das funções adaptativas.

3.5 Considerações sobre o Capítulo

Neste capítulo foram descritos inicialmente, a formulação geral de um dispositivo adaptativo e, em seguida, alguns dispositivos adaptativos. Inicialmente, foi descrito o funcionamento de um grafo dinâmico e definida formalmente sua estrutura geral, baseada na formulação geral de um dispositivo adaptativo. Este dispositivo foi descrito, pois mostra como pode ser utilizado para ilustrar a topologia na RSSF.

Em seguida, foram descritas a Tabela de Decisão Adaptativa (TDA) e a Tabela de Decisão Adaptativa Estendida (TDAE). Para que fosse possível descrever a TDAE, foi necessário anteriormente discursar sobre Teoria da Decisão e Multicritérios. A TDA será utilizada neste trabalho como decisor para executar mudanças na topologia da RSSF, sempre que necessário. A TDAE é uma outro

alternativa, porém mais custosa para tomada de decisão, que envolve múltiplos critérios.

4 Algoritmo para Manutenção de Topologia Utilizando Tabela de Decisão Adaptativa

A manutenção de topologia é um problema bastante estudado e importante para economia de energia, garantia de tolerância a falhas e reconstrução da topologia de uma rede quando ocorrem falhas em um ou mais nós, conforme apresentado no Capítulo 2. Entretanto, diversos algoritmos consideram apenas situações pré-determinadas, o que faz com que falhas não previstas por estes algoritmos provoquem falhas de comunicação na rede. Dessa forma, este capítulo apresenta o funcionamento da TDA e Grafos Dinâmicos aplicados ao problema de Manutenção de Topologias em RSSFs.

Este capítulo está organizado da seguinte maneira: a Seção 4.1 apresenta o problema e algumas definições. Nas Seções 4.2 apresenta a descrição do algoritmo, seu funcionamento e um exemplo de execução do mesmo. As Seções 4.3 e 4.5 descrevem brevemente alguns ambientes de simulação e aspectos de implementação e as considerações finais do capítulo, respectivamente.

4.1 Definição do Problema de Manutenção de Topologia

Este trabalho envolve a aplicação de uma TDA (apresentada no Capítulo 3, Seção 3.3) em Manutenção de Topologia em RSSFs. A idéia de utilizar uma TDA nesse contexto reside no fato de que critérios não previstos podem levar à reconstrução da topologia em RSSFs e incorporá-los ao algoritmo, podendo torná-lo mais eficiente do ponto de vista da conectividade. É importante ressaltar que não foram considerados casos de não-determinismo, ou seja, não existirão regras diferentes que atendam ao mesmo critério simultaneamente.

Diversas topologias podem ser utilizadas em uma RSSF; entretanto, neste trabalho será utilizada uma topologia em árvore, por ser simples e eficiente. Para representar a árvore de roteamento, será utilizado um grafo direcionado dinâmico. Um grafo dinâmico, apresentado no Capítulo 3, Seção 3.2, será utilizado para representar graficamente a topologia da RSSF.

4.2 Aspectos Gerais do Algoritmo Proposto

O algoritmo para Manutenção de Topologia em Redes de Sensores Sem Fio Utilizando Tabela de Decisão Adaptativa Estendida proposto tem como base os estudos apresentados em Gonda et al. (2009) e Gonda, Cugnasca e Neto (2010).

O algoritmo proposto é dividido em duas fases: construção da topologia inicial e reconstrução da topologia através de ações adaptativas, descritas nas Seções 4.2.1 e 4.2.2, respectivamente. É importante observar que a Adaptatividade será usada somente na fase de reconstrução de topologia, na qual são necessários critérios para escolha de um novo nó pai.

4.2.1 Construção da Topologia Inicial

Esta fase do algoritmo consiste em determinar uma configuração na qual as informações coletadas pelos nós fonte sejam encaminhadas para o nó sorvedouro. Esta fase não envolve problemas de decisão complexos, e portanto não é necessária a utilização de mecanismos adaptativos.

Para a construção da topologia inicial existem diversas alternativas (ZHOU; KRISHNAMACHARI, 2003), porém, neste algoritmo será adotada uma abordagem baseada no algoritmo *Earliest First Tree* (EF-Tree), pois em geral as mensagens de *início* são recebidas primeiramente dos nós mais próximos.

Dessa forma, como o objetivo é fazer com que as informações possam ser enviadas para o sorvedouro, este nó é responsável por enviar uma mensagem do tipo *início* na RSSF. Esta mensagem é responsável por detectar os nós existentes, bem como, determinar quais nós participarão da RSSF inicialmente. Ao receber uma mensagem do tipo *início*, cada nó escolhe o nó remetente como pai (cria-se uma aresta entre os dois nós) e encaminha esta mensagem através de *broadcast*. É importante observar que um nó pode receber a mensagem de *início* de mais de um nó. Neste caso, duas abordagens podem ser utilizadas: nenhuma ação é realizada ou os remetentes das mensagens que chegarem após a escolha do nó pai serem

adicionados em uma lista de possíveis pais para ser utilizada posteriormente. É importante observar que em ambos os casos nenhuma mensagem é propagada pelo nó para evitar tráfegos desnecessários na rede.

Após esta fase, a configuração c_0 do grafo G é obtida e os vértices e arestas selecionadas são utilizados na comunicação. É importante observar que não há problema algum de não existirem mecanismos adaptativos nesta fase, pois não há necessidade de tomada de decisão.

Um exemplo do resultado da execução desta etapa do algoritmo, sem a lista de possíveis pais, pode ser visualizado na Figura 8.

Figura 8: Exemplo de Construção da Topologia Inicial.

Conforme descrito anteriormente, uma segunda abordagem pode utilizar as várias mensagens do tipo *início* recebidas pelo mesmo nó para montar uma lista adicional de possíveis pais deste nó. Neste caso, se após ter selecionado um nó pai, se um nó receber mais de uma mensagem do tipo *início*, o mesmo deve adicionar o nó remetente na lista de possíveis pais. Graficamente, um possível pai é representado por uma aresta pontilhada, conforme apresentado na Figura 9.

Figura 9: Exemplo de Construção da Topologia Inicial Utilizando Lista de Possíveis Pais.

Nesta abordagem, sempre que um nó decide escolher um novo pai, ele seleciona o mesmo da lista de possíveis pais, que pode ser modificado durante o

funcionamento da rede.

Contudo, podem ocorrer falhas de nós pais e de possíveis pais. Dessa forma, para garantir que o nó pai está ativo, mensagens periódicas para o nó pai podem ser enviadas ou o próprio nó pai pode enviar mensagens de alerta quando a sua energia estiver se esgotando. Mais ainda, uma combinação dos dois pode ser utilizada.

4.2.2 Reconstrução da Topologia

A etapa de reconstrução de topologia é responsável por determinar mudanças que possam ocorrer na topologia devido a algum evento que possa ter ocorrido na rede. Conforme ressaltado no Capítulo 2, a reconstrução pode ser realizada de maneira local ou de maneira global. Quando a reconstrução é de maneira local, apenas parte dos nós envolvidos é utilizada, o que pode fazer com que a topologia escolhida não seja a mais eficiente possível. Entretanto, esta abordagem envolve menos custo de comunicação. Quando a reconstrução é global, a topologia é construída levando-se em consideração todos os nós da rede, porém, o custo de comunicação e processamento é maior neste caso.

Para a reconstrução da topologia através do uso de TDA é necessário definir inicialmente os critérios e regras. O objetivo principal neste caso é reconstrução da topologia da rede por meio da obtenção de uma rota até o nó sorvedouro. Em uma Rede de Sensores Sem Fio os fatores que podem influenciar na comunicação e na topologia são eventos que podem vir a ocorrer na rede. Dessa forma, os critérios de escolha neste trabalho serão os eventos que podem influenciar na configuração da topologia da rede.

As alternativas a serem adotadas são a reconstrução local ou global da topologia, que vão depender dos eventos que ocorreram na rede. É importante ressaltar que a reconstrução global sempre será realizada a partir do nó sorvedouro da rede. Outra observação importante é que nem sempre que ocorrer falhas em um nó da rede será necessária a reconstrução da mesma. Por exemplo, se um nó folha na árvore (nó do tipo fonte) falhar não há a necessidade de reconstrução, já que a comunicação da rede não seria prejudicada por esta falha. Uma outra alternativa seria não realizar nenhuma operação de reconstrução; contudo neste trabalho só serão levadas em consideração as regras que envolvam reconstrução da topologia (tanto local como global).

A primeira etapa do algoritmo é a configuração da TDA para seu funciona-

mento, que será descrita na Seção 4.2.2.1.

4.2.2.1 Configuração da Tabela de Decisão Adaptativa

A primeira etapa da configuração é o estabelecimento de critérios, alternativas e regras da TDA. Os critérios a serem levados em consideração neste trabalho (não necessariamente em ordem de importância) são:

- tipo de nó (c1): neste critério o nó pode ser um nó fonte ou um nó de roteamento. O valor “S” indica que o nó é de roteamento e o valor “N” indica que o nó do tipo fonte. Este parâmetro é importante, pois se algum problema ocorre com um nó fonte não há a obrigatoriedade de se reconstruir a topologia. Entretanto, se o problema ocorrer com um nó de roteamento, é necessária, no mínimo, uma reconstrução local da topologia da rede.
- Falha de um nó (c2): este critério também apresenta dois valores possíveis e sempre que ele for levado em consideração (valor igual a “S”), indica que falha em algum nó da rede. É importante observar que neste algoritmo não foram incorporados mecanismos de detecção de falhas.
- Inserção de um novo nó (c3): sempre que uma regra encontrar este critério, é necessário que uma reconstrução da topologia deva ser executada.
- Exclusão de um nó da rede por desligamento temporário(c4): sempre que este evento acontecer (valor “S”), pode haver uma reconstrução da topologia da rede.
- Nó com baixa energia (c5): este evento pode indicar que existe algum nó com pouca energia (valor “S”).

Após o estabelecimento dos critérios e alternativas é necessário criar as regras iniciais da TDA. É importante ressaltar que as regras e critérios dependem do usuário e que várias configurações são possíveis. Por esta razão é necessária a camada adaptativa, pois nem sempre todas as possibilidades podem ser configuradas ou são configuradas pelo usuário. É importante observar que, ao contrário do que acontece na TDA proposta por Neto (2001) e também descrita por Tchemra (2009) que permite a modificação apenas das regras, é interessante que o algoritmo para Manutenção da Topologia possa também permitir a criação de novos critérios, já que eventos não previstos podem ocorrer. Após este passo, o dispositivo subjacente, representado pela Tabela de Decisão Não Adaptativa (TDN) é montado.

Uma possível configuração da TDN para os critérios e alternativas apresentados é mostrada na Figura 10.

	Regras		
	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_3$
c1	S	N	N
c2	N	S	S
c3	S	N	N
c4	S	N	N
c5	N	S	N
a1	X		
a2		X	X

Figura 10: Um exemplo de configuração de regras de uma TDA.

Na Figura 10 existem 3 regras previamente configuradas e que ainda podem sofrer modificações (pela TDA). Dados nc , nr e na o número de critérios, número de regras e número de alternativas da TDN, respectivamente, as seguintes afirmações devem ser consideradas:

- Se o critério c_j influencia na regra r_k , então na célula (c_j, r_k) o valor deve ser “S”. Caso contrário, o valor da célula (c_j, r_k) é “N”, $1 \leq j \leq nc$ e $1 \leq k \leq nr$.
- Se a_i for a alternativa para a regra r_k , então a célula (a_i, r_k) deverá ser marcada com um X, $1 \leq i \leq na$, $1 \leq k \leq nr$

Para que as decisões possam ser tomadas pela TDA, o usuário deve informar os pesos entre critérios antes da execução do algoritmo, pois conforme dito anteriormente, em geral não há a possibilidade de interação entre o usuário e a RSSF.

Após esta fase, é necessário especificar a Camada Adaptativa, que é responsável por consulta, inserção e remoção de novas regras, critérios e alternativas. Uma nova regra é inserida somente se a regra que está sendo buscada não existe, ou seja, todas as regras foram consultadas e nenhuma possui os mesmos valores dos critérios de nenhuma regra da tabela e o final das regras foi encontrada. Nesse ponto, uma função auxiliar FM que determina os valores dos critérios na regra e a alternativa associada à regra criada é chamada. Além disso, FM faz chamada a funções adaptativas de inclusão e exclusão de regras.

4.2.3 Funcionamento do Algoritmo

Na Seção 4.2.2.1, foi descrito o funcionamento da TDA que é parte do algoritmo de Manutenção de Topologia. No algoritmo, serão considerados os seguintes tipos de mensagem de controle:

- *início*: indica o início da RSSF, conforme descrito na Seção 4.2.1.
- *alerta*: esta mensagem indica que algum evento que pode modificar a topologia da rede ocorreu. Nesta mensagem deve ser especificado o código do evento e a identificação do nó.
- *local*: indica que uma reconstrução local deve ser executada a partir de um determinado nó (identificado na mensagem).
- *global*: indica que uma reconstrução global deve ser executada a partir do nó sorvedouro.
- *busca*: esta mensagem indica que um nó está buscando um novo pai na rede.

Após os conceitos iniciais apresentados, segue o detalhamento dos passos de um algoritmo para Manutenção de Topologia de Redes de Sensores Sem Fio:

Algoritmo: ADAP_RSSF

Passo 1. Nó sorvedouro envia uma mensagem do tipo *início* na RSSF.

Passo 2. Se nó é o nó sorvedouro então aguarda chegada de mensagens de controle.

Passo 2.1 Se mensagem é do tipo *alerta* então

Passo 2.1.1 Chama a TDA.

Passo 2.1.2 Se os critérios (eventos) procurados existem na regra r_k então

Passo 2.1.2.1 Executa a alternativa determinada na regra r_k .

Passo 2.1.3 Senão Se todos os critérios existem, mas a regra correspondente não, então

Passo 2.1.3.1 Adiciona a regra à TDA por meio da camada adaptativa.

Passo 2.1.4 Senão Se o critério não existir então

Passo 2.1.5 Adiciona o critério à tabela.

Passo 3 Se o nó não é sorvedouro então

Passo 3.1 Se evento ocorreu então

Passo 3.1.3 Envia mensagem de *alerta* pela rede para que possa reconstruir a TDA;

Fim do Algoritmo

Após uma descrição um pouco mais detalhada do algoritmo, na seção seguinte é apresentado um exemplo de execução do algoritmo.

4.2.3.1 Exemplo de Execução

Para ilustrar o funcionamento do algoritmo proposto deve-se considerar inicialmente os critérios e alternativas apresentados na Seção 4.2.2.1. O primeiro passo é a construção da topologia inicial. Considerou-se inicialmente que os nós sensores encontram-se dispostos como na Figura 11(a). Dessa forma, o nó sorvedouro envia uma mensagem do tipo *início* para construir a topologia inicial. Conforme descrito na Seção 4.2.1, cada nó sensor escolhe como nó pai o remetente da primeira mensagem do tipo *início* recebida. Além disso, se um nó receber uma mensagem do tipo *início* de mais de um nó, os remetentes das mensagens que chegarem após a escolha do nó pai são adicionados na lista de possíveis pais. É importante observar que o primeiro nó desta lista é o pai atual. A Figura 11(b) mostra a configuração da topologia da rede após a configuração inicial. Esta topologia consiste na configuração c_0 do grafo dinâmico que representa a RSSF.

Figura 11: Configuração c_0 do grafo dinâmico.

Uma possível configuração da TDN para os critérios e alternativas apresentados é mostrada na Figura 12.

		Regras				
		1	2	3	4	5
c1		S	N	N	S	S
c2		N	S	S	S	S
c3		S	N	N	N	S
c4		S	N	N	N	S
c5		N	S	N	N	N
a1		X			X	
a2			X	X		X

Figura 12: Configuração t_0 da TDA.

Após a configuração inicial, é necessário construir a TDA para que seja possível a manutenção da topologia através da reconstrução sempre que necessário. Como os critérios e alternativas já foram definidos, é necessário agora a definição das regras iniciais.

Será suposto agora, que ocorreu uma falha com o nó v_4 . Ao detectar esta falha, um vizinho, por exemplo, o nó v_5 envia uma mensagem do tipo *alerta* com a identificação do nó que falhou e indicando que o mesmo é um nó roteador. Assim, a regra a ser buscada na Tabela é apresentada na Figura 13.

0
S
S
N
N
N
X

Figura 13: Regra existente a ser buscada na TDA.

Como a regra procurada corresponde a uma regra da TDA (regra r_4), então a alternativa correspondente à regra, neste caso, uma reconstrução local da topologia é realizada. Assim, o nó sorvedouro envia uma mensagem do tipo *local* referenciando o nó v_4 . Em seguida, todos os nós que estavam conectados ao nó

v_4 devem escolher um novo pai da sua lista de possíveis pais. Caso não haja ninguém nesta lista de um nó, é necessário enviar uma mensagem do tipo *busca* para a rede para selecionar um novo pai. A Figura 14 mostra a configuração c_1 após a reconstrução local da topologia da rede e, conseqüentemente, adicionar um ou mais nós à lista de possíveis pais.

Figura 14: Configuração c_1 do grafo dinâmico que representa a RSSF.

Será suposto agora que ocorreram as falhas dos nós v_2 e v_7 e que um novo nó foi inserido na rede, conforme mostrado na Figura 15.

Figura 15: Falha nos nós v_2 e v_7 e inserção do nó v_{10} .

Assim, ao detectar tais eventos, os nós vizinhos geram mensagens do tipo *alerta* (uma para cada evento da rede). Ao receber as mensagens, o nó sorvedouro busca a regra mostrada na Figura 16 na TDA.

Como a regra não foi encontrada, uma função adaptativa do tipo *Before* deve ser executada para criação da nova regra na TDA, que deve ser executada em seguida. A configuração t_1 da TDA é apresentada na Figura 17.

É importante observar que apesar de não detalhado aqui, na criação das

r2
S
S
S
N
N
X

Figura 16: Regra a ser buscada e posteriormente inserida na TDA.

		Regras					
		r1	r2	r3	r4	r5	r6
c1		S	N	N	S	S	S
c2		N	S	S	S	S	S
c3		S	N	N	N	S	S
c4		S	N	N	N	S	N
c5		N	S	N	N	N	N
a1		X			X		
a2			X	X		X	X

Figura 17: Configuração t_1 da Tabela de Decisão Adaptativa.

regras é necessário consultar os pesos dos critérios e alternativas para verificar qual alternativa seria selecionada por esta regra. Supondo-se neste caso que uma reconstrução global seja necessária, ou seja, o nó sorvedouro envia por meio de uma mensagem de *broadcast* uma mensagem do tipo *global*, indicando para todos os nós da rede que os mesmos devem escolher um novo nó pai e reconstruir a lista de possíveis pais. Esta reconstrução utiliza a mesma idéia da construção inicial. Uma possível configuração após a execução é apresentada na Figura 18.

Uma observação importante é que sempre que um nó é escolhido como pai de outro nó, o mesmo retira o filho de sua lista de possíveis pais. Na Figura 18, o nó v_5 escolheu o nó v_6 como pai e, portanto, o nó v_6 retirou v_5 da sua lista de possíveis pais. Isto é necessário para que nunca ocorra de um nó ser pai e filho ao

Figura 18: Configuração c_2 do grafo dinâmico que representa a RSSF.

mesmo tempo de outro nó, o que ocasionaria um *loop* de roteamento. Uma outra observação importante é que ao inserir v_{10} na rede, após uma reconstrução v_{10} poderá ser adicionado à lista de possíveis pais de v_6 , caso outro nó, por exemplo, v_3 passe a ser o pai de v_{10} .

Os demais eventos que podem ocorrer são semelhantes aos apresentados nesta seção. É importante observar que as mudanças de configuração do grafo dirigido dinâmico e da TDA são dependentes, ou seja, uma tomada de decisão utilizando a TDA influencia na topologia da rede e, conseqüentemente, nos vértices e arestas do grafo. Além disso, cada mudança na topologia da RSSF corresponde à execução de uma função adaptativa de inclusão ou de exclusão.

Nota-se que para redes pequenas o mecanismo funciona a contento, pois na ocorrência de falhas, o tempo de resposta é razoável o suficiente para que se recupere das falhas sem que haja prejuízos significativos na comunicação da rede. Entretanto, para redes grandes, em caso de falhas que ocorrerem em nós mais periféricos (mais próximos das folhas da árvore), o tempo de resposta é proporcional a aproximadamente duas vezes a altura da árvore (tempo da mensagem de alerta e tempo da mensagem de reconstrução), o que pode ser muito longo.

Uma das formas de reduzir este tempo é a utilização de nós especiais que sejam responsáveis pela reconstrução local da topologia, ou seja, além do nó sorvedouro possuir uma TDA e um grafo dirigido dinâmico de toda a rede, alguns nós da rede poderiam armazenar TDA e Grafos Direcionados Dinâmicos locais que contenham informações parciais da rede. De posse destas informações parciais, estes nós especiais, que serão denominados sorvedouros intermediários, poderiam realizar tomadas de decisão que envolvem reconstrução local da rede.

O funcionamento deste nós seria análogo ao funcionamento do nó sorvedouro principal, com a diferença que apenas mensagens do tipo *local* seriam enviadas. Com a adição deste mecanismo, seria interessante que cada um dos nós da rede

pudesse identificar por meio de algum código, a qual nó sorvedouro intermediário o mesmo está ligado.

Com esta mudança, sempre que um nó recebe uma mensagem do tipo *local* o mesmo deve verificar se o sorvedouro intermediário é o mesmo que ele está ligado. Se for, uma reconstrução local é executada e a busca por um novo pai é realizada localmente e a mensagem é propagada para os demais nós. Caso contrário, a mensagem é descartada e nenhuma ação é realizada. Este mecanismo é importante para evitar que reconstruções locais desnecessárias sejam executadas.

É importante observar que sempre que uma mensagem do tipo *global* é recebida o mecanismo de reconstrução da topologia não é alterado, apesar da inserção de nós sorvedouros intermediários.

4.3 Algoritmo Proposto

Nesta seção são apresentadas uma representação gráfica do algoritmo proposto e decisões que foram tomadas durante a sua especificação para garantir seu funcionamento, sem que haja perda da conectividade da rede.

Uma observação importante é que após uma análise mais detalhada sobre as Tabelas de Decisão Adaptativa e a Tabela de Decisão Adaptativa Estendida, verificou-se que o *overhead* gerado pelo método multicritério AHP é muito alto para o Algoritmo ADAP_RSSF proposto. Segundo Tchemra (2009), o tempo de execução da TDAE depende das entradas e do número de funções auxiliares e funções adaptativas a serem utilizadas e pode ser quadrático em relação ao número de regras para algumas operações. Detalhes do tempo de execução podem ser obtidos em Tchemra (2009).

Além disso, foi verificado que, na maioria dos casos, problemas relacionados à reconstrução da topologia não envolvem vários critérios, mas sim dois critérios. Um outro fator é que os critérios em uma RSSF, de maneira geral, não possuem pesos entre si.

Dessa forma, para evitar que o custo computacional seja elevado, já que em geral os sensores possuem restrições de processamento, decidiu-se optar pela Tabela de Decisão Adaptativa proposta por Neto (2001). Apesar disso, é importante o estudo da Tabela de Decisão Adaptativa Estendida como uma alternativa para problemas de decisão multicritério e, principalmente, para analisar sua aplicabilidade em outros problemas relacionados a RSSFs.

4.3.1 Especificação do Algoritmo

Nesta subsecção é apresentada a especificação dos mecanismos de comunicação e manutenção de topologia algoritmo ADAP_RSSF. Para uma maior compreensão e facilidade de leitura utilizou-se Diagramas de Nassi-Shneiderman, que foram desenvolvidos em 1972 por Nassi e Shneiderman (1973), e são utilizados para representar graficamente códigos estruturados. OS pseudo-códigos referentes a estes diagramas encontram-se no Apêndice A.

Para o algoritmo foram definidos os seguintes tipos de nós:

- SINK: este é o nó sorvedouro principal e responsável por todo o funcionamento do mecanismo.
- MIDDLE_SINK: são nós responsável pela reconstrução local da topologia.
- INTERMEDIATE: nós pais, responsáveis pelo encaminhamento de pacotes.
- COMMON: nós folha, responsáveis por gerar dados.

A Figura 19 mostra o procedimento Receive do algoritmo.

Figura 19: Procedimento Receive do Algoritmo ADAP_RSSF.

O Procedimento Receive apresentado na Figura 19 é responsável por determinar qual das sub-rotinas serão chamadas, de acordo com o tipo de nó. Cada tipo de nó, SINK, MIDDLE_SINK, INTERMEDIATE e COMMON possui um funcionamento, apresentado nas Figuras 20, 21, 22 e 23, respectivamente.

Figura 20: Funcionamento do Recebimento de Pacotes no Nó SINK.

Figura 21: Funcionamento do Recebimento de Pacotes no Nó MIDDLE_SINK.

O procedimento de envio do pacote é apresentado na Figura 24. As informações a serem adicionadas ao pacote são passadas por parâmetro.

A especificação da Tabela de Decisão Adaptativa pode ser visualizada na Figura 25.

4.3.2 Análise do Comportamento do Algoritmo Proposto ADAP_RSSF

Para análise do desempenho de algoritmos de RSSF podem ser utilizadas diversas métricas, tais como taxa de entrega de dados, consumo de energia do protocolo,

Figura 22: Funcionamento do Recebimento de Pacotes no Nó INTERMEDIATE.

tempo de vida da rede, escalabilidade, tolerância a falhas, *hop count* e outras (TILAK; ABU-GHAZALEH; HEINZELMAN, 2002).

Nesta seção será realizada uma análise do comportamento do algoritmo proposto do ponto de vista do contador de saltos (*hop count*). Esta métrica foi escolhida em virtude da mesma possibilitar a análise de tráfego da rede de maneira relativamente fácil e eficiente. Além disso, métricas como consumo de energia, tempo de vida da rede e tolerância a falhas dependem de análise de resultados de implementação e/ou simulação.

Considerando o *hop count*, define-se h_i a altura do nó v_i na árvore até o nó sorvedouro principal, e h_{mi} a altura do nó v_i em relação à sub-árvore que tem como raiz o nó sorvedouro intermediário ao qual v_i está conectado. É importante observar que sempre $h_{mi} \leq h_i$.

Serão considerados para efeitos de análise, a transmissão de dados sensoriados e a recuperação quando algum problema de conectividade da rede ocorrer. No caso de transmissão de dados sensoriados, o *hop count* é a distância do nó fonte ao nó sorvedouro principal, ou seja, equivale a h_i . Em relação aos algoritmos definidos na Seção 2.7, o *hop count* do algoritmo ADAP_RSSF proposto é o mesmo. Entretanto, os mecanismos como LEACH e SPIN dependem de mecanismos definidos na aplicação para funcionarem adequadamente. Além disso, o SPIN utiliza

Figura 23: Funcionamento do Recebimento de Pacotes no Nó COMMON.

Procedimento send
define o tipo de pacote
adiciona o destinatário do pacote
adiciona o remetente do pacote
define o critério do pacote
configura o tipo de nó do pacote
envia o pacote

Figura 24: Funcionamento do Envio de Pacotes.

flooding para propagação das informações, o que gera um número de mensagens maior que o ADAP_RSSF. Os algoritmos LEACH e REMUDA têm a vantagem em relação ao ADAP_RSSF por possuírem mecanismos de agregação de dados (no caso do REMUDA, apenas para dados não urgentes). Assim, mesmo o *hop count* sendo igual, o número de mensagens trocadas pode ser inferior no REMUDA e no LEACH. No caso do MULTI, o número de mensagens depende do tráfego da aplicação. Se não houver um caminho até o nó sorvedouro estabelecido no MULTI, o ADAP_RSSF será mais eficiente, pois não necessita do *overhead* para o estabelecimento do caminho. Caso haja um caminho estabelecido, o MULTI e o ADAP_RSSF possuem funcionamento semelhante.

No caso de problemas que envolvem falhas de conectividade na rede o algoritmo ADAP_RSSF possui duas possibilidades: reconstrução local e reconstrução

Figura 25: Especificação da TDA.

global, além da lista de possíveis pais. A reconstrução global ocorre apenas com o envolvimento do sorvedouro principal, enquanto a reconstrução local pode ser realizada tanto pelos sorvedouros intermediários quanto pelo sorvedouro principal. Uma observação importante também é que nem todos os nós pai são sorvedouros intermediários.

Se uma falha ocorrer em um nó a reconstrução pode ser realizada diretamente com a escolha de um novo pai na lista de possíveis pais, o que não gera nenhum tráfego na rede e, portanto, não importa o *hop count*.

Em caso de uma reconstrução local, a informação deve ser enviada apenas ao nó sorvedouro intermediário, ou seja, o *hop count* seria de aproximadamente $2 * h_{mi}$, ou seja, h_{mi} saltos para a informação de falha chegar no nó sorvedouro intermediário e h_{mi} saltos para a mensagem de reconstrução local chegar ao nó. Uma observação importante é que a mensagem de ida é enviada utilizando-se *unicast* e a mensagem de volta é propagada utilizando-se *broadcast* para que todos os nós ligados ao nó sorvedouro intermediário restabeleçam os caminhos até o mesmo. Nota-se que nem todos os nós da rede recebem esta mensagem, pois se um nó não ligado ao sorvedouro intermediário que enviou o *broadcast* receber a mensagem, o mesmo descarta-a e não faz a propagação da mesma.

Em comparação com o REMUDA, esta abordagem apresenta o mesmo *hop*

count, porém, a mensagem de busca de nova topologia no REMUDA é propagada para todos os nós da rede. No algoritmo MULTI, se uma rota for estabelecida no momento do envio, o custo será de $2 * h_i$, pois esta rota deve ser estabelecida até o nó sorvedouro; se um caminho já tiver sido estabelecido, a reconstrução de um caminho também envolve todos os nós da rede, o que também resulta num *hop count* de $2 * h_i$, que é sempre maior ou igual a h_{mi} . Os algoritmos SPIN, LEACH e Directed Diffusion não apresentam mecanismos de reconstrução de rotas em sua descrição.

Se a reconstrução for global, todos os algoritmos apresentam o mesmo desempenho em relação ao *hop count*.

Apesar dos *hop counts* serem próximos, uma das vantagens do algoritmo ADAP_RSSF é a possibilidade de incorporar informações não previstas no funcionamento do mesmo, por meio da utilização de uma TDA. Entretanto, esta possibilidade envolve um *overhead* adicional na tomada de decisão.

A Tabela 4 apresenta um resumo das comparações do algoritmo ADAP_RSSF com outros algoritmos, levando-se em consideração o *hop count* e o número efetivo de mensagens.

Tabela 4: Comparação do Algoritmo ADAP_RSSF com outros algoritmos no caso de Reconstrução Local.

Algoritmo/Métrica	<i>hop count</i>	Número de Mensagens
REMUDA	= ADAP_RSSF	$\geq$ ADAP_RSSF
MULTI (Rota Já definida)	= ADAP_RSSF	= ADAP_RSSF
MULTI (Nova Rota)	$\geq$ ADAP_RSSF	$\geq$ ADAP_RSSF

Analisando a Tabela 4, é possível verificar que o algoritmo ADAP_RSSF apresenta desempenho melhor ou igual a outros algoritmos existentes. As comparações em relação ao envio de dados sensorizados e à reconstrução global não foram apresentados, pois estes algoritmos possuem o mesmo desempenho.

4.3.3 Algumas Considerações sobre o Algoritmo

Em relação à especificação do algoritmo ADAP_RSSF, foi necessária alteração do mesmo, que ao invés de utilizar uma TDAE, utilizou uma TDA, em virtude do número de critérios ser fixo e da complexidade computacional da TDAE, que é quadrática em relação ao número de critérios, além do tempo de execução do método AHP ser elevado, conforme apresentado em Tchemra (2009).

Para reduzir o tempo computacional da consulta na Tabela de Decisão Adaptativa foram adotadas algumas alternativas na especificação do algoritmo. A primeira delas é dividir a Tabela Adaptativa em duas, de acordo com o tipo de nó: uma TDA para o recebimento de mensagens do tipo alerta dos nós comuns (COMMON) e uma TDA para nós pais (INTERMEDIATE). É importante observar que isto só foi possível porque sempre um dos critérios é o tipo de nó, ou seja, sempre o tipo de nó que envia mensagens para o nó sorvedouro é COMMON ou INTERMEDIATE.

Outro aspecto importante é que alguns critérios para mudança da topologia podem ser previstos. Dessa forma, ao invés de inserir estes critérios na TDA decidiu-se utilizar estruturas condicionais, já que estes critérios sempre deverão ser consultados em caso de ocorrência de algum evento. Para os critérios não definidos previamente, é necessário inseri-los na Tabela de Decisão Adaptativa, ou seja, apenas critérios não previstos são inseridos na TDA. Assim, inicialmente a TDA está vazia e sempre que um critério não previsto for encontrado, uma nova regra é inserida na TDA. A vantagem desta abordagem é que não há a necessidade de consulta na TDA se o evento ocorrido na rede for conhecido previamente.

4.3.3.1 Restrições do Algoritmo

Uma das principais limitações do mecanismo no início do seu desenvolvimento é que todo o processamento de reconstrução da topologia era definido em um único nó sorvedouro (denominado no algoritmo de nó sorvedouro principal). O problema deste fato é que em alguns casos o tempo de processamento para reconstrução da topologia pode ser elevado dependendo da distância do nó em relação ao nó sorvedouro (raiz da árvore). Por exemplo, se um vizinho do nó sorvedouro envia uma mensagem de controle para alteração de topologia é necessário apenas um salto para que seja realizada alguma ação de reconstrução. Entretanto, se um nó está próximo às extremidades da topologia (folhas da árvore), o número de saltos é proporcional à altura da árvore o que pode demandar um tempo de processamento mais elevado, podendo ocasionar perda de pacotes, já que a reconstrução da topologia depende do recebimento dos pacotes de controle.

Uma possível solução para esta restrição foi a implementação da Tabela de Decisão Adaptativa em alguns nós intermediários, construindo sub-árvores para resolver problemas de reconstrução local de topologia, cujas raízes foram denominadas de nós sorvedouros intermediários. Estes nós devem ser nós com um poder de processamento mais elevado que os nós comuns. Entretanto, devido à limitação

de recursos computacionais dos nós sensores convencionais esta implementação ainda é difícil de ser adotada. Para que isso seja possível, são necessários nós com maior capacidade de processamento e armazenamento, já que é necessário armazenar informações de roteamento em sua memória. Trata-se de um desafio que enfrentam os fabricantes de RSSFs, uma vez que existe o compromisso de manter o baixo consumo de energia dos nós, que por outro lado devem possuir razoável capacidade de processamento e armazenamento, que levam a um maior consumo.

Um outro problema encontrado na implementação é que sempre que um novo evento ocorra na rede, é difícil identificar se algum dos critérios já adicionados na Tabela de Decisão Adaptativa corresponde ao evento em questão. Dessa forma, em alguns casos, o mesmo evento poderá ser adicionado na Tabela de Decisão Adaptativa mais de uma vez.

4.4 Considerações sobre a Simulação

Esta seção descreve aspectos de simulação e de comportamento do algoritmo proposto para Manutenção de Topologia especificado anteriormente. Alguns simuladores e *frameworks* podem ser utilizados para isso, tais como o TinyOS (TINYOS, 2006), NS-2 (NS-2, 2008) e MannaSim (MANNASIM, 2008).

O NS-2 é um simulador de eventos discretos cujo objetivo é o desenvolvimento de pesquisas na área de redes. Este simulador é bastante utilizado para simular novos algoritmos e protocolos.

Outra possibilidade é utilizar o TinyOS (TINYOS, 2006), que consiste em um Sistema Operacional de código aberto, baseado em componentes, e que também é utilizado para simulação em RSSFs.

O *framework* MannaSim (MANNASIM, 2008) também se constitui em uma possibilidade de simulador a se considerar. Este *framework* oferece um ambiente de simulação para RSSF, que funciona em conjunto com o NS-2, e permite o projeto, desenvolvimento e análise de diferentes aplicações em RSSF. Entretanto, para a utilização do MannaSim seria necessário também alterar as configurações do NS-2.

O NS-2 foi escolhido como base para a descrição de uma possível implementação da simulação, para investigar o funcionamento efetivo do algoritmo proposto. A escolha se deveu ao fato de ele permitir que novos protocolos pudessem ser utilizados na comunicação, o que não ocorre com o MannaSim. Além

disso, ele é mais difundido que o TinyOS, que foi desenvolvido para utilização de sistemas semelhantes aos nós sensores da Crossbow.

A seguir, será apresentada uma proposta de simulação baseada no NS-2. Dentre as dificuldades que podem ser encontradas na simulação, podem ser citadas a utilização de duas linguagens de programação diferentes (C++ e Tcl), a visualização dos resultados, que necessita de ferramentas específicas, já que as informações de saída são geradas em diversos arquivos textos. Assim, uma observação importante é que mesmo com as informações apresentadas nesta seção, a implementação da infra-estrutura de simulação necessária no NS-2, bem como a configuração dos *scripts* exigidos não se mostrou trivial.

A primeira etapa para a implementação de um novo algoritmo é conhecer o funcionamento do NS-2 por meio da execução de algoritmos já implementados e do uso de ferramentas adicionais, como por exemplo o NAM, para análise dos resultados e geração de gráficos e tabelas com os resultados da simulação. Informações detalhadas sobre o funcionamento do NS-2 podem ser obtidas em NS-2 (2008).

Uma outra restrição é que o NS-2 não possibilita testar o nível de energia dos nós sensores. Dessa forma, a ocorrência de eventos em relação à energia é realizada em instantes pré-determinados. Uma alternativa seria simular a energia dos nós sensores por meio de contadores, ou simular eventos em determinados períodos de tempo. Além disso, a documentação a respeito do desenvolvimento de novos algoritmos utilizando o NS-2 é bastante escassa e de difícil acesso.

Para implementar o algoritmo ADAP_RSSF seria necessário a criação de um novo diretório no diretório base do NS-2. A implementação do algoritmo a ser realizada deve utilizar a linguagem C++ e os *scripts* gerados em Tcl. Neste diretório devem ser criados, no mínimo, os seguintes arquivos:

- adap_rssf.h: neste arquivo de cabeçalho devem ser especificados todos os temporizadores e agentes de roteamento que desempenham o funcionamento do protocolo;
- adap_rssf.cc: neste arquivo devem ser implementados todos os agentes e temporizadores definidos no arquivo adap_rssf.h, além dos mecanismos de ligação entre o C++ e o Tcl para execução dos *scripts*;
- adap_rssf_pkt.h: neste arquivo devem ser definidos os tipos de pacotes do algoritmo ADAP_RSSF.

Para a criação de um novo protocolo no NS-2 é necessário criar um Agente que é derivado da classe *Agent*. Esta é a classe principal a ser utilizada na implementação do protocolo ADAP_RSSF. Além disso, esta classe oferece suporte para a ligação com *scripts* Tcl, possibilitando que o protocolo de roteamento possa ser utilizado usando *scripts* Tcl.

Os *Agents* representam sistemas finais nos quais os pacotes são gerados e consumidos. Esta classe é implementada parcialmente em C++ e parcialmente em Tcl, para que haja ligação entre os *scripts* Tcl e a implementação do protocolo (NS-2, 2008). Esta classe é responsável pelo roteamento de dados no algoritmo.

Além disso, é necessário definir o formato dos pacotes. Este formato pode ser definido no arquivo *adap-rssf-pkt*, com os pacotes possuindo os seguintes campos:

- *pkt_src_*: endereço de origem do pacote;
- *pkt_dest_*: endereço de destino do pacote;
- *pkt_type_*: tipo do pacote. O tipo de pacote será utilizado para determinar a ação do protocolo. Os pacotes apesar de possuírem nomes, são valores numéricos que iniciam em 0;
- *pkt_criteria*: tipo de critério. Este tipo de critério é utilizado apenas quando o pacote for do tipo alerta, pois estas mensagens podem causar mudanças na topologia da rede. Quando a mensagem for de outro tipo seu valor é -1. Caso contrário, seu valor varia de 0 até *num_criterios* - 1;
- *pkt_node_type_*: tipo do nó que enviou o pacote. O tipo de nó é utilizado também para determinar a ação do protocolo. Esta campo possui quatro valores possíveis: COMMON, MIDDLE_SINK, SINK and INTERMEDIATE. O tipo COMMON é o tipo de nó comum que não executa nenhuma operação de comunicação na rede, apenas se um evento ocorrer. O tipo de nó INTERMEDIATE é responsável pela propagação das informações na rede (Os pais dos nós sensores). O nó SINK é único e é responsável por toda a configuração da rede e ações de reconstrução global e local. Os nós do tipo MIDDLE_SINK são responsáveis pela reconstrução local da topologia.

O atributo *offset_* é utilizado para acessar os pacotes efetivamente. Após a implementação do formato do pacote é necessário fazer a ligação (*bind*) do

cabeçalho do pacote com o Tcl. Para executar esta ação, é necessário a implementação do arquivo *adap_rssf.cc*.

Além disso, é necessário definir um novo tipo de pacote, que será utilizado como pacotes de controle. No presente caso, o campo *pkt_type_* é responsável por isso. Para o controle dos instantes de ocorrência do de eventos é necessário utilizar objetos da classe *Timer*. Esta classe é utilizada para realizar o escalonamento de tarefas dentro da rede. A classe *Trace* é fundamental para registrar as informações nos arquivos de *log*.

Para implementar do protocolo é necessário realizar a implementação do agente propriamente dito. Esta implementação deve ser realizada nos arquivos *adap_rssf.h* e *adap_rssf.cc*. No arquivo *adap_rssf.h* está definida a classe *Adap_rssf* que possui os atributos e os métodos utilizados para acessar os atributos desta classe. No protocolo proposto serão considerados apenas o envio de pacotes de controle com a finalidade de verificar a funcionalidade do novo mecanismo.

Para a implementação são necessários alguns arquivos pré-definidos ou existentes no NS-2:

- *adap_rssf_pkt.h*: define o formato do pacote do protocolo;
- *common/agent.h*: é utilizada como classe base para os agentes;
- *common/packet.h*: é utilizada como classe base para os pacotes;
- *common/time-handler.h*: é utilizada como classe base para a criação de temporizadores;
- *trace/trace.h*: utilizado para gerar informações e resultados das simulações em arquivos de saída;
- *tools/random.h*: utilizado para a geração de números pseudo-aleatórios;
- *classifier/classifier-port.h*: utilizado para a classificação das portas a serem utilizadas pelas camadas superiores.

Além disso, foram criadas duas macros com o intuito de auxiliar no funcionamento do protocolo:

- *CURRENT_TIME*: para obter a hora atual da simulação;

- JITTER: utilizado para gerar números aleatórios para o envio de pacotes de controle para evitar colisões e que vários nós enviem dados no mesmo instante.

Neste arquivo está definida ainda uma classe para o controle de envio de pacotes de controle. Esta classe, denominada de *adap-rssf-timer*, é derivada da classe *Time-Handler* e é utilizada informar ao agente de roteamento que o mesmo deve enviar um pacote de controle.

Outra classe definida é *Adap-rssf*, que define as informações principais do protocolo e uma variável (*accessible_var*), que é utilizada para a escrita e leitura a serem realizadas pelos scripts Tcl, evitando que o NS-2 seja recompilado a cada mudança no comportamento do protocolo.

Um importante atributo definido nesta classe é o objeto do tipo *Trace*, que pode ser utilizado para gravar informações da Tabela de Roteamento em logs.

Para auxiliar no envio dos pacotes foram definidos alguns métodos, para o envio e recebimento de pacotes de controle (*forward_data*, *recv_adap-rssf-pkt* e *send_adap-rssf-pkt*). Além disso, são definidos um construtor e os métodos públicos *recv* e *command*, que são utilizados quando um nó recebe um pacote e pelo Tcl, respectivamente.

No arquivo *adap-rssf.cc* estão as implementações dos métodos e demais classes utilizadas pelo protocolo e pelo Tcl no acesso a informações. Uma das classes é denominada de *Adap-rssfClass* que é derivada da classe *TclClass* para obter informações sobre o protocolo por meio de um objeto Tcl.

Para a implementação dos protocolos foi utilizado um temporizador que determina quando os pacotes de controle devem ser enviados.

4.4.1 Implementação do Agente

A implementação do Agente é a parte principal e é realizada por meio de um construtor da classe *Adap-rssf*, no qual são utilizados comandos para que as informações da classe possam ser acessadas pelo *script* Tcl. Este acesso é realizado por meio de uma variável lógica (booleana) denominada de *accessible_var* e pela função *bind_bool()*.

Para realizar as operações do Agente por meio do método *command*. Neste método são implementadas as ações iniciais a serem realizadas pelo agente.

Uma das operações que o agente deve realizar é o recebimento de pacotes.

Para a implementação desta operação foi criado um método *recv*. Neste método são considerados apenas pacotes de controle do algoritmo *ADAP_RSSF* e pacotes de dados para verificar a viabilidade do mecanismo adaptativo. Para o recebimento de um pacote do protocolo *ADAP_RSSF* pode ser utilizado o método *recv_adap_pkt* que processa os pacotes de controle do protocolo de acordo com o tipo e a especificação do mesmo.

O método *send_adap_pkt* deve ser criado para o envio de pacotes de controle do protocolo. O primeiro passo no envio de pacote é a alocação do mesmo por meio da função *allocpkt* definido para os agentes. Em seguida, são definidos os campos do pacote *adap_rssf*.

Estes pacotes são encapsulados em outros pacotes de nível superior para serem enviados pela rede. Após o encapsulamento, os pacotes de controle são enviados pela rede por meio de uma função denominada *schedule* que recebe como parâmetro um objeto Tcl.

4.4.2 Alterações no NS-2

Para que o NS-2 reconheça o novo protocolo a ser implementado é necessário fazer mudanças com a finalidade de incorporar o novo protocolo ao simulador do NS-2.

Uma das mudanças a ser realizada é informar o tipo de pacote do protocolo que foi desenvolvido. No caso do NS-2, os tipos de pacote são definidos por meio de uma enumeração no arquivo *commom/packet.h*. Dessa forma, foi definida uma constante *PT_ADAP_RSSF* na enumeração *packet_t* do arquivo *packet.h*. No mesmo arquivo é definida uma classe chamada *p_info* que contém informações sobre os pacotes. Uma das informações é um vetor *name_* que armazena em formato texto os tipos de pacotes. Para definir um nome textual é necessário alterar o construtor da classe *p_info* e adicionar um nome textual para o protocolo criado.

4.5 Considerações sobre o Capítulo

Este capítulo apresentou uma proposta de algoritmo para manutenção de topologia em RSSF. O algoritmo utiliza Tecnologia Adaptativa para garantir o funcionamento da rede, mesmo que situações não previstas ocorram. Dentre os dispositivos adaptativos usados no algoritmo a Tabela de Decisão Adaptativa.

Além disso, foram apresentadas especificação do algoritmo e comentários sobre possíveis implementações e simulações

Com o objetivo de comparar o algoritmo proposto com outros existentes, adotou-se a métrica *hop count*, concluindo-se que baseado nesta métrica o algoritmo possui desempenho semelhante aos existentes.

Também foi apresentado e definido o mecanismo de grafo dinâmico, que mostra um mapeamento do estado da topologia da rede, cujas mudanças ocorrem em virtude de eventos, conforme apresentado pelo mecanismo que utiliza a Tabela de Decisão Adaptativa.

5 Conclusões

Neste trabalho é apresentada uma proposta de manutenção de topologia de RSSFs com entrega de dados Híbrida, utilizando Tabelas de Decisão Adaptativa, com o intuito de manter a conectividade da rede. A idéia de utilizar Tecnologia Adaptativa na manutenção de topologia de RSSFs surgiu devido ao fato da necessidade de auto-configuração inerente a elas, o que acarreta na necessidade de constantes mudanças em sua topologia.

Como o usuário não pode ou não interage com a RSSF, especialmente na parte de roteamento e envio de dados, a utilização de mecanismos automáticos é fundamental. O uso de TDA mostrou que em relação à métrica de *hop count* o mecanismo apresenta desempenho semelhante ou melhor do que alguns protocolos existentes e já conhecidos. Além disso, em alguns casos, apesar do *hop count* ser o mesmo, o número de mensagens enviadas pelo algoritmo ADAP_RSSF é menor pelo fato de o mesmo utilizar *unicast* ao invés de *broadcast* em algumas situações.

Além disso, como a proposta é mostrar a viabilidade de aplicar a TDA na reconstrução de topologia de RSSF, alguns critérios, como falha de um nó e nó com baixa energia foram estabelecidos, pois podem influenciar no tempo de vida da rede e na comunicação da rede, independente de se utilizar uma TDAE ou TDA. Entretanto, não fez parte do escopo deste trabalho o estudo de técnicas e métodos de detecção de falhas em RSSFs, nem de medição do nível de energia dos nó sensores. Algoritmos e métodos de detecção de falhas e detecção do nível de energia dos nós em RSSF podem ser encontrados em Chen, Kher e Somani (2006), Shebli et al. (2007), Lee e Choi (2008), Sun et al. (2008), Jiang (2009), Choi et al. (2009).

Na etapa de especificação do algoritmo proposto neste trabalho foi discutido que a Tabela de Decisão Adaptativa Estendida é um mecanismo adaptativo extremamente custoso e desnecessário para esta abordagem para controle de Topologia em RSSF, pois sempre se tem dois critérios: tipo de nó e tipo de mensagem de alerta. Para que a Tabela de Decisão Adaptativa fosse utilizada, decidiu-se utilizar a Tabela de Decisão Adaptativa convencional, por meio do desenvolvimento

de um novo algoritmo. Em seguida, este trabalho apresentou a especificação dos procedimentos de envio e recebimento de pacotes definidos neste novo protocolo.

Assim, esta tese propôs um novo algoritmo que integra duas áreas distintas: Redes de Sensores Sem Fio e Tecnologia Adaptativa. Dessa forma, o algoritmo proposto permite o mapeamento da manutenção da topologia de uma RSSF em um Grafo Dirigido Dinâmico e tratamento de eventos com uma TDA, com a finalidade de garantir que sempre que um evento que possa influenciar na comunicação da rede ocorra, a topologia da rede deva ser configurada novamente (local ou globalmente).

Uma observação importante é que o número de regras e critérios atualmente presentes no problema de Controle de Topologia de RSSF tendem a ser finitos e que no caso do uso da TDA haverá apenas inclusão de regras e critérios (não haverá exclusão). Além disso, as ações adaptativas serão executadas sempre após a inclusão de uma regra e nunca antes.

Esta tese discute também a utilização do simulador NS-2 para uma possível implementação do algoritmo ADAP_RSSF. Este simulador, entretanto, não modela as reais características de um nó sensor, como mecanismos nativos para a medição de energia, pois o mesmo foi desenvolvido para a simulação de redes convencionais cabeadas.

Este trabalho traz contribuições nos aspectos teóricos de manutenção de Topologia de Redes de Sensores Sem Fio e na Aplicação de Tecnologia Adaptativa, conforme apresentado na próxima seção.

5.1 Contribuições

A proposta de aplicação da TDAE e da TDA em RSSF apresentada neste trabalho tem como uma das contribuições a integração da Tecnologia Adaptativa com a área de Redes de Sensores Sem Fio, mostrando que multidisciplinaridade e a utilização de teorias bem fundamentadas pode ser bastante interessante e importante na resolução de problemas.

Uma outra contribuição que pode ser citada é que devido a restrições tecnológicas, torna-se difícil a aplicação da TDAE em RSSFs convencionais, devido ao seu custo de processamento e armazenamento necessários pelos métodos multicritérios, conforme citado na Seção 4.3.3.3 e detalhado em (TCHEMRA, 2009).

Do ponto de vista da Tecnologia Adaptativa esta tese contribui no sentido

de mostrar a viabilidade de uma nova aplicação na qual a Tecnologia Adaptativa, em particular Grafos Dinâmicos e a TDA, pode ser utilizada com pequenas modificações.

Além disso, é apresentada também a formulação de um grafo dinâmico como dispositivo adaptativo e mostra sua aplicação para ilustrar graficamente as mudanças de topologia em uma RSSF.

Uma contribuição conceitual do ponto de vista de RSSF é a utilização de uma nova técnica para a manutenção de topologia. Os algoritmos adaptativos ou dinâmicos apresentados na literatura envolvem sempre a escolha entre diversos critérios, eventos e comportamentos já previamente estabelecidos, ou seja, caso algum comportamento não considerado pelo algoritmo fosse encontrado o algoritmo não teria como resolvê-lo de maneira adequada.

Com a utilização da TDA, é possível que novas regras, critérios ou alternativas possam ser incorporados ao mecanismo de tomada de decisão e mesmo os comportamentos não previstos possam ser devidamente considerados na manutenção da topologia da rede.

5.2 Trabalhos Futuros

Em relação ao algoritmo proposto, as seguintes atividades complementares podem ser sugeridas:

- Avaliar o que é mais custoso, em relação a tempo de processamento, entre manter poucos critérios e regras iniciais, e inserir regras e critérios na medida em que forem necessários, ou inserir todas as regras e critérios necessários e gastar tempo adicional na busca de regras na Tabela de Decisão;
- Determinar a melhor estratégia de escolha e distribuição dos nós sorvedouros intermediários, com o intuito de melhorar o desempenho do algoritmo e reduzir o tráfego na rede, sem que haja prejuízo na conectividade da rede e sem que haja um número muito grande de nós sorvedouros intermediários;
- Avaliar o algoritmo de acordo com outras métricas, como consumo de energia, número efetivo de mensagens trocadas, distribuição de nós, dentre outras;
- Simular o consumo de energia de uma bateria por meio de contadores regressivos que vão sendo decrementados a cada instante de tempo e a cada

ação de sensoriamento, envio e recebimento de pacotes, com a finalidade de incorporar mais mecanismos ao algoritmo;

- Incorporar mecanismos adicionais ao algoritmo, como, por exemplo, agregação de dados e detecção de falhas.

Com os avanços da tecnologia, em especial, o aumento da capacidade de processamento e armazenamento dos nós sensores, mas com um consumo baixo de energia, as seguintes abordagens podem ser levadas em consideração:

- Implementação da TDAE no problema de roteamento em RSSFs. Em caso da implementação utilizando uma TDAE, uma sugestão de continuidade deste trabalho é a de avaliação de diversos valores para os critérios com a finalidade de detectar qual a melhor configuração se ajusta melhor aos diferentes tipos de comunicação existentes em RSSFs. Por exemplo, classificar os pesos dos diversos critérios de maneira a identificar quais possuem desempenho melhor em RSSFs nas quais os tráfegos são mais esporádicos, quais são mais eficientes em redes com alto volume de tráfego e assim por diante.
- O estudo da viabilidade de outros métodos multicritério envolvendo uma TDAE, para verificação de quais deles mais se ajustariam ao problema de Manutenção de Topologia. Dessa forma, a comparação entre diversos métodos permitiria determinar qual método consumiria menos recursos dos nós da rede.

A utilização de Tecnologia Adaptativa em outros problemas relacionados à área de RSSFs se mostra promissora para a realização de novos trabalhos. Como exemplo, poder-se-ia citar o estudo sobre a cobertura da área a ser monitorada por uma RSSF. Nesse caso, o estudo de técnicas de Otimização, como Programação Linear, associadas à Tecnologia Adaptativa com a finalidade de modelar o consumo de energia em uma determinada RSSF, mesmo quando variáveis e regras do problema possam ser modificadas em virtude de mudanças nos nós da rede, corresponde a um tema de pesquisa a ser explorado.

Neste caso, como os métodos de Otimização, em geral, demandam um alto poder de processamento, a única possibilidade de implementação com os recursos tecnológicos disponíveis atualmente, seria a implementação a ser realizada no nó sorvedouro.

Uma outra abordagem seria a implementação de um algoritmo de controle de topologia que tenha como dispositivo principal outro dispositivo adaptativo, como um grafo dinâmico, por exemplo, que neste trabalho foi utilizado para representar a topologia da rede, mas não para a tomada de decisão. No grafo dinâmico, os vértices representam os nós da rede e uma aresta (i, j) significa que existe comunicação entre o nó i e o nó j .

Assim, sempre que um nó j fosse desligado, para representar que não há comunicação com este nó, todas as arestas que chegassem a j ou partissem de j seriam removidas. Analogamente, sempre que um nó j fosse inserido, assim que o mesmo estabelece comunicação com outros nós sensores, arestas seriam inseridas no grafo dinâmico.

Neste dispositivo a adaptatividade aparentemente pode ser mais bem aproveitada, pois existirão inúmeras inserções e remoções de nós e arestas durante o tempo de vida da rede. Devido a restrições de processamento, uma alternativa para a implementação deste mecanismo é a implementação do grafo dinâmico no nó sorvedouro.

Por fim, como continuidade a este trabalho tem-se a conclusão da implementação utilizando o NS-2 para avaliação de outras métricas do algoritmo proposto e a utilização de outros simuladores como J-SIm e Sense 3.0 para verificação do comportamento do mecanismo, já que estes simuladores, apesar de não serem usados na mesma proporção que o NS-2, foram desenvolvidos especificamente para integrar todas as características das RSSFs.

5.3 Considerações Finais

A Tecnologia de Redes de Sensores Sem Fio está em constante evolução e diversos mecanismos de comunicação têm sido desenvolvidos, devido a suas restrições, que impossibilitam que protocolos e algoritmos tradicionais de comunicação sejam utilizados.

Neste sentido, esta tese apresenta a especificação de um algoritmo para manutenção da conectividade de RSSFs, utilizando Tecnologia Adaptativa, que corresponde a uma contribuição inovadora. A topologia adotada no algoritmo foi em árvore, tendo como raiz o nó sorvedouro principal. A análise do *hop count* mostrou que baseado nesta métrica o algoritmo possui desempenho satisfatório e equivalente a outros protocolos já existentes.

Para finalizar, a contribuição principal desta tese é o desenvolvimento de um

algoritmo para comunicação em RSSFs, por meio da integração das áreas de RSSF e Tecnologia Adaptativa, mais especificamente, Tabela de Decisão Adaptativa e Grafos Direcionados Dinâmicos.

Referências

- AKKAYA, K.; YOUNIS, M. *Ad Hoc Networks*, v. 3, p. 325–349, 2005.
- AKYILDZ, I. F. et al. Wireless sensor networks: a survey. *Computer Networks*, v. 38, p. 393–422, 2002.
- ALMEIDA, A. T. de; GOMES, C. F. S.; GOMES, L. F. *Tomada de decisão gerencial: enfoque multicritério*. São Paulo - SP: Atlas, 2002.
- AMATO, G.; CARUSO, A.; CHESSA, S. Application-driven, energy-efficient communication in wireless sensor networks. *Computer Communications*, v. 32, p. 896–906, 2009.
- ANASTASI, G. et al. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, v. 7, p. 537–568, 2009.
- ARAYA, M. C. G.; CARIGNANO, C.; GOMES, L. F. A. M. *Tomada de decisões em cenários complexos*. São Paulo, SP: Thomson, 2004.
- ARDIZZONE, E. et al. An integrated architecture for surveillance and monitoring in an archaeological site. In: ACM International Workshop on Video Surveillance and Sensor Networks, 3, 2005. *Anais...* [S.l.]: ACM, 2005. p. 79–85.
- AZIZ, O. et al. A pervasive body sensor network for measuring postoperative recovery at home. *Surgical Innovation*, v. 14, n. 2, p. 83–90, Junho. 2007.
- BABBITT, T. A. et al. Self-selecting reliable paths for wireless sensor network routing. *Computer Communications*, v. 31, p. 3799–3809, 2008.
- BAI, H.; ATIQUZZAMAN, M.; LILJA, D. Wireless sensor network for aircraft health monitoring. In: International Conference on Broadband Networks (BROADNETS 2004), 5, San José, Califórnia. *Anais...* [S.l.]: IEEE Computer Society, 2004.
- BALESTRIN, A. Uma análise da contribuição de hebert simon às teorias organizacionais. *Revista Eletrônica de Administração*, v. 8, n. 4, p. 1–17, Julho. 2002.
- BARONTI, P. et al. Wireless sensor networks: A survey on the state of the art and 802.15.4 and zigbee standards. *Computer Communications*, v. 30, n. 7, p. 1655–1695, 2007.
- BASAGNI, S. et al. Controlled sink mobility for prolonging wireless sensor networks lifetime. *Wireless Networks*, v. 14, n. 6, p. 831–858, Dezembro. 2008.
- BINDER, F. V. *Sistemas de Apoio à Decisão*. São Paulo - SP: Érica, 1994.

- BISIANI, R. et al. A logical approach to home healthcare with intelligent sensor-network support. *The Computer Journal*, p. 1–20, Maio. 2009.
- BRAGINSKY, D.; ESTRIN, D. Rumor routing algorithm for sensor networks. In: Workshop on Sensor Networks and Applications, Atlanta, GA. *Anais...* [S.l.]: ACM, 2002. p. 22–31.
- CAMARGO, R.; RAUNHEITTE, L. Usando adaptatividade na identificação de padrões. In: Workshop de Tecnologia Adaptativa, 4, São Paulo - SP. *Anais...* [S.l.]: LTA: São Paulo - SP, 2010.
- CAMILLI, A. et al. From wireless sensors to field mapping: Anatomy of an application for precision agriculture. *Computers and Electronics in Agriculture*, v. 58, p. 25–36, Janeiro. 2007.
- CERTO, S. C. *Modern Management: diversity, quality, ethics and the global environment*. New Jersey: Prentice Hall, 2000.
- CHAER, I. *Um estudo sobre a Teoria da Predição aplicada à análise semântica de Linguagens Naturais*. 72 p. — Dissertação (Mestrado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2010.
- CHAN, E. P.; YANG, Y. Shortest path tree computation in dynamic graphs. *IEEE Transactions on Computers*, v. 58, n. 4, p. 541–557, Abril. 2009.
- CHEN, B. et al. Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *ACM Wireless Networks*, v. 8, n. 5, 2002.
- CHEN, J.; KHER, S.; SOMANI, A. Distributed fault detection of wireless sensor networks. In: International Conference on Mobile Computing and Networking, Los Angeles - EUA. *Anais...* [S.l.]: ACM, 2006. p. 65–72.
- CHOI, J.-Y. et al. An adaptive fault detection scheme for wireless sensor networks. In: WSEAS International Conference on Software engineering, parallel and distributed systems, 8, Winsconsin, EUA. *Anais...* [S.l.]: World Scientific and Engineering Academy and Society (WSEAS), 2009.
- CHOI, S.-H. et al. An implementation of wireless sensor network for security system using bluetooth. *IEEE Transactions on Consumer Electronics*, v. 50, n. 1, p. 236–244, Fevereiro. 2004.
- CHU, M.; HAUSSECKER, H.; ZHAO, F. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *The International Journal of High Performance Computing Applications*, v. 16, n. 3, p. 293–313, 2002.
- CROSSBOW. *Wireless Sensor Network Platforms*. Acessado em Setembro, 2006. Disponível em: <www.xbow.com/Products/wproductsoverview.aspx>.
- CULLER, D.; ESTRIN, D.; SRIVASTAVA, M. Overview of sensor networks. *IEEE Computer Society*, v. 37, n. 8, p. 41–49, Agosto. 2004.
- DEMETRESCU, C.; FINOCCHI, I.; ITALIANO, G. F. Dynamic graph algorithms. In: _____. YELLEN, J.; GROSS, J. L. *Handbook of Graph Theory*. [S.l.: s.n.], 2003.

- DEMETRESCU, C.; ITALIANO, G. F. Dynamic shortest paths and transitive closure: Algorithmic techniques and data structures. *Journal of Discrete Algorithms*, v. 4, n. 3, p. 353–383, 2006.
- DÍAZ, J.; MITSCHKE, D.; PÉREZ-GIMÉNEZ, X. Large connectivity for dynamic random geometric graphs. *IEEE Transactions on Mobile Computing*, v. 8, n. 6, p. 821–835, Junho. 2009.
- DOI, F. *Objetos Adaptativos: Aplicação de Tecnologia Adaptativa à Orientação a Objetos*. 104 p. — Dissertação (Mestrado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2007.
- DRESSLER, F. A study of self-organization mechanisms in ad hoc and sensor networks. *Computer Communications*, v. 31, p. 3018–3029, Fevereiro. 2008.
- FERNÁNDEZ-LUQUE, L. et al. ediab: A system for monitoring, assisting and educating people with diabetes. *Lecture Notes in Computer Science*, v. 4061, p. 1342–1349, Julho. 2006.
- FERREIRA, R. P.; AFFONSO, C.; SASSI, R. J. Roteirização dinâmica de veículos combinada à previsão do comportamento do tráfego urbano utilizando uma rede neuro *Fuzzy*. In: Workshop de Tecnologia Adaptativa, 4, São Paulo - SP. *Anais...* [S.l.]: LTA: São Paulo - SP, 2010.
- FIGUEIREDO, C. M. S.; NAKAMURA, E. F.; LOUREIRO, A. A. Protocolo adaptativo híbrido para disseminação de dados em redes de sensores sem fio auto-organizáveis. In: Simpósio Brasileiro de Redes de Computadores, 22, Gramado - RS. *Anais...* [S.l.]: SBC, 2004. p. 43–56.
- FREITAS, A. V. de. *Considerações sobre o Desenvolvimento de Linguagens Adaptativas de Programação*. 122 p. — Tese (Doutorado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2008.
- FULÖP, J. *Introduction to Decision Making Methods*. [S.l.]: Laboratory of Operations Research and Decision Systems, Computer and Automation Institute, Hungarian Academy of Sciences, 2005.
- GALETTO, M.; PRALIO, B. Optimal sensor positioning for large scale metrology applications. *Precision Engineering*, v. 34, p. 563–577, 2010.
- GANZELI, H. S. et al. Skan – skin scanner: Software para o reconhecimento de câncer de pele utilizando técnicas adaptativas. In: Workshop de Tecnologia Adaptativa, 4, São Paulo - SP. *Anais...* [S.l.]: LTA: São Paulo - SP, 2010.
- GHICA, O. et al. Controlled multi-path routing in sensor networks using bezier curves. *The Computer Journal*, p. 1–25, 2010.
- GILDERSLEEVE, T. R. *Decision tables and their practical application in data processing*. Englewood Cliffs, New Jersey: Prentice Hall, 1970.
- GONDA, L. et al. Aplicação de tecnologia adaptativa em redes de sensores sem fio. In: Workshop de Tecnologia Adaptativa, 3, São Paulo - SP. *Anais...* [S.l.]: LTA: São Paulo - SP, 2009.

- GONDA, L.; CUGNASCA, C. E.; NETO, J. J. Uso de tabelas de decisão adaptativas em redes de sensores sem fio. In: Workshop de Tecnologia Adaptativa, 4, São Paulo - SP. **Anais...** [S.l.]: LTA: São Paulo - SP, 2010.
- HE, Y.; LEE, I.; GUAN, L. Distributed algorithms for network lifetime maximization in wireless visual sensor networks. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 19, n. 5, p. 704–718, Maio. 2009.
- HEINZELMAN, W. R.; CHANDRAKASAN, A.; BALAKRISHNAN, H. Energy-efficient communication protocol for wireless microsensor networks. In: Hawaii International Conference on Systems Services, 1, Maui, Hawaii. **Anais...** [S.l.]: IEEE, 2000.
- HEINZELMAN, W. R.; KULIK, J.; BALAKRISHNAN, H. Adaptive protocols for information dissemination in wireless sensor networks. In: International Conference on Mobile Computing and Networking, 5, Seattle, Washington. **Anais...** [S.l.]: ACM, 1999. p. 174–185.
- HUGHES, M. L.; SHANK, R. M.; STEIN, E. S. **Decision Tables**. Wayne, Pennsylvania: Midi Publications, Management Development Institute, Divisions of Information, Inc., 1968.
- INTANAGONWIWAT, C.; GOVINDAN, R.; ESTRIN, D. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In: International Conference on Mobile Computing and Networking, 6, Boston, Massachusetts. **Anais...** [S.l.]: ACM, 2000. p. 56–67.
- JIANG, P. A new method for node fault detection in wireless sensor networks. **Sensors**, n. 9, p. 1282–1294, Fevereiro. 2009.
- JUNIOR, A. A. de C. **Aspectos de Projeto e Implementação de Linguagens para Codificação de Programas Adaptativos**. 91 p. — Tese (Doutorado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2009.
- KEENEY, R. L.; RAIFFA, H. **Decisions with multiple objectives: preferences and value tradeoffs**. New York: Willey, 1976.
- KOHVAKKA, M. et al. Energy-efficient neighbor discovery protocol for mobile wireless sensor networks. **Ad Hoc Networks**, v. 7, p. 24–41, 2009.
- KOMORI, F. S. et al. Xo trânsito: Utilização de algoritmos adaptativos no planejamento automático de rotas de trânsito. In: Workshop de Tecnologia Adaptativa, 3, São Paulo - SP. **Anais...** [S.l.]: LTA: São Paulo - SP, 2009.
- LANDSIEDEL, O.; WEHRLE, K. Aeon: Accurate prediction of power consumption in sensor networks. In: Second IEEE Workshop on Embedded Networked Sensors (EmNetS-II), 2, 2005. **Anais...** [S.l.]: IEEE Computer Society, 2005. p. 37–44.
- LEE, M.-H.; CHOI, Y.-H. Fault detection of wireless sensor networks. **Computer Communications**, v. 31, n. 5, p. 3469–3475, Setembro. 2008.
- LEWIS, H.; PAPADIMITRIOU, C. H. **Elements of the theory of computation**. Upper Saddle River: Prentice Hall, 1998.

LI, E. L.; HALPERN, J. Y. Minimum-energy mobile wireless networks revisited. In: IEEE International Conference on Communications (ICC 2001), Helsinki, Finlândia. **Anais...** [S.l.]: IEEE, 2001.

LINDSEY, S.; RAGHAVENDRA, C.; SIVALINGAM, K. Data gathering in sensor networks using the energy*delay metric. In: IPDPS Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, São Francisco, Califórnia. **Anais...** [S.l.]: IEEE, 2001. p. 924–935.

LINDSEY, S.; RAGHAVENDRA, C. S. Pegasis: Power-efficient gathering in sensor information systems stephanie lindsey cauligi s. raghavendra. In: IEEE Aerospace Conference, Big Sky, Montana. **Anais...** [S.l.]: IEEE, 2002.

LOUREIRO, A. A. et al. Redes de sensores sem fio. In: Simpósio Brasileiro de Redes de Computadores, 21, Natal - RN. **Anais...** [S.l.]: SBC, 2003. p. 179–226.

LUNG, C.-H.; ZHOU, C. Using hierarchical agglomerative clustering in wireless sensor networks: An energy-efficient and flexible approach. **Ad Hoc Networks**, v. 8, p. 328–344, 2010.

LUO, H. et al. Ttdd: Two-tier data dissemination in large-scale wireless sensor networks. **Wireless Networks**, v. 11, p. 161–175, Fevereiro. 2005.

MANJESHWAR, A.; AGRAWAL, D. P. Aptein: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. In: International Parallel and Distributed Processing Symposium, Fort Lauderdale, Florida. **Anais...** [S.l.]: IEEE, 2002.

MANNASIM. *MannaSim: Wireless Sensor Networks Simulation*. Acessado em Julho, 2008. Disponível em: <<http://www.mannasim.dcc.ufmg.br>>.

MAO, S.; HOU, T. Beamstar: An edge-based approach to routing in wireless sensor networks. **IEEE Transactions on Mobile Computing**, v. 6, n. 11, p. 1284–1296, Novembro. 2007.

MARGI, C. B. **Energy Consumption Trade-Offs in Power Constrained Networks**. 185 p. — Tese (Doutorado) - University of California, Santa Cruz, 2006.

MARKHAM, A.; TRIGONI, N. The automatic evolution of distributed controllers to configure sensor network operation. **The Computer Journal**, p. 1–18, 2010.

MATSUNO, I. P. **Um Estudo do Processo de Inferência de Gramáticas Regulares e Livres de Contexto Baseados em Modelos Adaptativos**. 107 p. — Dissertação (Mestrado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2006.

MIN, X. et al. Energy efficient clustering algorithm for maximizing life time of wireless sensor networks. **International Journal of Eletronics and Communications**, v. 64, p. 289–298, 2010.

MIT. *10 Emerging Technologies That Will Change the World*. Acessado em Agosto, 2007. Disponível em: <www.mtr.net/article-5377.html>.

MORAES, M. de. *Alguns aspectos de tratamento de dependências de contexto em linguagem natural empregando tecnologia adaptativa*. 150 p. — Tese (Doutorado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2006.

MOUSTAPHA, A. I.; SELMIC, R. R. Wireless sensor network modeling using modified recurrent neural networks: Application to fault detection. *IEEE Transactions on Instrumentation and Measurement*, v. 57, n. 5, p. 981–988, Maio. 2008.

NAKAMURA, E. F. *Fusão de Dados em Redes de Sensores Sem Fio*. 174 p. — Tese (Doutorado) - Universidade Federal de Minas Gerais, Belo Horizonte, 2007.

NASSI, I.; SHNEIDERMAN, B. Flowchart techniques for structured programming. *ACM SIGPLAN Notices*, v. 8, n. 8, p. 12 – 26, 1973.

NETO, J. J. Adaptive rule-driven devices - general formulation and case study. *Lecture Notes in Computer Science*, v. 2494, p. 234–250, Julho. 2001.

NETO, J. J. Um levantamento da evolução da adaptatividade e da tecnologia adaptativa. *IEEE Latin America*, v. 5, n. 7, p. 496–505, Novembro. 2007.

NS-2. *The Network Simulator NS-2*. Acessado em outubro, 2008. Disponível em: <<http://www.isi.edu/nsnam/ns>>.

O'BRIEN, J. A. *Sistemas de Informação e as Decisões Gerenciais na Era da Internet*. 2. ed. São Paulo, SP: Saraiva, 2004.

PEDRUZZI, P. R. N.; JR., R. A. R.; NETO, J. J. Uma proposta do uso de adaptatividade na busca de padrões musicais. In: Workshop de Tecnologia Adaptativa, 3, São Paulo - SP. *Anais...* [S.l.]: LTA: São Paulo - SP, 2009.

PELEGRINI Éder J. *Códigos Adaptativos e Linguagem para Programação Adaptativa: Conceitos e Tecnologia*. 143 p. — Dissertação (Mestrado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2009.

PIDD, M. *Modelagem Empresarial: Ferramentas para Tomada de Decisão*. Porto Alegre, RS: Artes Médicas, 1998.

PISTORI, H. *Tecnologia Adaptativa em Engenharia de Computação: estado da arte e aplicações*. 174 p. — Tese (Doutorado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2003.

PISTORI, H.; SOUZA, K. P. de. Tecnologia adaptativa aplicada à biotecnologia: Estudos de caso e oportunidades. In: Workshop de Tecnologia Adaptativa, 4, São Paulo - SP. *Anais...* [S.l.]: LTA: São Paulo - SP, 2010.

POMEROL, J.-C.; ADAM, F. On the legacy of hebert simon and his contribution to decision-making support systems and artificial intelligence. *Intelligent Decision-making Support Systems, Springer London*, p. 25–43, Março. 2007.

POTTER, J. et al. Wireless sensor networks for ecology. *BioScience*, v. 55, n. 7, p. 561–572, Julho. 2005.

RAGHUNATHAN, V. et al. Energy-aware wireless microsensor networks. ***IEEE Signal Processing Magazine***, p. 40–50, 2002.

RAGSDALE, C. T. ***Spreadsheet Modeling and Decision Analysis***. 3. ed. Petrópolis, RJ: South-Western College Pub., 2001. 820 p.

RAIFFA, H. ***Teoria da Decisão: aulas introdutórias sobre escolhas em condições de incerteza***. Petrópolis, RJ: Vozes, 1977.

RAMASUBRAMANIAN, V.; HAAS, Z. J.; SIRER, E. G. Sharp: A hybrid adaptive routing protocol for mobile ad hoc networks. In: ACM international Symposium on Mobile Ad Hoc Networking and Computing, 4, Annapolis, Maryland, E.U.A.. ***Anais...*** [S.l.]: ACM: Nova York, E.U.A., 2003. p. 303–314.

RAPPORT, T. ***Wireless Communications: Principles and Practice***. 2. ed. [S.l.]: Prentice Hall, 2002. 736 p. ISBN 0130422320.

RICCHETTI, P. M. ***Geração de Analisadores Sintáticos Baseados em Transdutores a partir de Especificações Gramaticais***. 147 p. — Dissertação (Mestrado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2005.

RODOPLU, V.; MENG, T. H. Minimum energy mobile wireless networks. ***IEEE Journal on Selected Areas in Communications***, v. 17, n. 8, p. 1333 – 1344, 1999.

ROY, B. ***Multicriteria Methodology For Decision Aiding***. [S.l.]: Kluwer Academic Publishers, 1996.

SAATI, T. L. ***Método de Análise Hierárquica***. Rio de Janeiro, RJ: McGraw - Hill, Makron, 1991.

SAATI, T. L. ***Fundamentals of decision making and priority theory with the Analytic Hierarchy Process***. Pittsburgh: RWS Publications, 1994.

SANTI, P. Topology control in wireless ad hoc and sensor networks. ***ACM Computing Surveys***, v. 37, n. 2, p. 164–194, Julho. 2005.

SHEBLI, F. et al. A new optimization approach for energy consumption within wireless sensor networks. In: Advanced International Conference on Telecommunications (AICT 2007), 3, Mauritius. ***Anais...*** [S.l.]: IEEE Computer Society, 2007.

SHIBATA, D. P. ***Tradução Grafema-Fonema para a Língua Portuguesa Baseada em Autômatos Adaptativos***. 104 p. — Dissertação (Mestrado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2008.

SOUSA, M. A. de Abreu de. ***Mapeamento de Ambientes Desconhecidos por Robôs Móveis Utilizando Autômatos Adaptativos***. 157 p. — Dissertação (Mestrado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2006.

SUBRAMANIAN, L. et al. An architecture for building self-configurable systems. In: Workshop on Mobile Ad Hoc Networking and Computing, Boston, Massachusetts. ***Anais...*** [S.l.]: IEEE/ACM, 2000. p. 63–73.

SUN, B. et al. Algorithms for balancing energy consumption in wireless sensor networks. In: ACM International Workshop on Foundations of Wireless Ad Hoc and Sensor Networking and Computing, 1, Honk Kong, China. **Anais...** [S.l.]: ACM, 2008.

SUN, L.-M.; YAN, T.-X.; BI, Y.-Z. Remuda: A practical topology control and data forwarding mechanism for wireless sensor networks. **ACTA AUTOMATICA SINICA**, v. 32, n. 6, p. 867–874, Março. 2006.

TCHEMRA, A. H. **Tabela de Decisão Adaptativa na Tomada de Decisão Multicritério**. 171 p. — Tese (Doutorado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2009.

TCHEMRA, A. H.; CAMARGO, R. Descoberta de padrões em bases de dados utilizando técnicas adaptativas. In: Workshop de Tecnologia Adaptativa, 3, São Paulo - SP. **Anais...** [S.l.]: LTA: São Paulo - SP, 2009.

THOMAZ, J. P. da C. F.

Concepção de um modelo multicritério de apoio à decisão — Dissertação (Mestrado) - Universidade Lusíada, Lisboa, 2000.

TILAK, S.; ABU-GHAZALEH, N. B.; HEINZELMAN, W. A taxonomy of wireless micro-sensor network models. **Mobile Computing and Communications Review**, v. 6, n. 2, p. 28–36, 2002.

TINYOS. *TinyOS Operating System*. Acessado em Setembro, 2006. Disponível em: <www.tinyos.net>.

TULONE, D.; MADDEN, S. An energy-efficient querying framework in sensor networks for detecting node similarities. In: International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems, 9, Terromolinos, Espanha. **Anais...** [S.l.]: ACM, 2006. p. 191–300.

WANG, S.-C.; KUO, S.-Y. Communication strategies for heartbeat-style failure detectors in wireless ad hoc networks. In: International Conference on Dependable Systems and Networks (DSN'03), 3, São Francisco, EUA. **Anais...** [S.l.]: IEEE Computer Society, 2003. p. 361–370.

WARK, T. et al. Transforming agriculture through pervasive wireless sensor networks. **Pervasive Computing**, p. 50–57, Abril-Junho. 2007.

XU, Y.; HEIDEMANN, J.; ESTRIN, D. Geography-informed energy conservation for ad hoc routing. In: ACM/IEEE International Conference on Mobile Computing and Networking, 7, Roma, Itália. **Anais...** [S.l.]: ACM/IEEE, 2001. p. 70–84.

YU, Y.; GOVINDAN, R.; ESTRIN, D. **Geographical and Energy Aware Routing: a recursive data dissemination protocol for wireless sensor networks**. Maio. 2001. (UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023).

ZERGER, A. et al. Environmental sensornetworksforvegetation,animalandsoilsciences (article in press). **International Journal of Applied Earth Observation and Geoinformation**, Maio. 2010.

ZHOU, C.; KRISHNAMACHARI, B. Localized topology generation mechanisms for self-configuring sensor networks. In: IEEE Global Telecommunications Conference (GLOBECOM'03), 1, São Francisco, EUA . ***Anais...*** [S.l.]: IEEE, 2003. p. 1269–1273.

Apêndice A – Pseudo-Código

Este apêndice apresenta o pseudo-código das operações de envio e tomada de decisão do algoritmo ADAP_RSSF proposto.

Procedimento: Receive

Entrada: um pacote *packet*

Saída: um evento a ser realizado e 0 ou 1 regras adicionadas à Tabela Adaptativa.

```

Se my_type = SINK Então
  Se packet.Nodetype == COMMON Então
    // Se for uma mensagem de alerta
    Se packet.type == alerta Então
      Se packet.criteria == INSERT Então
        send (newpacket, local,
              packet.SourceAddress, -1)
      Senão Se packet.criteria == DELETE Então
        send (newpacket, local,
              packet.SourceAddress, -1)
      Senão Se packet.criteria == FAIL Então
        send (newpacket,
              local, packet.SourceAddress, -1)
      Senão
        Adaptive_table (num_regras1,
                        num_criterios1,
                        packet.criteria)
    // Se for uma mensagem de dados
    Senão Se packet.type == DATA Então
      return
  Fim Se
Senão Se packet.Nodetype == INTERMEDIATE ou
packet.Nodetype == MIDDLE_SINK Então

```

```

Se packet.type == alerta Então
    Se packet.criteria == INSERT Então
        send (newpacket, local,
              packet.SourceAddress, -1)
    Senão Se packet.criteria == TURNOFF Então
        send (newpacket, global,
              packet.SourceAddress, -1)
    Senão Se packet.criteria == LOW_ENERGY Então
        send (newpacket, global, BROADCAST, -1)
    Senão
        Adaptive_table (num_regras2,
                        num_criterios2,
                        packet.criteria)

    Fim Se
Senão Se packet.type == DATA Então
    return;
Fim Se
Fim Se
Senão Se my_type == MIDDLE_SINK Então
    // Se for uma mensagem de alerta
    Se packet.type == alerta Então
        Se packet.criteria == INSERT Então
            send (newpacket, local,
                  packet.SourceAddress, -1)
        Senão Se packet.criteria == DELETE Então
            send (newpacket, local,
                  packet.SourceAddress, -1)
        Senão Se packet.criteria == FAIL Então
            send (newpacket,
                  local, packet.SourceAddress, -1)
        Senão
            Adaptive_table (num_regras_local,
                            num_criterios_local,
                            packet.criteria)

    // Se for uma mensagem de dados
    Senão Se packet.type == DATA Então
        return
Fim Se

```

```

Senão Se my_type == INTERMEDIATE e
    (packet.DestAddress == my_address
    ou packet.DestAddress == BROADCAST) Então
Se packet.type == data Então
    // Propaga o pacote
    send(packet, packet.type, my_dest.address, -1)
Senão Se packet.type == inicio ou
    packet.type == global Então
    // Configura o novo pai
    my_dest.address = packet.SourceAddress
    // Propaga esta informação via broadcast
    send(packet, packet.type, BROADCAST, -1)
Senão Se packet.type == local Então
    // Verifica se o Nó que envio é o que ele
    // está ligado e propaga esta informação
    Se MIDDLE_SINK == my_sink ou
        MIDDLE_SINK == SINK Então
        send(packet, packet.type,
            packet.DestAddress, -1);
        // busca um novo pai
        send(newpacket, busca, BROADCAST, -1);
Senão Se packet.type == busca Então
    // Envia um pacote se candidatando a pai
    send (newpacket, RESP_BUSCA,
        packet.SourceAddress, -1);
Senão Se packet = alerta Então
    // Propaga esta informação para seu pai
    send (packet, alerta, my_address, packet.criteria)
Fim Se
Senão Se my_type == COMMOM e
    (packet.DestAddress == my_address
    ou packet.DestAddress == BROADCAST) Então
Se packet.type == inicio ou packet.type == global Então
    // Configura um novo pai
    my_dest.address = packet.SourceAddress
    // Propaga esta informação
    send(packet, packet.type, BROADCAST, -1)
Senão Se packet.type == RESP_BUSCA Então

```

```

        // Se o Nó que enviou um pacote
        // é diferente do meu atual pai
        Se packet.SourceAddress <> my_dest.address Então
            // Configura um novo pai
            my_dest.address = packet.SourceAddress
            drop(packet)
        Fim Se
    Senão Se packet.type == local Então
        // Envia um pacote procurando um novo pai
        send (newpacket, busca, BROADCAST, -1)
    Fim Se
Fim Se
Fim Procedimento

```

Procedimento: send

Entrada: um pacote *packet*, o tipo de pacote e o endereço de destino, o critério utilizado

Saída: um pacote a ser enviado

```

    global.num_seq = global.num_seq + 1
    packet.type = type
    packet.DestAddress = dest
    packet.SourceAddress = my_address
    packet.criteria = criteria
    packet.nodetype = my_type
    packet.NumSeq = global.num_seq
    send_packet(packet)
Fim Procedimento

```

Procedimento: Adaptive_Table

Entrada: o número de critérios, o número de regras, o critério a ser buscado

Saída: Uma ou zero regra adicionada à Tabela de Decisão e uma ação de reconstrução

```

add = 1
Para i de 1 ate num_regras Faça
    Se table[criterio][i] == "S" Então

```

```

        // Encontrou a regra e não precisa adicionar
        add = 0
        // Executa a ação correspondente
send(newpacket,
    Fim se
Fim para
Se add == 1 Então
    // Aumenta criterios e regras
    num_regras = num_regras + 1
    num_criterios = num_criterios + 1
    // Configura uma nova regra
    Para j = 1 até num_criterio - 1 Faça
        table[j][num_regras] = "N"
    Fim Para
    table[num_criterios][num_regras] = "S"
    // Define uma ação a ser executada
    Se my_type == SINK Então
        packet.type := global
    Senão
        packet.type := local

    Fim Se
Fim Procedimento

```