Instituto de Matemática / Núcleo de Computação Eletrônica
Universidade Federal do Rio de Janeiro

Bruno Ávila Galvão

Um Protocolo Tolerante a Falhas para Disseminação de Dados em Redes de Sensores Sem Fio

Rio de Janeiro
2005

Bruno Ávila Galvão

Um Protocolo Tolerante a Falhas para Disseminação de Dados em Redes de Sensores Sem Fio

Dissertação submetida ao corpo docente do Núcleo de Computação Eletrônica / Instituto de Matemática da Universidade Federal do Rio de Janeiro – UFRJ, como parte dos requisitos necessários à obtenção do grau de Mestre em Ciências em Informática.

Orientador: Prof[a] Luci Pirmez

Rio de Janeiro
2005

G182 Galvão, Bruno Ávila.
Um Protocolo tolerante a falhas para disseminação de dados em redes de sensores sem fio / Bruno Ávila Galvão. – Rio de Janeiro, 2005.
106 f.: il.

Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro, Instituto de Matemática, Núcleo de Computação Eletrônica, 2005.

Orientadora: Luci Pirmez

1. Redes de Sensores Sem Fio – Teses. 2.Disseminação de Dados – Teses. 3. Tolerância a Falhas – Teses. I. Luci Pirmez (Orient.). II. Universidade Federal do Rio de Janeiro. Instituto de Matemática. Núcleo de Computação Eletrônica. III. Título.

CDD

Bruno Ávila Galvão

Um Protocolo Tolerante a Falhas para Disseminação de Dados em Redes de Sensores Sem Fio

Dissertação submetida ao corpo docente do Núcleo de Computação Eletrônica / Instituto de Matemática da Universidade Federal do Rio de Janeiro – UFRJ, como parte dos requisitos necessários à obtenção do grau de Mestre em Ciências em Informática.

Aprovada em

_____

Prof[a]. Luci Pirmez - Orientador
D.Sc., COPPE/UFRJ, Brasil

_____

Prof. Paulo Henrique Aguiar Rodrigues
Ph.D, UCLA , Estados Unidos

_____

Prof[a]. Flávia Coimbra Delicato
D.Sc., COPPE/UFRJ, Brasil

_____

Prof. Luís Henrique Maciel Kosmalski Costa
Dr., UPMC, França

RESUMO

GALVÃO, Bruno Ávila. Um Protocolo Tolerante a Falhas para Disseminação de Dados em Redes de Sensores Sem Fio**.** Rio de Janeiro, 2005. Dissertação (Mestrado em Informática) - Instituto de Matemática/Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2005.

Esta dissertação aborda a questão da tolerância a falhas em protocolos hierárquicos e baseados em *clusters* para Redes de Sensores sem Fio. Nesse sentido, é proposto um protocolo hierárquico de disseminação de dados em Redes de Sensores sem fio, assim como um mecanismo para torná-lo robusto em relação a falhas. Para tal, é explorada a redundância de nós na rede para eleger um nó reserva, ativado caso o *cluster-head* original falhe. A adoção de um nó reserva para solucionar problemas de falhas em redes não é uma novidade. Entretanto, o critério de seleção desse nó reserva proposto neste trabalho é inovador. Além disso, problemas relativos a perdas de pacotes provocadas por limitações físicas dos nós sensores são expostos e são propostas soluções para se estabelecer a sincronização entre os nós da rede. As propostas são avaliadas através de simulação e os resultados mostram a eficiência do protocolo proposto.

ABSTRACT

GALVÃO, Bruno Ávila. Um Protocolo Tolerante a Falhas para Disseminação de Dados em Redes de Sensores Sem Fio**.** Rio de Janeiro, 2005. Dissertação (Mestrado em Informática)- Instituto de Matemática/Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2005.

This work addresses the fault tolerance issue in hierarchical, cluster-based protocols for wireless sensor networks. A data dissemination protocol for wireless sensor networks, using a hierarchic topology, is proposed as well as a mechanism to provide fault tolerance. This is accomplished by exploiting the redundancy in nodes to elect a spare leader, activated if the originally elected one is faulty. The adoption of a spare leader to provide fault tolerance is not an inovative mechanism. However, the criteria used to elect the spare leader are inovative. Additionally, issues related to packet loss caused by physical limitation of sensor nodes are studied and solutions are proposed. Simulations are performed to evaluate the proposals. Results show that hierarchical protocols enhanced with the proposed mechanisms achieve larger network robustness to node failures and larger average sensing coverage in contrast with protocols without such mechanisms.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

RSSF	Rede de Sensores Sem Fio
LEACH	*Low Energy Adaptive Clustering Hierarchy*
DD	*Directed Diffusion*
ICA	*Inter Cluster Routing Algorithm*
TEEN	*Threshold sensitive Energy Efficient sensor Network*
PEGASIS	*Power Efficient Gathering in Sensor Information Systems*
NS-2	*Network Simulator 2*
CH	*Cluster Head*
TDMA	*Time Division Multiple Access*
IP	*Internet Protocol*
AM	*Amplitude Modulation*
FM	*Frequency Modulation*
PM	*Phase Modulation*
ISM	*Industry, Science, Medicine*
FDMA	*Frequency Division Multiple Access*
CDMA	*Code Division Multiple Access*
CSMA	*Carrier Sense Multiple Access*
CSMA/CA	*Carrier Sense Multiple Access / Collision Avoidance*
WLAN	*Wireless Local Area Network*
ESS	*Extended Service Set*
iBSS	*independent Basic Service Set*
BSA	*Basic Service Area*
AP	*Access Point*
STA	*Station*
SMACS	*Self Organizing MAC for Sensor Networks*
CRC	*Cyclic Redundancy Check*
ARQ	*Automatic Repeat Request*
FEC	*Forward Error Correction*
TTDD	*Two-Tier Data Dissemination*
SEAD	*Scalable Energy-efficient Asynchronous Dissemination*
Ht	*Hard Threshold*
St	*Soft Threshold*
SOP	*Self Organizing Protocol*
STALK	*Stabilizing Tracking viA Layered linKs*
MAC	*Multiple Access Channel*
QoS	*Quality of Service*
TX	Transmissão
RX	Recepção
RX CH	Recepção de *Cluster-head*

SUMÁRIO

Capítulo 1 Introdução

Recentes avanços tecnológicos têm viabilizado a integração de dispositivos de baixo custo e tamanho reduzido com capacidade de processamento, de comunicação e de sensoriamento em um mesmo circuito integrado. Esses dispositivos, denominados de nós sensores, são dotados de recursos de comunicação de curto alcance e formam um tipo de rede *ad hoc* (sem infra-estrutura) conhecido como Redes de Sensores sem Fio (RSSF).

As RSSFs são geralmente compostas por um grande número de nós sensores de baixo custo e que podem ser distribuídos aleatoriamente na área de monitoração. Os nós sensores, quando usados de maneira cooperativa, podem coletar com precisão informações sobre o ambiente (ex: temperatura) onde se encontram instalados. Além disso, as informações coletadas pelos nós sensores têm como destino final um ou mais nós conhecidos como sorvedouros, através dos quais os dados coletados são disponibilizados para os usuários para posterior processamento. Entretanto, as capacidades de processamento, memória e energia desses nós sensores são limitadas. Logo, é crucial para as redes de sensores sem fio a utilização eficiente dos recursos de energia para maximizar o seu tempo de vida útil e a robustez do sistema [2]. Portanto, os protocolos de disseminação de dados devem ser concebidos de forma a serem eficientes em termos de energia bem como tolerantes a falhas.

As redes de sensores sem fio, por serem esquemas eminentemente distribuídos de monitoramento, apresentam, conforme descrito em [1], diversas vantagens em relação às redes de sensores utilizadas no passado (esquema infra-estruturado), que consistiam em um pequeno número de nós conectados através de cabos a uma central de processamento. Uma vantagem das RSSFs, por exemplo, é quando a localidade exata de um determinado fenômeno é desconhecida. Nesse caso, o sensoriamento distribuído, além de ser mais flexível, leva a uma maior precisão em comparação ao que poderia ser obtido com o esquema infra-estruturado antigo devido ao reduzido número de nós. Outra vantagem das RSSFs é que, em geral, múltiplos sensores são necessários para vencer possíveis obstáculos de comunicação existentes no ambiente que se deseja monitorar. A obstrução física, por exemplo, entre dois nós sensores, inviabilizando a comunicação direta entre eles, pode ser contornada através de um terceiro nó sensor que possua comunicação direta com os outros dois nós.

Diferentemente das redes tradicionais, as RSSFs apresentam peculiaridades e limitações que impõem severas restrições no projeto dos protocolos de camada de enlace e de rede. Dentre essas restrições, pode-se destacar a quantidade de energia disponível, a capacidade de

processamento, de memória e, dependendo da aplicação, a imprevisibilidade do posicionamento inicial dos nós. A energia disponível nos nós sensores é limitada devido às suas dimensões e peso reduzidos, o que impõe o uso de baterias de tamanhos diminutos. Outro fator relevante é que, em aplicações onde a área de interesse está localizada em uma região de difícil acesso, a recarga das baterias pode ser inviável técnica ou economicamente. Logo, os protocolos devem ser concebidos considerando a máxima redução de processamento e uso de memória. Além disso, os recursos de comunicação devem ser usados de forma eficiente, pois a maior fonte de consumo de energia ocorre na transmissão e recepção dos sinais de rádio.

Quanto à capacidade em termos de processamento e memória, os nós de uma RSSF são extremamente limitados, possuindo, em geral, uma unidade de processamento capaz de executar um conjunto bastante reduzido de instruções, e dispondo de pouca memória (armazenamento primário). Desse modo, um protocolo para RSSFs deve exigir dos nós sensores o mínimo possível de processamento e armazenamento, procurando sempre enviar os dados coletados, relativos a um determinado fenômeno, imediatamente após a coleta.

Quanto à imprevisibilidade do posicionamento dos nós, ocorre que, na maioria das aplicações, a área de interesse, por estar localizada em regiões remotas, obriga que a instalação dos sensores seja feita por intermédio da dispersão aleatória dos sensores. Isso resulta em uma distribuição aleatória dos nós na área de interesse, podendo parte dessa área ficar descoberta (sem sensores monitorando) enquanto outra parte ficar com excesso de sensores. Uma das formas para minimizar tal problema é o uso de um grande número de nós, ou seja, adoção de uma rede com alta densidade de nós. Entretanto, com uma grande quantidade de nós monitorando uma mesma área, produzindo informações redundantes, cria-se uma super utilização do meio de comunicação, normalmente compartilhado, o que leva a um consumo excessivo e desnecessário de energia devido às retransmissões. Então, os protocolos para RSSFs devem considerar essas características visando a economia de energia.

Além das restrições mencionadas, o desempenho dos protocolos para RSSFs é dependente das características das aplicações alvo. Basicamente, essas aplicações se dividem em dois tipos principais: orientadas a eventos e periódicas. O primeiro tipo consiste na aquisição de dados a partir da ocorrência de eventos específicos. Tais eventos de interesse podem ser descritos através de solicitações de dados (interesse) submetidas pelas aplicações através do nó sorvedouro. O interesse define, por exemplo, o tipo de fenômeno que a aplicação deseja ser notificada sobre sua ocorrência. Além disso, em determinado momento, somente os dados de uma sub-região podem ser necessários, tornando vital a desativação de

parte dos nós para economia de energia. Com isso, os protocolos devem prever mecanismos para deixar alguns dos nós em modo de baixo consumo de energia, quando não localizados na região de interesse.

No segundo tipo, periódico, os nós são configurados de modo a ficar periodicamente coletando e enviando dados em direção ao nó sorvedouro, não estando tais envios condicionados à ocorrência de algum evento. Tal periodicidade, dependendo da aplicação, pode levar os nós a uma disputa intensa pelo meio de comunicação, visto que podem ter suas transmissões ocorrendo ao mesmo tempo, repetidamente. Com isso, cria-se a necessidade de se ter protocolos na camada de rede e enlace que disciplinem as transmissões efetuadas pelos nós, visando sempre não desperdiçar energia com transmissões que potencialmente falhariam ou consumiriam energia excessivamente.

Em ambos os tipos de aplicação, tem-se um esquema onde os nós sensores coletam dados, enviando-os para um ou mais nós sorvedouros. Ao longo do percurso entre esses nós e os sorvedouros, os dados passam possivelmente por vários nós de encaminhamento, onde podem sofrer agregações para a eliminação de redundâncias ou de dados espúrios, visando minimizar o consumo de energia com transmissões de dados duplicados.

Dentre exemplos comuns de aplicações para RSSFs, pode-se destacar a monitoração de grandezas como temperatura e umidade [46], monitoração de *habitats* [42,43,44] (p. ex., acompanhamento do comportamento das populações da fauna e da flora de uma região), vigilância militar [45], etc. De fato, devido à natureza pervasiva dos nós sensores, as RSSFs têm potencial para revolucionar a maneira como os sistemas de monitoração de fenômenos são construídos.

Quanto ao método usado para fazer com que os nós sensores transmitam seus dados para o sorvedouro, podem-se destacar duas principais abordagens. Na primeira, simplesmente são feitas transmissões diretas a partir do nó até o distante sorvedouro, envolvendo um consumo excessivo de energia, visto que a quantidade de energia necessária para uma transmissão é dada em função da distância até o destino, conforme mostrado em [48]. Além disso, tais transmissões diretas são normalmente impraticáveis, visto que o alcance do rádio dos nós tende a ser menor que a distância entre o nó sensor e o sorvedouro.

A segunda abordagem é caracterizada pelo método conhecido como *multi-hop*, onde é explorada a comunicação de curto alcance através de múltiplos saltos em direção ao nó sorvedouro. Exemplos de protocolos que adotam a comunicação *multi-hop* podem ser encontrados em [26,29,57,58]. Além disso, as vantagens desse método sobre a comunicação direta são detalhadamente descritas em [9].

Quanto à topologia, pode-se dizer que existem dois tipos de protocolos de disseminação de dados para RSSFs: protocolos de topologia hierárquica e protocolos de topologia plana. Nos protocolos hierárquicos de dois níveis, por exemplo, a comunicação entre os nós segue as regras de uma hierarquia onde os nós da rede são divididos em grupos (*clusters*). Cada grupo é formado por um conjunto de nós (segundo nível da hierarquia) e um nó líder chamado de *cluster-head* (primeiro nível da hierarquia). Assim, os nós de um determinado grupo só podem se comunicar com o *cluster-head* do seu respectivo grupo. Esse, por sua vez, pode se comunicar com os outros *cluster-heads*. Já nos protocolos de topologia plana, não existe tal hierarquização, sendo permitido para qualquer nó, comunicar-se com qualquer outro.

1.1 Objetivos

Esta dissertação propõe um protocolo hierárquico de disseminação de dados para redes de sensores sem fio diferenciado em relação aos existentes na literatura. Tal protocolo incorpora um mecanismo de recuperação de falhas, bem como um esquema híbrido de envio de dados, orientado a eventos ou periódico.

O protocolo proposto foi desenvolvido a partir da proposta do protocolo LEACH (*Low Energy Adaptive Clustering Hierarchy*) [27]. Foram realizadas modificações e extensões no protocolo LEACH com o intuito de principalmente ***eliminar o requisito de que os nós sensores da rede devem obrigatoriamente possuir comunicação direta com o nó sorvedouro***. Em contrapartida, o protocolo proposto impõe um número mínimo de *clusters* para garantir a alcançabilidade entre *clusters* vizinhos. A grande vantagem da eliminação da restrição relativa à comunicação direta entre os *cluster-heads* e o sorvedouro é a desvinculação do tamanho da área de monitoramento (de interesse) com o raio de alcance dos nós, ou seja, com a nova abordagem é possível monitorar áreas de interesse com diâmetros maiores que o raio de alcance de cada nó. Outra vantagem é a economia de energia obtida devido à redução da potência necessária na transmissão dos sinais de rádio além da diminuição do custo dos nós.

No LEACH, a recuperação de falhas em nós está inerentemente incorporada ao próprio protocolo, já que o nó defeituoso de um *cluster* é automaticamente eliminado da rede após o processo periódico de reorganização de topologia. Esse esquema de recuperação de falhas do LEACH pode levar alguma área da rede à falta de cobertura por um período equivalente ao intervalo entre fases de reorganização da rede. Tal indisponibilidade surge quando uma falha ocorre em um nó líder de *cluster* (*cluster-head)*, tornando sua área descoberta, ou seja, sem coleta e monitoração até a próxima fase de reorganização.

A estratégia de recuperação de falhas introduzida no protocolo proposto se baseia na definição de um nó *cluster-head* reserva e atua logo após a falha, que ocorre entre os momentos de reorganização da topologia. Tal estratégia visa diminuir a latência de recuperação se comparada com a do LEACH e, conseqüentemente, reduzir o tempo de indisponibilidade. A contribuição relevante da estratégia de recuperação de falhas proposta está na concepção do método para eleição do *cluster-head* reserva. Tal método resultou em uma maior uniformidade do consumo de energia dos nós da rede.

O LEACH, ao adotar somente a estratégia de coleta periódica como método de disseminação de dados, limita seu uso a um menor número de aplicações. Em contrapartida, o protocolo proposto atende tanto a aplicações orientadas a eventos quanto a periódicas (esquema híbrido), aumentando a gama de possíveis aplicações. A amplitude na gama de aplicações é possível graças a um maior controle sobre a função de coleta de informações, permitindo que o protocolo proposto, assim como o protocolo Directed Diffusion [26], possa também ser usado para aplicações orientadas a interesse.

Vários trabalhos têm sido propostos com o intuito de criar protocolos hierárquicos mais eficientes que o LEACH em termos de energia. O protocolo ICA (*InterCluster Routing Algorithm*) [51], por exemplo, estabelece uma topologia hierárquica com vários níveis usando comunicação *multi-hop*, o que desobriga, pelo menos inicialmente, os nós sensores de terem o nó sorvedouro nos seus respectivos raios de alcance. Entretanto, esse protocolo permite que os nós *cluster-heads* decidam, em determinado momento, baseados em suas quantidades residuais de energia, se vão continuar a tarefa de encaminhamento. Caso optem por interromper suas tarefas de encaminhamento, os seus *cluster-heads* filhos, ao contrário da presente proposta, são obrigados a transmitir diretamente para o sorvedouro, o que mantém a restrição que o LEACH impõe de que todos os nós devem ser capazes de alcançar diretamente o sorvedouro. Além disso, o protocolo ICA não se mostra adequado a aplicações onde a latência na entrega das informações é crítica, devido ao seu esquema de transmissão de dados para o sorvedouro.

Um outro protocolo que deve ser mencionado é o TEEN [30]. Em tal protocolo são propostos aprimoramentos ao LEACH em termos de economia de energia reduzindo o envio de informações redundantes ou desnecessárias à aplicação. Esta redução é obtida flexibilizando o método de coleta para permitir tanto um esquema periódico quanto um esquema sensível a valores limites. Dessa forma, os nós só enviam informações quando o valor detectado se enquadra em um determinado perfil, pré-definido pela aplicação. Contudo, diferentemente do protocolo de disseminação de dados proposto neste trabalho, o TEEN não

prevê comunicação *multi-hop*, mantendo as transmissões diretas a partir dos *cluster-heads* para o sorvedouro, o que acarreta em gastos adicionais de energia.

Um outro trabalho relevante é o protocolo PEGASIS [29], que propõe uma solução capaz de obter ganhos significativos em termos de energia sobre o LEACH. Porém, em termos de tolerância a falha, tal protocolo apresenta o mesmo problema que o LEACH, visto que na ocorrência de falha de algum *cluster-head*, esta só é sanada durante a próxima fase de reorganização da topologia, deixando a área correspondente indisponível. Na presente proposta, o esquema de recuperação de falhas adotado reduz sensivelmente o tempo de indisponibilidade que se traduz numa menor perda de dados conforme constatado nas simulações realizadas.

Com o objetivo de atestar a melhoria de desempenho obtida com as modificações e extensões realizadas no protocolo LEACH, diversas tarefas foram executadas, tendo sido divididas em duas fases: programação e simulação. A fase de programação consistiu em incluir no código do simulador *ns-2* [39], todas as modificações e extensões propostas. Já na fase de simulação, foram coletados, para cenários representativos, parâmetros de desempenho (métricas) tais como: energia média dissipada pelos nós sensores, taxa de perda de dados, cobertura média e atraso na entrega de dados. Finalmente, ainda nessa fase, as métricas coletadas foram confrontadas com aquelas do protocolo LEACH, assim como o comportamento do protocolo proposto em cenários distintos. Foram também apontados os ganhos e as perdas obtidas.

1.2 Organização da Dissertação

O conteúdo desta dissertação está dividido em 5 capítulos, incluindo este capítulo introdutório. No capítulo 2, estão descritos os conceitos básicos que regem o funcionamento das RSSFs assim como diversos trabalhos relacionados. O Capítulo 3 descreve detalhadamente o protocolo proposto. Os detalhes de implementação da proposta no simulador *ns-2* estão expostos no Capítulo 4. Ainda nesse capítulo, são mostrados, analisados e justificados os resultados obtidos através das simulações. Finalmente, o Capítulo 5 apresenta algumas colocações finais assim como as conclusões e sugestões de trabalhos futuros.

Capítulo 2 Conceitos Básicos e Trabalhos Relacionados

Em geral, redes de sensores sem fio (RSSFs) são formadas por um conjunto de nós sensores, que são normalmente distribuídos de forma aleatória na área de interesse, e um ou mais nós **sorvedouros**. Os nós sensores são responsáveis por coletar informações do ambiente (temperatura, pressão, umidade, etc.) e enviá-las para os nós sorvedouros que, por sua vez, têm o papel de coletar as informações enviadas pelos nós sensores e entregá-las ao usuário ou aplicação final. O sorvedouro, diferentemente dos nós sensores da rede, é geralmente um nó mais poderoso computacionalmente, sem restrição de energia e localizado geograficamente distante da área onde os sensores estão instalados.

Devido às características de implantação das RSSFs, é comum ter-se cenários onde os sensores não são facilmente alcançáveis para manutenção após sua instalação. Assim, sensores são considerados dispositivos tipicamente descartáveis que duram até terem sua energia esgotada. Portanto, a energia é um recurso crucial para as RSSFs e tem que ser gerenciada de forma inteligente a fim de estender o tempo de vida da rede. Dentre as estratégias comumente empregadas para maximizar a vida útil da rede destacam-se a minimização dos seguintes parâmetros: (i) número de transmissões efetuadas, (ii) tamanho das mensagens, (iii) potência utilizada nas transmissões e (iv) consumo de energia de cada nó. Outra estratégia é manter o maior número possível de nós desligados quando, por exemplo, não estiverem sendo efetivamente usados na atividade de observação do meio. Essa estratégia é viável devido à alta densidade de nós tipicamente encontrada nesses tipos de rede, gerando um número menor de informações redundantes.

Adicionalmente, em redes de sensores sem fio é possível ter situações onde alguns ou até mesmo todos os nós são móveis. Tal mobilidade pode acontecer, por exemplo, quando o nós são implantados em animais selvagens [42].

Este capítulo apresenta os principais conceitos relativos às redes de sensores sem fio necessários para o completo entendimento da proposta desta dissertação. O capítulo está organizado em 3 seções. A Seção 2.1 apresenta uma introdução às redes de sensores sem fio (RSSFs), citando os seus principais conceitos e ressaltando as características que as diferenciam das redes tradicionais. Na Seção 2.2, os protocolos LEACH e *Directed Diffusion* são descritos, visto que o presente trabalho fundamenta-se nesses dois protocolos. A Seção 2.3, aborda os trabalhos relacionados descrevendo seis outros protocolos que foram

desenvolvidos visando adicionar aprimoramentos ao protocolo LEACH, assim como mecanismos de tolerância a falhas, que são, entre outros, os objetivos do protocolo proposto neste trabalho.

2.1 Aspectos Específicos das RSSFs

Nesta seção são enumeradas e descritas as peculiaridades das redes de sensores sem fio em relação as redes tradicionais. Dentre tais peculiaridades, pode-se citar a sua organização topológica (Seção 2.1.1), o método usado para a disseminação de dados na rede (Seção 2.1.2) e os aspectos relativos a tolerância a falhas (Seção 2.1.3). Além dessas características, outras peculiaridades das RSSFs como a mobilidade e detalhes das camadas física e de enlace são descritas na Seção 2.1.4.

2.1.1 Organização Topológica e Identificação dos nós

Uma vez que a forma como as RSSFs são organizadas (topologia) tem impacto direto no consumo de energia [56], vários protocolos para RSSFs foram propostos nos últimos anos [24,26,27,29,30,31,32]. Alguns são baseados em topologias de rede plana [26,27], onde não há nenhum tipo de hierarquização entre os nós, sendo permitida a comunicação entre quaisquer nós. Outros protocolos são baseados em topologias hierárquicas [27,29], na qual os nós são organizados em grupos (*clusters*).

Dentre os critérios usados para a divisão da rede em *clusters*, a localização geográfica dos sensores representa uma boa opção. Assim, os nós que estão localizados em regiões geográficas adjacentes são agrupados em um mesmo *cluster*. Cada *cluster* possui um líder, chamado de *cluster-head* (CH). A comunicação entre os nós de um mesmo *cluster* pode, por exemplo, ser feita através de um cronograma TDMA. Tal cronograma é definido pelo *cluster-head* e enviado via *broadcast* para todos os membros do seu *cluster*. Assim, cada nó do *cluster* transmite suas informações para o seu *cluster-head* no seu *slot* de tempo correspondente. Após o recebimento de informações de todos os membros do seu *cluster*, o *cluster-head* pode efetuar algum processamento sobre os dados recebidos (fusão, agregação) e enviar o resultado de tal processamento para o sorvedouro [28].

Uma das principais motivações para adotar um esquema baseado em *clusters* é que a distância entre membros de um mesmo *cluster* e o respectivo *cluster-head* é geralmente menor do que a distância entre eles e o sorvedouro[27]. Portanto, os sensores dentro de um *cluster* economizam energia, pois transmitem informações apenas para o seu *cluster-head* e este, por sua vez, encaminha-as para o sorvedouro. Esse mecanismo de encaminhamento gera

um grande consumo de energia para os *cluster-heads*. Uma solução para este problema é fazer uma rotação periódica do líder entre os sensores do *cluster*. Dessa forma, os nós economizam energia nos períodos em que não são *cluster-heads.* Na seção 2.3, são mostrados exemplos de protocolos hierárquicos baseados em *clusters*.

Quanto ao sistema usado para identificação dos nós, as RSSFs são exemplos de sistemas distribuídos onde as comunicações de baixo nível não são baseadas em endereços de nós, mas em atributos dos nós que sejam relevantes para a aplicação. Com um grande número de nós sensores, torna-se inviável o estabelecimento de identificadores únicos (usando endereçamento IP, por exemplo) para todos os nós, de forma que um sistema de nomes baseado em atributos [8] deve ser utilizado. Os nomes podem ser baseados no tipo de sensor (temperatura, pressão, etc.) ou na sua localização geográfica. Por exemplo, *"quais as regiões que estão apresentando uma temperatura inferior a 10 graus?"* é uma consulta mais comum em RSSFs do que *"qual a temperatura percebida por um determinado nó?"*[9]

2.1.2 Disseminação de Dados

Atualmente, o desenvolvimento de protocolos para RSSFs, capazes de disseminar os dados coletados de maneira eficiente em termos de energia, concentra grande parte dos esforços de pesquisa. Existem dois métodos básicos para transmitir dados em uma rede de sensores sem fio: transmissão direta e transmissão indireta. A comunicação direta consiste em ter os sensores transmitindo dados diretamente para o sorvedouro, gerando um consumo maior de energia devido à maior distância.

Considerando que a economia de energia é um fator crucial para RSSFs, em geral a comunicação indireta, *multi-hop*, é mais adequada do que a direta, de longa distância, assumindo que a energia despendida nas transmissões é função da distância entre transmissor e receptor [48]. Na comunicação *multi-hop*, os sensores localizados em regiões mais próximas ao sorvedouro tendem a ter suas fontes de energia esgotadas antes dos outros nós da rede, o que pode levar ao isolamento dos nós mais distantes do sorvedouro, impedindo a entrega dos dados. Uma solução para distribuir o gasto de energia uniformemente pelos nós da rede é adotar uma política onde os nós alternam-se entre as funções de sensoriamento e encaminhamento de dados, visto que a função de encaminhamento de dados é mais intensa [27] em termos de consumo de energia. Outra possível solução seria adotar arquiteturas com mais de um nó sorvedouro, para obter uma melhor distribuição do tráfego pela rede.

Quanto ao modelo de entrega de informações, os protocolos para redes de sensores sem fio podem ser classificados em periódicos [27], orientados a eventos[26] ou híbridos. Nos

protocolos periódicos, os nós coletam informações do meio e enviam os dados periodicamente para o sorvedouro.

Nos protocolos orientados a eventos, os nós sensores só enviam informações para o sorvedouro quando algum fenômeno de interesse para a aplicação é detectado, seguindo características descritas em uma solicitação (interesse) previamente submetida à rede. Tal interesse é disseminado pela rede de modo que o usuário final possa atribuir tarefas de sensoriamento para os nós da rede. Existem dois métodos mais comuns para a disseminação de um interesse por determinada informação. No primeiro, o sorvedouro faz um *broadcast* do interesse para todos os sensores da rede. Ao receber tal interesse, cada nó faz uma auto-verificação para descobrir se atende ou não ao interesse. Em caso positivo, inicia a coleta e transmissão das informações em direção ao sorvedouro solicitante. No segundo método, cada nó sensor divulga o tipo de informação que possui e, em seguida, aguarda os pedidos de coleta de informações.

Finalmente, nos protocolos híbridos, a rede pode operar tanto no modo orientado a eventos quanto no modo periódico. A vantagem de se adotar um esquema híbrido é que torna-se possível atender a uma gama maior de aplicações. Por outro lado, têm-se uma maior complexidade de implementação.

2.1.3 Tolerância a Falhas

Para avaliar plenamente o desempenho de um protocolo para redes de sensores sem fio, deve-se considerar a possibilidade de falhas nos nós. Tais falhas podem ocorrer de maneira repentina, devido a causas externas, ou por esgotamento de energia, levando um ou mais nós ao fim de sua vida útil. Os nós que falham, repentinamente, sejam eles próximos entre si ou em posições completamente aleatórias, podem ocasionar desde pequenas perdas de dados coletados até a imprecisão na informação fornecida pela rede. Essas falhas podem provocar também situações de partição da rede, caracterizadas pela existência de regiões descobertas.

Para protocolos hierárquicos, são considerados dois principais tipos de falhas nos nós. No primeiro tipo, a falha ocorre em um determinado nó que esteja desempenhando apenas o papel de nó sensor. Pode-se afirmar que as falhas do primeiro tipo não devem apresentar impactos significativos no desempenho do protocolo, exceto na questão da precisão da informação, pois tal tipo de falha pode levar uma região a ter uma redução de cobertura. Nesse caso, como exemplo, um objeto móvel cujo movimento estivesse sendo monitorado pela rede, encontrar-se-ia fora de alcance quando este passasse pela região de abrangência do nó que falhou. No segundo, as falhas ocorrem nos nós que estão desempenhando o papel de

cluster-head de um determinado *cluster*. Quando um *cluster-head* falha, toda a região coberta por esse *cluster* fica descoberta e, conseqüentemente, para agravar a situação, todas as regiões cobertas por seus *clusters* filhos também ficam descobertas.

Considerando os tipos de falha citados, pode-se afirmar que o de maior impacto para protocolos hierárquicos é aquele que apresenta falhas em nós desempenhando o papel de *cluster-head*.

Dentre os principais mecanismos existentes para o tratamento de falhas em RSSFs, pode-se destacar aqueles que exploram a alta densidade de nós visando criar nós reserva (*backup*) [55], e aqueles que adotam esquemas de roteamento por múltiplos caminhos[22] para contornar regiões de falha na rede. No primeiro mecanismo, por exemplo, cria-se um nó reserva (*backup*) que pode ser mantido em um modo de baixo consumo de energia durante a maior parte do tempo. Porém, em caso de falha de um de seus nós vizinhos, o nó reserva assume o papel de encaminhamento que antes era realizado pelo nó que falhou.

O segundo mecanismo, roteamento por múltiplos caminhos, permite o estabelecimento de múltiplos caminhos entre origem e destino, sendo um primário e os demais alternativos, conforme a classificação feita em [22]. O **caminho primário** é aquele considerado o melhor caminho entre dois nós quaisquer da rede. Assim, do ponto de vista da aplicação, o ideal é transmitir informações por este caminho. Entretanto, para que seja possível que a rede recupere-se de uma falha no **caminho primário** (de maneira escalável), é necessário que sejam construídos e mantidos um ou mais **caminhos alternativos**.

O conceito clássico de roteamento por múltiplos caminhos tem sido útil principalmente para tratar duas questões [16,17,18]. A primeira envolve o conceito de balanceamento de carga. Nesse contexto, o tráfego entre determinado par origem/destino é dividido por múltiplos caminhos na rede, com o intuito de balancear a quantidade de tráfego, distribuindo de maneira mais uniforme o consumo de energia dos nós da rede.

A segunda questão está relacionada ao aumento da confiabilidade na entrega da informação e maior tolerância a falhas. Nesse caso, múltiplas cópias de um mesmo dado são enviadas por diferentes caminhos, aumentando as chances da informação efetivamente chegar ao destino mesmo que alguns dos caminhos falhem, mas incorre em um maior consumo de energia.

O roteamento por múltiplos caminhos pode ser efetuado considerando duas estratégias: (i) caminhos disjuntos; (ii) caminhos entrelaçados.

- **Caminhos Disjuntos**

Em mecanismos que usam caminhos disjuntos, são formados caminhos alternativos que não possuem nenhum enlace em comum entre si. Além disso, estes caminhos também não apresentam interseção, em termos de nós, com o caminho primário. Conseqüentemente, estes caminhos alternativos não são afetados quando ocorrem falhas no caminho primário. Entretanto, tais caminhos podem apresentar características inadequadas para algumas aplicações. Como exemplo, pode-se imaginar que tais caminhos podem apresentar maiores latências para o tráfego de informações, sendo pouco recomendados para aplicações onde o tempo na entrega da informação é um fator crucial.

A definição desse mecanismo assume um conhecimento prévio global sobre a topologia e as características da rede. Uma solução alternativa que conta apenas com informação local de cada nó para a montagem dos caminhos é proposta em [22]. Tal solução usa o mesmo conceito de reforço de caminhos usado pelo algoritmo *Directed diffusion* [26]. A diferença básica é que o conceito é estendido para múltiplos caminhos, ou seja, logo após a criação do caminho primário, caminhos alternativos podem ser criados, um a um, usando o mesmo procedimento usado para a formação do caminho primário do algoritmo *Directed Diffusion*.

- **Caminhos Entrelaçados**

Apesar dos caminhos disjuntos apresentarem características adequadas às RSSFs em termos de tolerância a falhas, eles podem ser ineficientes em termos de energia. Tal conseqüência pode ser observada se considerarmos que estes caminhos tendem a ser mais longos que o caminho primário, consumindo uma quantidade significativamente maior de energia quando comparados a ele. Considerando que este impacto em termos de energia pode afetar drasticamente a vida útil das redes de sensores sem fio, considera-se a adoção de outro tipo de caminho: os caminhos entrelaçados. Nessa abordagem, os caminhos alternativos não são completamente disjuntos do caminho primário, ou seja, possuem alguns nós em comum com ele. Assim, a definição de um método simples para a construção destes caminhos entrelaçados poderia ser feita como descrito. Para cada nó do caminho primário, encontrar o melhor caminho que não contém o nó. Tal caminho não precisa ser completamente disjunto em termos de nó em relação ao caminho primário.

No final da execução desse algoritmo, obtêm-se um conjunto de caminhos que, incluindo o caminho primário, pode ser considerado o conjunto ideal de caminhos entrelaçados entre a origem e o destino. Além disso, esse conjunto de caminhos tende a estar

geograficamente próximo ao caminho primário, conseqüentemente tendo características de consumo de energia similares a ele, ou seja, são eficientes em termos de energia.

Assim como o mecanismo rudimentar exposto na seção anterior para formação de múltiplos caminhos disjuntos, esse mecanismo para formação de caminhos entrelaçados também assume o conhecimento de informações globais de topologia e características da rede. Porém, existem soluções [22] alternativas, que contam apenas com informações locais em relação ao nó que está executando o algoritmo, visto que em RSSFs é sempre mais adequado usar interações localizadas do que assumir conhecimento global.

2.1.4 Camada Física e Camada de Enlace nas RSSFs

Dentre as principais funcionalidades da camada física, pode-se destacar (i) a seleção da freqüência portadora que será utilizada nas transmissões entre os nós, (ii) a geração da onda portadora, (ii) a detecção de sinais sendo propagados no meio físico e, finalmente, (iii) a codificação e decodificação dos sinais sendo transmitidos.

A freqüência portadora é a freqüência nominal de oscilação da onda portadora, por intermédio da qual são transmitidas informações. Tais transmissões são executadas através de modulações feitas em algum dos três parâmetros da onda portadora: freqüência (FM – *Frequency Modulation*), amplitude (AM – *Amplitude Modulation*) e fase (PM – *Phase Modulation*). Quanto à faixa do espectro a ser usada pelas transmissões de uma RSSF, pode-se considerar a adoção de tecnologias de rádio, infra-vermelho ou meios ópticos. Para o caso de RSSFs, é necessário que a camada física seja implementada por um dispositivo transmissor de baixo consumo, baixo custo e tamanho reduzido como pode ser observado, por exemplo, nos weC Motes [5]. Nesse caso, é adotado um mecanismo de transmissão via rádio usando a faixa de 916,5 MHz.

Uma outra possibilidade para implementar a comunicação entre os nós, é a utilização de transmissores infra-vermelhos. Neste caso, diferentemente da tecnologia de rádio, a utilização do espectro é livre de licenças e não sofre interferências de dispositivos elétricos, como acontece com as opções via rádio que adotam faixas públicas do espectro (e.g., faixa ISM, *Industrial, Scientific, and Medical*). Porém, a desvantagem do infra-vermelho é a necessidade de se ter os dois dispositivos visíveis entre si, ou seja, não podem haver obstáculos entre o transmissor e o receptor. Considerando que é comum haver obstáculos nos cenários de redes de sensores sem fio, a tecnologia de rádio é em geral preferida por não apresentar tal restrição de maneira tão rigorosa.

Quanto à geração da freqüência portadora e à detecção de sinais no meio físico, pode-se dizer que tais questões estão diretamente relacionadas a aspectos do *hardware* e do transmissor adotado, sendo portanto, considerado um assunto que está fora do escopo desta dissertação. Nos próximos parágrafos, é dada uma especial atenção a questões como eficiência em termos de energia, efeitos da propagação no meio físico e esquemas de modulação para o rádio usado em redes de sensores.

Sabe-se que a comunicação sem fio de longa distância é um recurso custoso tanto em termos de energia quanto em termos de complexidade de implementação. Por exemplo, a potência mínima necessária para transmitir um sinal a uma distância d é proporcional a d^n , onde $2 <= n <= 4$. O expoente n tende a 4 para antenas pequenas e meios de propagação próximos ao chão [3]. Portanto, como é comum em aplicações de RSSF ter-se dispositivos com estas características, a questão da comunicação fica ainda mais dificultada, visto que quantidades maiores de energia tendem a ser despendidas com as transmissões.

Em relação ao esquema de modulação adotado, deve-se ressaltar que é uma questão crítica para uma comunicação confiável em redes de sensores. Esquemas de modulação binária e n-ária são comparados em [4]. Esquemas n-ários são capazes de reduzir o tempo gasto pelo nó sensor com o processo de transmissão. Neste caso, um mesmo símbolo pré-definido do protocolo transporta múltiplos bits de informação. Porém, tal mecanismo implica a utilização de circuitos mais complicados e um consumo de energia mais elevado do que o método tradicional binário, onde os bits de informação são transmitidos um por vez e apresentam circuitos mais simples, além de um consumo de energia mais adequado para redes de sensores sem fio.

A camada de enlace é responsável pela multiplexação de vários fluxos de dados, delimitação de quadros de dados, controle de acesso ao meio físico e controle de erros. Um dos principais objetivos da camada de enlace de uma rede sem fio é transformar um meio não confiável em um meio confiável.

Um protocolo de enlace para redes de sensores sem fio deve atender a dois objetivos principais. O primeiro objetivo é a criação de uma infra-estrutura de comunicação entre os nós. Sabendo que é comum ter-se uma grande quantidade de nós densamente espalhados pelo meio, a camada de enlace deve estabelecer enlaces de comunicação entre os nós da rede para a transferência de dados. Tais enlaces formam a infra-estrutura básica necessária para a comunicação *multi-hop* entre os nós da rede.

O segundo grande objetivo da camada de enlace é compartilhar eficientemente e de maneira justa os recursos de comunicação entre os nós da rede.

Em relação aos protocolos de enlace, constata-se que os protocolos usados em redes sem fio tradicionais possuem características que não são adequadas à realidade das redes de sensores sem fio, as quais, como visto, possuem restrições em termos de recursos de processamento, de memória e, principalmente, de energia. Por exemplo, a limitação quanto ao recurso de energia em uma rede celular não é um fator crucial. As baterias dos celulares podem ser recarregadas periodicamente e as estações-base possuem energia "ilimitada". Nesse tipo de rede, os protocolos de enlace estão voltados para questões como qualidade de serviço e utilização eficiente da banda disponível.

Existem alguns protocolos de enlace clássicos para redes sem fio que devem ser mencionados. Entre eles, destacam-se os protocolos TDMA, FDMA, CDMA, CSMA e CSMA/CA. No protocolo TDMA, o tempo é dividido em períodos chamados *slots*. Cada *slot* é atribuído a um único nó, de modo que durante tal período de tempo apenas um transmissor possa usar o meio físico para transmissões. Tal método é suficiente para impedir acessos simultâneos. Para o caso do protocolo FDMA, é usado um esquema onde todos os nós podem transmitir simultaneamente, porém, cada um em uma freqüência. Uma conseqüência imediata dessa divisão do espectro em várias faixas é a redução da largura de banda disponível para cada um dos nós. Da mesma maneira, o protocolo CDMA também permite transmissões simultâneas. Entretanto, tais transmissões podem ser feitas na mesma freqüência. Para isso, é adotado um esquema de codificação dos sinais transmitidos de modo que os receptores sejam capazes de filtrar apenas o sinal de interesse. Tal processo de codificação incrementa substancialmente o tamanho de cada unidade de informação transmitida, conseqüentemente diminuindo a largura de banda disponível.

Existem, também, soluções híbridas para a camada de enlace, baseadas em TDMA e FDMA [4]. Enquanto um esquema completamente baseado em TDMA dedica toda a banda disponível para um único nó por vez, um esquema puramente FDMA dedica uma banda mínima para um determinado nó, viabilizando a participação de múltiplos nós simultaneamente, em freqüências diferentes. Apesar da solução TDMA oferecer uma taxa de dados maior devido a uma banda passante disponível maior, ela nem sempre é a melhor opção para a camada de enlace devido aos custos de sincronização demandados. Em [4], é apresentada uma fórmula para RSSFs que adotam o esquema híbrido. Tal fórmula busca encontrar um número ótimo de canais FDMA e promover a máxima economia de energia. Analisando a fórmula, percebe-se que o número de canais é função da razão entre o consumo de energia das transmissões e das recepções. Além disso, a solução híbrida tende para o TDMA quando as transmissões são mais custosas do que as recepções. Em contrapartida, a

mesma solução tende para o FDMA quando os circuitos de recepção dos nós consomem mais energia.

Outro tipo de protocolo clássico para redes sem fio é aquele baseado em contenção. Por exemplo, o protocolo CSMA (*Carrier Sense Multiple Access*) faz com que cada nó, antes de transmitir, verifique se o meio físico está livre. Caso não esteja, o nó é obrigado a aguardar a sua liberação antes de transmitir. Tal processo de contenção visa evitar colisões. Porém, a possibilidade de colisões persiste e pode ser ainda tratada com o protocolo CSMA/CA (*Carrier Sense Multiple Access/Collision Avoidance*). Tal protocolo conta com um esquema para evitar colisões que obriga cada nó a enviar um sinal em *broadcast* antes de qualquer transmissão. Este sinal tem como objetivo reservar o meio para a sua transmissão.

Outros protocolos de enlace possuem características mais adequadas às RSSFs. Entretanto, esses protocolos ainda não se enquadram completamente nas necessidades específicas de uma RSSF. Por exemplo, o protocolo definido pelo padrão IEEE 802.11 é normalmente empregado na formação de WLANs (*Wireless Local Area Networks*). Redes implementando tal padrão apresentam um alcance da ordem de centenas de metros tendo seus dispositivos de rádio operando em faixas de freqüência de domínio público (e.g.: ISM - *Industry, Science and Medicine*). Tal padrão não leva em conta aspectos importantes para redes de sensores sem fio como, por exemplo, a otimização do consumo de energia.

Sintetizando, os protocolos de enlace especialmente projetados para redes de sensores sem fio devem otimizar o seu consumo de energia. Isso pode ser feito, por exemplo, evitando o uso de mecanismos de acesso ao meio baseados em contenção, para que os nós não precisem ficar monitorando o canal para descobrirem o momento adequado para transmitir, o que não é econômico em termos de energia.

O protocolo SMACS [6], que é um protocolo próprio para redes de sensores sem fio, usa um método distribuído para construir uma infra-estrutura de comunicação. Esse método envolve a descoberta de vizinhos por parte de cada nó da rede e, também, o estabelecimento de cronogramas para transmissão e recepção. O estabelecimento desse cronograma, conforme mencionado, é distribuído e, portanto, não é necessária a presença de nenhum nó intermediário responsável por centralizar esse papel. Em tal protocolo, a comunicação entre dois nós é feita através de um par de *slots* de tempo operando em uma freqüência fixa aleatoriamente escolhida a partir de um conjunto de freqüências pré-definidas. Tal processo evita a necessidade de haver uma sincronização global envolvendo todos os nós da rede. Portanto, é necessário apenas sincronização local entre os dois nós que precisam se

comunicar. Adicionalmente, pode-se economizar energia mantendo os nós desligados durante os *slots* de tempo que não podem transmitir ou quando não possuem dados a serem enviados.

Devido ao fato das redes sem fio apresentarem taxas de erro elevadas, é também importante que possuam mecanismos de detecção/correção de erros nas camadas inferiores da pilha de protocolos. Portanto, a camada de enlace deve prover algum mecanismo nesse sentido, visando evitar que os erros sejam apenas detectados/corrigidos nas camadas superiores, onde as unidades de informação são maiores, o que implica um maior custo de recuperação.

A ação executada pela camada de enlace diante da recepção de uma informação com erro pode consistir em um processo de detecção e notificação ao remetente, ou pode contar com um mecanismo que seja capaz de corrigir a informação que foi recebida com erro. Para a primeira abordagem, basta que o transmissor acrescente informações redundantes que sejam suficientes para o receptor detectar se há ou não algum erro (CRC - *Cyclic Redundancy Check*), e solicitar a retransmissão (ARQ - *Automatic Repeat Request*).

Para a segunda abordagem, onde o receptor precisa corrigir o erro detectado, o processo demanda uma inclusão maior de informações redundantes no dado transmitido para que a informação original possa ser reconstruída em caso de erro. Um exemplo de tal abordagem pode ser encontrado em [7], onde tem-se um estudo da utilização do *Hamming Code* em redes de sensores sem fio. Essa abordagem é chamada de FEC (*Forward Error Correction*), e gera uma complexidade de decodificação maior, considerando que tal mecanismo deve ser embutido no circuito de recepção. Entretanto, pode ser adequado adotar esquemas baseados em códigos mais simples do que o *Hamming Code* para o controle de erros. Esses esquemas devem ter um processo de codificação e decodificação o mais simples possível, visando sempre a economia de energia.

2.1.5 Mobilidade

A maioria das arquiteturas para redes de sensores sem fio assume que os nós sensores da rede são estacionários. Entretanto, muitas aplicações podem apresentar mobilidade tanto no nó sorvedouro como nos nós sensores [23, 24, 25]. Em uma rede de sensores sem fio onde há mobilidade, a troca periódica de informações sobre rotas entre os nós é fundamental para os protocolos de disseminação, visto que a existência de rotas válidas durante o processo de monitoramento é crucial para o sucesso na entrega das informações coletadas ao seu destino final, considerando variações de topologia ao longo do tempo. Somando-se a este fator questões como a economia de energia e a utilização eficiente da banda disponível, torna-se

um verdadeiro desafio o projeto de um protocolo que contemple todas estas questões. Além disso, os fenômenos sendo monitorados podem ser tanto dinâmicos quanto estáticos, dependendo da aplicação. Por exemplo, em uma aplicação de detecção/acompanhamento de objetos móveis, têm-se características dinâmicas. Em contrapartida, a monitoração de uma floresta consiste em uma aplicação estática.

Para viabilizar a manutenção da conectividade diante da mobilidade dos nós, existem protocolos que montam estruturas de disseminação capazes de permitir uma perfeita conectividade da rede, mesmo quando o sorvedouro está se movendo. Por exemplo, o protocolo TTDD (*Two-Tier Data Dissemination*) [24] descreve um método para disseminação de dados em dois níveis, que cria um esquema escalável e eficiente para a entrega de informações para múltiplos sorvedouros móveis. Nesse método, cada nó fonte de informação constrói de maneira pró-ativa uma estrutura em forma de grade. T*al* grade é formada por um conjunto de células quadradas e permite que os sorvedouros móveis recebam informações continuamente, mesmo durante a movimentação, tendo apenas que divulgar seu interesse pelas informações através de um método de inundação local em relação a uma única célula. Este protocolo é adequado para aplicações onde os nós sensores são estáticos e conscientes de sua localização geográfica para viabilizar a construção das grades com um mínimo de *overhead*.

Em contrapartida, o protocolo SEAD(*Scalable Energy-efficient Asynchronous Dissemination*)[25], que tem como alvo o mesmo tipo de aplicação que o TTDD, ou seja, aplicações onde os sorvedouros são móveis e o restante da rede é estático, obtém menores níveis de consumo de energia com a construção e a manutenção das estruturas de disseminação, neste protocolo, chamadas de *d-trees*. Tais estruturas são sempre acessadas pelos sorvedouros através de um nó folha da estrutura chamado de "nó de acesso". Porém, tal economia de energia é balanceada com um aumento no atraso da entrega de informações. Estes fatores podem ser balanceados variando-se um parâmetro do protocolo. Tal parâmetro define o número de saltos máximo que um determinado sorvedouro pode ficar em termos de distância até o seu nó de acesso à *d-tree*.

2.2 Protocolo LEACH e *Directed Diffusion*

Esta seção descreve as características principais de dois protocolos de disseminação de dados para RSSFs. Tais protocolos estão relacionados à proposta desta dissertação, visto que têm seus resultados de simulação comparados aqueles obtidos nas simulações do protocolo proposto. Trata-se do protocolo LEACH [27], que é um protocolo hierárquico, baseado em

clusters e voltado para coletas periódicas, e o *Directed Diffusion* [26], que é um protocolo de topologia plana e adota um esquema orientado a interesses para a coleta de informações continuamente ou de maneira periódica.

No protocolo LEACH, cada *cluster* é representado por um nó *cluster-head*. Tal nó é responsável por receber as informações de cada um dos nós de seu *cluster* e enviá-las diretamente para o sorvedouro. Considerando que tais transmissões, por serem feitas diretamente para o nó sorvedouro, fazem com que o papel dos *cluster-heads* seja mais custoso em termos de energia, faz-se necessário que haja um esquema de balanceamento do consumo de energia entre os nós. No protocolo LEACH, existe um processo de rotação periódica dos *cluster-heads*. Tal processo de rotação é distribuído de modo que periodicamente todos os nós da rede entram na fase de rotação. Nessa fase, cada nó decide se passa a ser *cluster-head* ou não, considerando uma determinada probabilidade. Após cada fase de rotação os nós sabem qual o seu novo papel na rede (*cluster-head* ou nó puramente sensor), e a que *cluster* pertencem. Uma característica limitante deste protocolo é que ele assume que todos os nós são capazes de transmitir informações diretamente para o nó sorvedouro, visto que qualquer nó pode assumir o papel de *cluster-head*.

Quando os *clusters* estão sendo formados, um determinado número de nós P (que é um parâmetro pré-configurado em todos os nós da rede), elegem-se como *cluster-heads*. A decisão de se tornar um *cluster-head* é feita escolhendo-se um número aleatório entre 0 e 1. Se o número gerado for menor do que um limite T então o nó se torna um *cluster-head* e divulga essa informação através de um sinal em *broadcast*. Os nós que são *cluster-heads* no round 0, não podem ser novamente nas próximas 1/P rodadas. Assim, a probabilidade de um nó que ainda não foi *cluster-head*, passar a ser, vai aumentando visto que existem menos nós elegíveis. Os nós decidem a qual *cluster* vão pertencer baseados na potência dos sinais de divulgação recebidos dos *cluster-heads*. Isso é feito porque é provavelmente este líder que se encontra mais perto e, portanto, será o que exigirá o menor gasto de energia para a comunicação. O *cluster-head* recebe todas as mensagens dos nós que gostariam de pertencer ao seu grupo. Baseado no número de nós no *cluster*, o *cluster-head* cria um cronograma TDMA informando a cada nó quando ele pode transmitir. Este cronograma é transmitido para todos os nós do *cluster*.

Assim que os *clusters* são formados e o cronograma TDMA transmitido, a transmissão dos dados pode começar. Assumindo que os nós sempre possuem dados a enviar eles o enviam para o *cluster-head* durante o tempo alocado a ele para fazer esta transmissão. Além

disso, tais transmissões são feitas com um código CDMA diferente para cada cluster, visando evitar interferências entre *clusters* vizinhos.

O protocolo *Directed Diffusion* é um exemplo de protocolo de disseminação de dados eficiente para redes de sensores. Cada nó transforma o sinal gerado pelo alvo que está sendo monitorado em uma descrição relativamente simples de um evento. Tal descrição possui um conjunto de atributos. Aplicações que desejam obter informações enviam seus interesses também descritos em forma de atributos através de algum nó sorvedouro usando um método de inundação que difunde o interesse para todos os nós da rede. Quando tal interesse é recebido por um nó que se enquadra em suas descrições, o nó passa a enviar as informações desejadas também através de um método de inundação, visando fazer com que tais informações alcancem o sorvedouro que originou o interesse. Porém, o envio dessas informações é feito inicialmente em baixa freqüência. Quando o sorvedouro recebe tais informações, possivelmente duplicadas e por vários caminhos, ele escolhe o caminho que apresentou a melhor qualidade segundo alguma métrica (ex: latência), e envia um pedido de reforço por este caminho, rumo ao nó fonte das informações. Esse pedido de reforço marca tal caminho de modo que o nó fonte, ao recebê-lo, passa a enviar informações através do caminho reforçado e usando uma freqüência mais alta. Para tal, cada nó da rede precisa manter uma tabela de interesses e informações locais sobre os caminhos que são formados. Além disso, o sorvedouro precisa reenviar periodicamente o interesse para manter os processos de coleta de informações desejados ativos.

Finalmente, tal protocolo é iniciado pelo receptor, orientado a interesses (possibilitando uma coleta contínua ou periódica) e adota uma topologia de rede plana.

2.3 Trabalhos Relacionados

Nesta seção são apresentados seis outros protocolos que, da mesma forma que o presente trabalho, consistem em protocolos hierárquicos para disseminação de dados em RSSFs. Nas seções 2.3.1, 2.3.2 e 2.3.3 são apresentadas as principais características dos protocolos PEGASIS, TEEN e ICA, considerando que possuem topologias hierárquicas e que têm como objetivo aprimorar o protocolo LEACH em termos de eficiência no consumo de energia. Nas seções 2.3.4, 2.3.5 e 2.3.6, são apresentados protocolos existentes na literatura que propõem mecanismos de tolerância a falhas usando topologias hierárquicas.

2.3.1 PEGASIS

O protocolo PEGASIS (*Power Efficient Gathering in Sensor Information Systems*)[29], tem o seu funcionamento baseado na montagem de uma estrutura linear contendo todos os nós da rede. Tal estrutura pode ser montada de maneira centralizada ou distribuída. No primeiro caso, o sorvedouro monta a estrutura localmente e divulga a topologia. No segundo, um algoritmo guloso é executado a partir do nó mais distante do sorvedouro, de modo que forme um caminho que passe por todos os nós da rede. Concluída a formação da estrutura linear, o protocolo entra em um ciclo dividido em uma fase de coleta de informações e uma fase de remontagem da estrutura. Tais fases são separadas no tempo pela morte de algum nó, ou seja, toda vez que algum nó da rede morre, toda a estrutura deve ser remontada. A fase de coleta de informações envolve a eleição de um único líder para a estrutura. Tal líder é responsável por gerar *tokens* e transmití-los para os seus vizinhos que, por sua vez, vão retransmitindo-os até os nós folha da estrutura. Quando os nós folha recebem o *token*, enviam suas informações para o vizinho que enviou o *token*. Tal vizinho agrega as suas informações ao pacote recebido e encaminha o novo pacote adiante, rumo ao líder que, ao receber um pacote de informação, o envia para o nó sorvedouro. Vale ressaltar que o processo de eleição do líder da rodada adota um esquema capaz de eleger um nó da rede por vez, visando também uma distribuição uniforme do consumo de energia.

2.3.2 TEEN

Considerando agora o protocolo TEEN (*Threshold sensitive Energy Efficient sensor Network*) [30], trata-se de um protocolo de roteamento hierárquico, de coleta contínua, similar ao LEACH, exceto pelo fato de que os nós sensores podem não possuir dados a serem transmitidos de tempos em tempos. Ele utiliza a estratégia de formação de *clusters* do LEACH, mas adota uma estratégia diferente na fase de transmissão dos dados. Ele faz o uso de dois parâmetros chamados *Hard Threshold* (Ht) e *Soft Threshold* (St) para determinar a necessidade de transmissão do dado coletado. Se o valor detectado exceder *Ht* pela primeira vez, ele é armazenado em uma variável e transmitido durante o intervalo (*slot*) de tempo alocado à transmissão do nó. Em seguida, se o valor monitorado exceder o valor armazenado por uma magnitude de *St* o nó transmite o dado imediatamente. O valor enviado é armazenado para comparações futuras.

2.3.3 ICA

O protocolo ICA (*Inter Cluster Routing Algorithm*)[51] consiste em uma solução hierárquica, baseada em *clusters* para disseminação de dados em RSSFs. O objetivo principal de tal protocolo é obter uma vida útil mais longa para a rede, tendo como base de comparação o protocolo LEACH.

Dentre as principais inovações do ICA em relação ao LEACH, pode-se destacar (i) a topologia hierárquica adotada, (ii) o mecanismo de recusa de encaminhamento por parte dos *cluster-heads* e (iii) a sincronização entre as transmissões dos *cluster-heads*. Em relação à topologia hierárquica adotada, pode-se dizer que é criado um *backbone* de *cluster-heads* da mesma maneira como é feito no protocolo proposto nesta dissertação. Tal *backbone* visa permitir a comunicação indireta entre os *cluster-heads* mais distantes e o sorvedouro, economizando energia com as transmissões de longa distância.

O protocolo ICA provê um mecanismo através do qual os nós *cluster-heads* podem se recusar a encaminhar, rumo ao sorvedouro, informações provenientes de outros *cluster-heads*. Com isso, torna-se possível poupar os nós mais próximos ao sorvedouro, permitindo que eles decidam, baseados na energia residual disponível, se aceitam ou não encaminhar informações de outros *cluster-heads* mais distantes do sorvedouro. Com isso, inicialmente, todos os *cluster-heads* encaminham as informações para os seus *cluster-heads* pais e, progressivamente, vão se recusando a efetuar tal encaminhamento, começando pelos *cluster-heads* mais próximos ao sorvedouro, pois são eles que consomem mais energia. Uma vez que um *cluster-head* se recusa a encaminhar informações, o outro *cluster-head*, que solicitou o encaminhamento, deve passar a efetuar seus envios diretamente para o sorvedouro, o que mantém a restrição imposta pelo protocolo LEACH onde todos os nós precisam alcançar diretamente o sorvedouro, restringindo o tamanho máximo da área de monitoração ao alcance do rádio dos nós.

Em relação à sincronização entre as transmissões, o ICA faz com que cada *cluster-head* da rede calcule o momento da sua transmissão baseado na sua distância até o sorvedouro, de modo que os nós mais distantes transmitam primeiro, e, em seguida, os nós mais próximos, gerando uma onda em direção ao sorvedouro. Dessa forma, são evitadas colisões entre os *cluster-heads* localizados em posições com distâncias diferentes. Para os *cluster-heads* localizados em regiões eqüidistantes em relação ao sorvedouro, inclui-se uma componente aleatória ao momento da transmissão, visando evitar colisões.

Finalmente, deve-se ressaltar o fato de que o protocolo ICA é adequado apenas para aplicações de coleta periódica, não apresentando nenhuma funcionalidade para esquemas de coleta orientados a interesse.

2.3.4 Um mecanismo baseado em consenso

Em [34], é proposto um mecanismo de tolerância a falhas para redes de sensores sem fio hierárquicas. O principal objetivo do método proposto é prover, em tempo de execução, a reintegração dos sensores de um *cluster* cujo *cluster-head* apresentou falhas. Tal mecanismo é dividido em duas etapas: detecção e reintegração. Para que seja possível reintegrar os sensores de um *cluster*, é importante detectar com precisão se ocorreu ou não uma falha no sistema. O método de detecção segue um modelo orientado ao consenso. Em tal modelo, os *cluster-heads* devem convergir para um consenso antes que uma falha seja efetivamente oficializada. A necessidade de haver um consenso é enfatizada pois é preciso manter a consistência entre os nós da rede em termos do *status* e da cardinalidade de cada *cluster-head*. A cardinalidade de um *cluster-head* reflete o número de sensores que pertencem ao seu *cluster*. Esta solução também propõe mecanismos para evitar conflitos entre as informações de status possuídas pelos vários *cluster-heads* de uma rede hierárquica.

A segunda fase provê mecanismos para identificar o tipo de falha ocorrida, assim como mecanismos para reintegrar os nós da rede que perderam a sua conectividade devido à falha. Para tal, durante o processo de *clusterização*, cada nó *cluster-head* cria um conjunto chamado *RSet*, contendo todos os nós que estão dentro do seu alcance de rádio e que pertencem ao seu *cluster*. Adicionalmente, para implementar o mecanismo de reintegração, cada *cluster-head* cria um outro conjunto chamado *BSet*, contendo nós que estão em seu alcance de rádio mas não pertencem ao seu *cluster*. Dessa forma, quando um nó precisa ser reintegrado à rede, cada *cluster-head* verifica se o nó pertence ao seu conjunto *BSet*. Caso pertença, o nó é re-associado ao *cluster* em questão. Além disso, caso o nó pertença a múltiplos *BSets*, ele é associado ao *cluster-head* com menor custo de comunicação.

2.3.5 SOP (Self Organizing Protocol)

Em [32], é descrito um protocolo chamado SOP(*Self Organizing Protocol*), auto-organizável e adequado a arquiteturas dotadas de sensores heterogêneos. Esses sensores podem ser móveis ou estacionários. Alguns dos sensores monitoram o ambiente e encaminham os dados coletados para um conjunto de nós que são chamados de roteadores. Os

nós roteadores são estacionários, possuem as mesmas características físicas que os nós sensores, e formam um *backbone* de comunicação. Com isso, os dados coletados são enviados para o sorvedouro através dos nós roteadores. Portanto, cada nó sensor deve ser capaz de alcançar um nó roteador. Uma arquitetura de roteamento que adota um mecanismo capaz de endereçar cada nó através do roteador ao qual ele está conectado é usada. Além disso, tal arquitetura é hierárquica, onde grupos de nós são formados e fundidos quando necessário.

Nesse protocolo, os nós roteadores se auto-configuram formando uma rede, enquanto os nós puramente sensores têm o papel de manterem-se sempre conectados ao roteador mais próximo que está "vivo". O protocolo consiste em 4 fases. Na primeira, chamada de Fase de Descoberta, cada nó descobre quem são seus vizinhos e define o raio máximo para suas transmissões. Na segunda fase, chamada de Fase de Organização, os nós se organizam em grupos, formando uma hierarquia. Além disso, a cada nó é alocado um endereço baseado em sua posição dentro da hierarquia e também é computada uma tabela de roteamento para todos os nós da rede. Finalmente, é criada uma árvore de *broadcast* atingindo todos os nós da rede.

Na terceira fase, chamada de Fase de Manutenção, cada nó acompanha periodicamente o seu nível de energia, e envia uma mensagem *"estou vivo"* para seus vizinhos. Além disso, as tabelas de roteamento são atualizadas e a árvore de *broadcast* para tolerância a falhas é re-organizada.

Na quarta fase, chamada de Fase de Re-organização, os nós detectam partições nos grupos, ou falhas nos nós, e atualizam suas tabelas de roteamento, baseando-se na nova topologia. Se todos os vizinhos de um nó falharem, então o nó deve repetir a Fase de Descoberta.

Finalmente, o protocolo tem como foco principal a economia de energia, minimizando o seu consumo em todas as fases. Além disso, as estruturas em árvore são tolerantes a falha pois os nós que falham são sempre transformados em folhas da árvore.

2.3.6 STALK

Em [33] é apresentado um protocolo hierárquico tolerante a falhas que consiste basicamente em um algoritmo para acompanhamento de objetos móveis em RSSF chamado *Stalk* (*Stabilizing Tracking viA Layered linKs*). Tal algoritmo emprega uma estrutura hierárquica de acompanhamento que funciona como um conjunto de caminhos impostos entre os *clusters* formados em uma outra estrutura subjacente, também hierárquica. Para atualizar a estrutura de acompanhamento, que é implementada na camada superior, são adotadas duas ações básicas: crescer (*grow*) e diminuir (*shrink*). A ação *grow* permite que um caminho

traçado por um objeto móvel "cresça" a partir da nova localização até níveis superiores da hierarquia, conectando-se ao caminho original. A ação *shrink*, por sua vez, limpa as ramificações da hierarquia que foram abandonadas pelo objeto móvel. Da mesma forma que a ação *grow*, a ação *shrink* é iniciada nos níveis mais inferiores da hierarquia rumo aos níveis superiores. Apesar de ambas as ações serem executadas ao longo da rede concorrentemente, o acompanhamento de objetos móveis conta ainda com temporizadores adequadamente escolhidos para que seja possível determinar o momento ideal para a execução de cada uma das ações. Desse modo, o protocolo *Stalk* é capaz de evitar a propagação de falhas na estrutura além de um determinado número de níveis na hierarquia.

Partindo de um estado arbitrário de falha, esse protocolo atende à sua especificação em termos de tempo e comunicação de maneira proporcional ao tamanho da falha ocorrida na rede. Isso é possível pois o protocolo faz com que a velocidade de propagação das informações de falha seja inversamente proporcional ao nível na hierarquia, ou seja, níveis mais altos propagam tal tipo de informação mais lentamente, permitindo que as informações mais recentes geradas nos níveis inferiores possam sobrescrever as informações desatualizadas dos níveis superiores. Além disso, o algoritmo apresenta resultados positivos em termos de desempenho, pois é capaz de acompanhar o movimento de um objeto através de um percurso de distância d, por exemplo, com uma complexidade em termos de tempo e comunicação para atualizar a estrutura igual a O(d * log D), onde D representa o diâmetro da rede.

2.4 Considerações Finais

Neste capítulo foram apresentados alguns conceitos que servem de preparação para o entendimento do próximo capítulo. Tais conceitos foram apresentados através de descrições breves de outros trabalhos além da descrição de alguns assuntos já tradicionais para redes convencionais, porém, ressaltando as diferenças quando aplicados às redes de sensores sem fio.

No próximo capítulo, é apresentada a proposta propriamente dita desta dissertação. Adicionalmente, o capítulo contempla um problema periférico encontrado durante o desenvolvimento deste trabalho de pesquisa. Tal problema está relacionado a limitações de existentes na plataforma de *hardware* adotada nas simulações.

Capítulo 3 Protocolo Tolerante a Falhas para RSSFs

3.1 Introdução

O objetivo deste capítulo é propor um protocolo para disseminação de dados em redes de sensores sem fio tolerante a falhas e eficiente em termos de energia. O protocolo proposto apresenta algumas características que o diferenciam do protocolo de topologia hierárquica LEACH [27] e outras similares ao protocolo de topologia plana *Directed Diffusion* [26].

Primeiramente, o protocolo organiza a rede em *clusters* e, diferentemente do protocolo LEACH, por exemplo, não exige que os líderes de *clusters*, denominados *cluster-heads*, sejam capazes de estabelecer uma comunicação direta com o sorvedouro. Para aumentar a vida útil da rede, os dados são transmitidos usando comunicação *multi-hop* entre os *cluster-heads* e o nó sorvedouro.

A segunda característica relevante é que o protocolo, assim como no protocolo de topologia plana Directed Diffusion, adota uma política de ativação baseada em um esquema *publish/subscribe*. Uma vez que os nós já estão divididos em *clusters*, interesses representando a consulta do usuário são enviados para a rede, que opera como um repositório temporal de informações, i.e., as informações fornecidas mudam de acordo com o estado do ambiente e de acordo com a necessidade do usuário. O protocolo proposto foi projetado de forma a atender as necessidades de diferentes classes de aplicações, garantindo ainda assim o consumo de energia. Em outras palavras, o protocolo proposto é eficiente em termos de consumo de energia, funciona sob demanda e é baseado no interesse da aplicação cliente. Tal interesse dispara o envio de dados que pode acontecer tanto periodicamente como apenas na ocorrência de eventos interessantes.

Adicionalmente, o protocolo proposto conta com um mecanismo de tolerância a falhas que visa evitar partições na rede após falhas apenas em nós *cluster-head*. A solução de contingência para os *clusters* em caso de falhas é obtida através da criação de um *cluster-head* reserva.

Em suma, a proposta desta dissertação consiste em um protocolo hierárquico e tolerante a falhas de disseminação de dados para RSSFs. Esse protocolo adota uma hierarquia de dois níveis de comunicação. O primeiro nível ocorre entre os sensores de um *cluster* e seu respectivo *cluster-head* e o segundo entre os *cluster-heads* e o sorvedouro.

Quanto à organização, esse capítulo foi dividido em três seções, incluindo esta Seção introdutória. Na Seção 3.2, o protocolo é descrito em linhas gerais. Na Seção 3.3, é detalhado

o funcionamento do protocolo, abordando todas as informações necessárias para o completo entendimento da proposta seguindo a organização a seguir.

3.2 Funcionamento Geral

O objetivo do protocolo proposto é, em linhas gerais, organizar as atividades de coleta de dados em um conjunto de nós espalhados em um ambiente que se deseja monitorar. O funcionamento de tal protocolo pode ser descrito em 3 grandes etapas: *Inicialização*, *Rotação e Coleta/Encaminhamento*.

Na fase de inicialização do protocolo, os nós são responsáveis por descobrir o seu papel na rede. Nessa fase, conforme explicado na Seção 3.3.3, cada nó é informado pelo sorvedouro se ele foi eleito como um novo *cluster-head* (CH) ou se ele é um nó meramente sensor. No caso de nós que são apenas sensores, eles são informados sobre a qual *cluster* pertencem e quem é o seu *cluster-head*. Para o caso dos nós *cluster-heads*, eles são informados sobre quem é o seu *cluster-head* pai na árvore de *cluster-heads* e também sobre quem são seus *cluster-heads* filhos, se houver algum. Desse modo, monta-se uma árvore formada por *cluster-heads* cuja raiz é o sorvedouro.

Em seguida, o protocolo entra em um ciclo de execução, que é dividido nas fases de *Rotação* e *Coleta/Encaminhamento*. A fase de Rotação, assim como no protocolo LEACH [27], acontece periodicamente e é responsável pela eleição de um novo *cluster-head* para cada *cluster*, visando uma distribuição justa do consumo de energia.

Na fase de Coleta/Encaminhamento, os nós puramente sensores têm o objetivo de coletar informações sobre o meio monitorado e enviá-las para o *cluster-head* do seu *cluster*. A transmissão dessas informações por parte dos nós de um mesmo *cluster* é regida pelo protocolo TDMA, onde é atribuído um *slot* de transmissão para cada nó. O *cluster-head* do *cluster*, por sua vez, tem o papel de receber tais informações e encaminhá-las para o próximo *cluster-head* da árvore no caminho até o sorvedouro. Esse último *cluster-head* também tem o papel de receber informações diretamente dos seus *cluster-heads* filhos e encaminhá-las para o próximo *cluster-head* da árvore no caminho até o sorvedouro. Tal processo se repete até as informações chegarem ao sorvedouro. As transmissões entre os *cluster-heads* são chamadas de **transmissões de encaminhamento**.

A atividade de coleta de informações do nó sensor é composta por quatro módulos conceituais, conforme ilustrado na Figura 1: Módulo Controlador, Módulo de Rádio, Módulo

de Comunicação Inter-*Cluster* e Módulo de Comunicação Intra-*Cluster*. Tais módulos conceituais são detalhados a seguir:

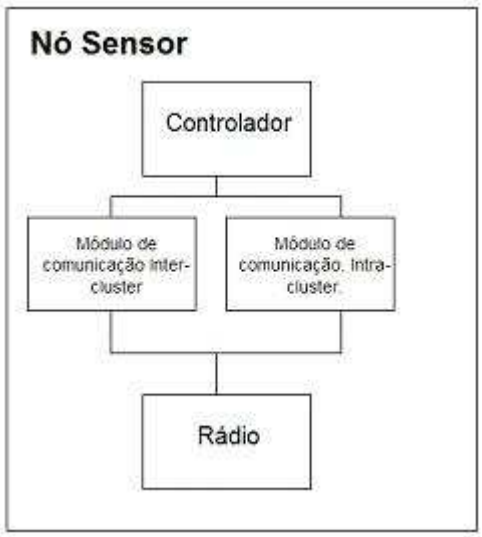

Figura 1- Processo de Coleta de Informação

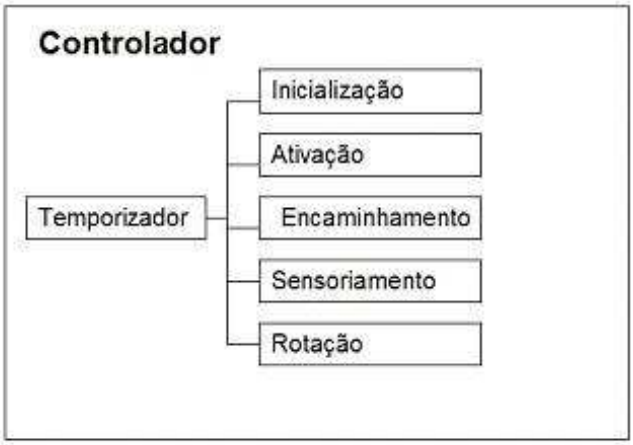

Figura 2 - Sub-Módulos do Módulo Controlador

3.2.1 Controlador

O processo de coleta de informações é coordenado pelo módulo **Controlador**, conforme ilustrado no diagrama da Figura 2**.** O **Controlador** é a entidade onde se concentra toda a inteligência do protocolo e deve ser igualmente implementado em todos os nós da rede. Dentre as suas funcionalidades, destacam-se as tarefas executadas pelos sub-módulos de Inicialização, Ativação, Encaminhamento, Sensoriamento, Rotação e Temporizador, conforme descritos a seguir.

- **Inicialização -** O sub-módulo de Inicialização é executado uma única vez e logo após a instalação dos nós no ambiente de monitoramento. Sua execução consiste basicamente em efetuar uma troca de algumas mensagens necessárias para que o nó deixe de ser apenas um nó isolado e passe a ser um membro de uma rede organizada, capaz de fazer com que informações coletadas no ambiente cheguem ao nó sorvedouro. Durante este processo, conforme detalhado na Seção 3.3.1, o nó sorvedouro recebe as coordenadas de todos os nós da rede. Em seguida, o nó sorvedouro, com posse das coordenadas de todos os nós, define a topologia global da rede e a divulga para os sensores.

- **Ativação -** O sub-módulo de Ativação é executado no momento da recepção de uma ordem específica proveniente do nó sorvedouro ou de maneira programada, por intermédio do sub-módulo Temporizador, que é descrito em seguida. Um exemplo da forma programada ocorre no processo de ativação do *cluster-head* reserva. O nó *cluster-head* reserva, definido na Seção 3.3.8, fica desativado a maior parte do tempo e é ativado apenas no momento em que seu sub-módulo Temporizador chega a zero.

  O processo de desativação tira proveito da habilidade que os sensores têm de se manterem desligados em determinadas situações com o intuito de economizar energia. Por exemplo, quando uma aplicação, logo após executar o processo de inicialização, não tem nenhum interesse em monitorar o ambiente, os nós da rede são desativados pelo respectivo módulo de controle de cada um, através do sub-módulo de Ativação. Os sensores permanecem desligados até que uma mensagem específica de ativação seja re-enviada pelo nó sorvedouro.
- **Temporizador -** Tal sub-módulo pode ser usado pelos outros sub-módulos para agendar a execução de tarefas para momentos específicos. Para tal, é utilizado o relógio interno do *hardware* do nó.
- **Encaminhamento -** O sub-módulo de encaminhamento possui todas as funcionalidades necessárias para que um nó seja capaz de, após receber uma informação, decidir o destino para o qual a informação deve ser encaminhada e, posteriormente, encaminhá-la para o destino determinado. Para que não haja perda de informações recebidas, filas são implementadas nesse sub-módulo. Esse sub-módulo é responsável por tratar as transmissões e as recepções oriundas dos dois módulos de comunicação, *Intra-cluster* e *Inter-Cluster*. Esses dois módulos recebem, respectivamente, informações originadas no próprio *cluster* ou nos *clusters* vizinhos e entregam-nas para o sub-módulo de encaminhamento em questão.

  Tal sub-módulo utiliza o sub-módulo temporizador, disponível no *hardware* do nó, para programar suas tarefas de encaminhamento.

  Adicionalmente, o sub-módulo de encaminhamento, quando executado em um nó *cluster-head*, tem o papel de criar e manter uma tabela na memória contendo a energia residual de cada nó do seu *cluster*. Tal dado é obtido da recepção de cada pacote de informação proveniente de cada nó do *cluster*. Em outras palavras, cada pacote de informação deve conter um campo informando o nível de energia residual do nó remetente.

Vale ressaltar que tal sub-módulo quando executado em nós não *cluster-head*, só possui o papel de transmitir as informações coletadas pelo sub-módulo de sensoriamento para o seu *cluster-head*, através de chamadas ao módulo de comunicação *Intra-cluster*.

- **Sensoriamento -** O sub-módulo de sensoriamento é responsável por controlar o(s) sensor(es) disponíveis no *hardware* do nó. Tal tarefa inclui a ativação e desativação do(s) sensor(es), a recepção e o processamento das informações detectadas pelo(s) sensor(es), e sua entrega para o sub-módulo de encaminhamento.
- **Rotação -** O sub-módulo de Rotação é responsável por executar a rotação dos *cluster-heads* da rede. Apesar dessa rotação ser executada apenas pelos nós *cluster-head*, esse sub-módulo deve estar disponível em todos os nós da rede uma vez que a cada nova rotação, qualquer um dos nós de um determinado *cluster* pode ser eleito como sendo o novo *cluster-head*.

  Este sub-módulo é acionado periodicamente através do sub-módulo Temporizador e isso ocorre simultaneamente em todos os nós da rede. Tal sub-módulo é acionado em todos os nós, pois eles precisam ficar aguardando uma notificação oriunda do *cluster-head* corrente. Essa notificação informa quem foi eleito para ser o novo *cluster-head*.

  Adicionalmente, através de um processo similar à seleção do novo *cluster-head*, também cabe ao sub-módulo de Rotação definir quem assumirá o papel de *cluster-head* reserva. Essa seleção baseia-se na tabela de energias residuais que é mantida na memória de cada *cluster-head* pelo sub-módulo de encaminhamento.

3.2.2 Comunicador Intra-*Cluster*

Este módulo, existente em todos os nós, é responsável por implementar o protocolo de comunicação usado dentro de um mesmo *cluster*. Entretanto, o comportamento desse módulo nos nós puramente sensores é diferente do comportamento nos nós *cluster-head.*

No caso do nó ser *cluster-head*, cabe ao nó organizar a comunicação entre ele e os nós do seu *cluster* com o intuito de evitar colisões. O TDMA é um dos protocolos que pode ser adotado por esse módulo para organizar a comunicação interna do *cluster*. Nesse caso, cabe ao *cluster-head*, através do seu módulo de comunicação Intra-Cluster, definir um cronograma TDMA de transmissão interna para o *cluster*, onde não ocorram duas ou mais transmissões simultâneas dos nós puramente sensores. Tal cronograma, uma vez definido, é divulgado para

os nós do *cluster* através de uma chamada ao módulo de rádio que também está disponível em todos os nós da rede.

Por outro lado, quando o nó é apenas sensor, cabe a esse nó executar uma tarefa mais simples que a do *cluster-head*. Tal tarefa consiste em fazer com que as informações sensoriadas sejam transmitidas para o *cluster-head* no momento certo, respeitando o *slot* de tempo definido no cronograma TDMA.

Adicionalmente, para evitar interferência entre *clusters* adjacentes, o protocolo CDMA pode ser aplicado na comunicação interna de cada *cluster*.

3.2.3 Comunicador Inter-*Cluster*

Esse módulo é responsável por implementar o protocolo de comunicação usado entre *clusters*. Cabe a este módulo dar suporte à função de encaminhamento do Controlador para o caso de nós *cluster-heads*. Tal função tem como objetivo repassar informações entre *cluster-heads* de modo que as informações cheguem até o nó sorvedouro. O CSMA é um dos protocolos que pode ser adotado por este módulo para organizar a comunicação entre os *cluster-heads*, embora não seja uma solução eficiente em termos de energia [52].

Adicionalmente, são usados os protocolos CDMA e o TDMA. O protocolo CDMA pode ser utilizado para evitar interferência entre tráfegos distintos e o TDMA pode ser usado pelo *cluster-head* para determinar o *slot* de transmissão onde serão efetuadas suas transmissões de encaminhamento via CSMA, seja do cronograma do seu próprio *cluster* ou do cronograma do *cluster* pai.

Em suma, o esquema de protocolos adotados na camada MAC para regularizar as comunicações *intra-cluster* e *inter-cluster*, faz uso da associação do protocolo CDMA com o TDMA. Tal associação é também conhecida como *slotted CDMA* [36]. Adicionalmente, como mencionado, também é adotado o protocolo CSMA.

3.3 Descrição do funcionamento do protocolo proposto

Existem vários protocolos de disseminação de dados em RSSFs que, entre outras coisas, já consideram a formação de *clusters*, comunicação *multi-hop* até o sorvedouro, coleta de dados orientada a interesses e coleta periódica. Porém, o protocolo proposto introduz um esquema tolerante a falhas que faz uso de um mecanismo de *clusterização* centralizado para subdividir a rede em partes menores, onde cada parte é representada por um nó líder, chamado

de *cluster-head* (CH). Esse *cluster-head* tem o papel de centralizar a coleta de informações em uma determinada sub-região que é coberta por um subconjunto (*cluster*) dos nós da rede, assim como no protocolo LEACH, onde é adotada uma organização semelhante.

Para viabilizar a proposta deste protocolo, é utilizado um mecanismo de comunicação *multi-hop* até o sorvedouro e é necessário que um número mínimo de *clusters* seja formado, visando garantir a conectividade (*inter-cluster*) entre nós de clusters adjacentes ao longo da vida útil da rede.

Para melhor explicar o funcionamento do protocolo proposto, este foi divido em sete fases correspondendo às sete seções seguintes. Na Seção 3.3.1, é descrita a fase de configuração inicial. Na Seção 3.3.2, é descrita a metodologia adotada para que se possa garantir a conectividade entres *clusters* da rede. A Seção 3.3.3 descreve o processo de organização da rede em *clusters* (considerando o número mínimo de *clusters* necessário) e também a formação da topologia *multi-hop*. A Seção 3.3.4 descreve a fase denominada Período de Atenção do *Cluster*. Na Seção 3.3.5, é descrita a etapa de divulgação de interesses e montagem de cronogramas TDMA. Finalmente, nas Seções 3.3.6 e 3.3.7, são descritas respectivamente as fases de encaminhamento e rotação de *cluster-heads*.

3.3.1 Configuração

Inicialmente, após a instalação de cada nó sensor no ambiente de sensoriamento, inicia-se uma fase de configuração. Nessa fase, o nó sorvedouro recebe de cada um dos nós informações como, por exemplo, a sua posição geográfica (latitude e longitude), a sua quantidade de energia atual e o(s) seu(s) tipo(s) de sensor(es) disponível(is). As informações de latitude e longitude podem ser obtidas através de algum método como, por exemplo, o proposto em [53], onde a única necessidade adicional é a existência de um grupo de nós localizadores, posicionados possivelmente fora da área de monitoração e capazes de transmitir *beacons* para os nós sensores. O envio das coordenadas para o sorvedouro por parte de cada nó sensor é feito usando um método de inundação. A confirmação de recepção de tal informação por parte do sorvedouro pode ser feita diretamente para cada nó. A confirmação direta por parte do sorvedouro não geraria um grande impacto na vida útil da rede visto que o sorvedouro não é limitado em termos de energia.

3.3.2 Metodologia usada para garantia de conectividade

Em relação à conectividade entre nós de *clusters* adjacentes, existe uma questão que não poderia deixar de ser mencionada. Tal questão diz respeito ao alcance do rádio adotado nos nós sensores. Deve-se considerar a situação onde um dado *cluster-head* seja incapaz de alcançar o próximo *cluster-head* da sua rota ou ser alcançado por um de seus *cluster-heads* filhos devido ao alcance limitado do seu rádio. Essa falta de conectividade pode acontecer em duas possíveis situações. A primeira acontece logo no início da execução do protocolo, caso um *cluster-head* filho seja incapaz de transmitir dados para o seu *cluster-head* pai devido à distância ser superior ao alcance do seu rádio. A segunda situação pode ocorrer após a fase de rotação de *cluster-heads*, caso os dois *cluster-heads* recém eleitos estejam distantes demais entre si. Tal situação vale também para os nós *cluster-head* reservas eleitos pelo mecanismo de tolerância a falhas.

Dessa forma, faz-se necessário garantir que para qualquer par de nós *U* e *V*, a distância entre eles seja menor do que o alcance máximo do rádio adotado. *U* e *V*, pertencem respectivamente aos clusters *X* e *Y*, o cluster *X* sendo pai do cluster *Y*. A metodologia usada para garantir isso baseia-se no processo inicial de formação dos *clusters*, que adota o algoritmo *Simulated Annealing*[35] e tem como um de seus parâmetros de entrada o número desejado de *clusters*. Esse número deve ser previamente calculado conforme descrito a seguir.

Nos experimentos executados para esta dissertação, como será mostrado no Capítulo 4, foi adotado um rádio capaz de efetuar transmissões para um receptor localizado a uma distância A (alcance do rádio), usando o modelo de propagação implementado nas extensões ao *ns-2* feitas para o projeto μAMPS [40]. Nesse caso, cabe ao projetista da rede configurá-la de acordo com as limitações do rádio disponível, como é explicado nos próximos parágrafos.

Portanto, sendo A o alcance máximo do rádio adotado, C o número de *clusters* passado como parâmetro para o algoritmo de clusterização (*Simulated Annealing*) e L o comprimento do lado da região quadrada de monitoramento, pode-se dizer que é possível ajustar o valor de C para garantir-se a alcançabilidade citada nos parágrafos anteriores, considerando um dado valor fixado para A e L. Por exemplo, cenários onde os nós apresentam um alcance de rádio (A) muito reduzido, devem apresentar uma maior quantidade de *clusters* (C) para garantir *clusters* com menores diâmetros e conseqüentemente evitar o problema da alcançabilidade entre os nós de *clusters* adjacentes. Finalmente, o parâmetro C é passado para o algoritmo de *clusterização* no início de todo o processo. Nos próximos parágrafos, é definida uma fórmula

que permite calcular aproximadamente o número de *clusters* (C) necessários visando garantir essa conectividade.

Através de uma equação relacionando as dimensões dos *clusters* e as dimensões da área de interesse, é definida uma expressão matemática relacionando o alcance do rádio adotado e o número de *clusters* necessários para que se possa garantir que qualquer par de nós, pertencentes respectivamente a dois *clusters* adjacentes, estejam sempre alcançáveis entre si. Para reduzir a complexidade do problema, algumas simplificações foram assumidas. Na primeira, assume-se que os *clusters* possuem todos o mesmo tamanho e forma geométrica (neste caso um círculo de raio R) ocupando uniformemente a área de interesse que, por sua vez, é um quadrado de lado L. Para simplificar o desenvolvimento matemático, também foram considerados apenas cenários com um número C de *clusters*, onde *C* é um quadrado perfeito.

Como é mostrado na Figura 3, o *cluster* H possui 8 *clusters* adjacentes. Tais clusters podem ser divididos em dois tipos, **vizinhos paralelos (Ex: I)** ou **vizinhos diagonais (Ex: J)**.

C = Número de *clusters*

R = Raio do *cluster*

L = Lado da área quadrada que está sendo monitorada

A = Alcance necessário para o Rádio

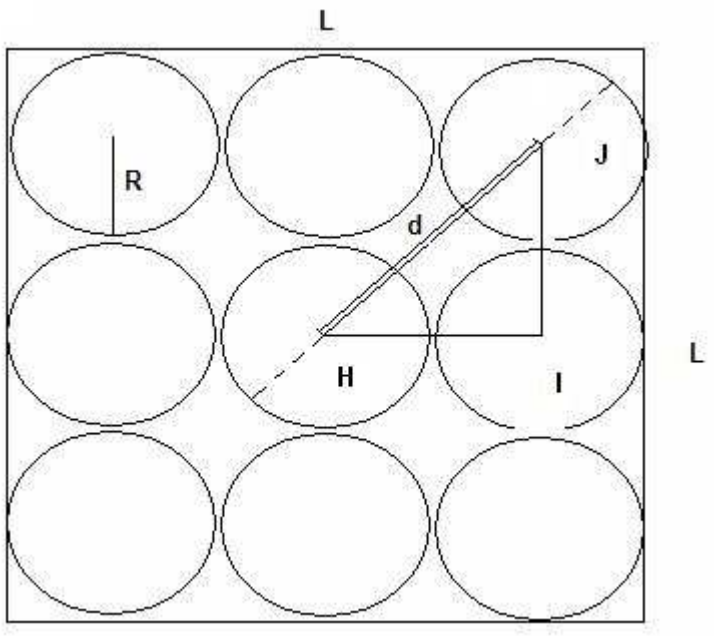

Figura 3 - Relação entre o lado (L) da área quadrada e o alcance do rádio (A)

Para obtenção da expressão considera-se apenas a distância entre vizinhos diagonais já que apresentam uma distância maior entre si, como pode ser visto na Figura 3. Portanto, já está sendo considerado o pior caso em termos de distância entre vizinhos.

Desse modo, o alcance do rádio necessário (A) para a distância de pior caso (maior possível) entre dois nós localizados respectivamente nos *clusters* H e J pode ser dado pela equação (i):

(i) $$A = d + 2R$$

Sendo que,

(ii) $$d = 2R\sqrt{2}$$ (pelo teorema de Pitágoras)

Logo:

(iii) $$R = \frac{A}{2\left(\sqrt{2}+1\right)}$$

Com isso, tem-se uma fórmula relacionando o Alcance do rádio (A) e o raio (R) do *cluster*.

Além disso, considerando que a região de monitoração é quadrada de lado L e que o número de *clusters* desejado (C) é um quadrado perfeito, pode-se dizer que:

(iv) $$L = 2R\sqrt{C}$$

Portanto, a partir das equações (iii) e (iv), obtêm-se a equação entre o Alcance do Rádio (A) e o número de *clusters* adotado:

(v) $$C = \frac{L^2}{A^2}5.8$$

Com isso, é possível garantir a alcançabilidade entre quaisquer dois nós de *clusters* adjacentes, sendo para isso necessária apenas a criação de um número de *clusters* suficiente para atender às limitações do rádio adotado. Para tal, dado o alcance do rádio previamente conhecido, e as dimensões da região, basta adotar o número de *clusters* (C) dado pela equação [v]. Além disso, como todo o processo de clusterização é centralizado, o sorvedouro pode marcar (antes de divulgar a organização em *clusters*) os nós de cada *cluster* que forem

considerados potencialmente mal localizados no que diz respeito a garantia de conectividade. Por exemplo, na aproximação feita na Figura 3, os nós de cada *cluster* que estiverem localizados fora da área circular definida na aproximação podem ser marcados como "deficientes", de modo que nunca sejam eleitos *cluster-heads*, evitando rupturas no backbone de *cluster-heads* causadas pela aproximação assumida quanto a forma (circular) dos *clusters*.

3.3.3 Organização em *Clusters* e Geração da Topologia *Multi-Hop*

Usando como parâmetros as informações de localização de cada nó e o número de *clusters* necessário (seguindo a metodologia da Seção 3.3.2), o nó sorvedouro executa o algoritmo *Simulated Annealing* [35] para definir a organização dos *clusters* da rede. Esse algoritmo seleciona os nós da rede que serão considerados como *cluster-heads* e fornece ainda os nós componentes de cada *cluster*, ou seja, para cada nó ele diz qual será seu CH imediato.

Uma vez definida a estrutura de *clusters* da rede, segue-se a etapa referente à montagem da topologia *multi-hop*, que visa garantir que todos os *clusters* formados sejam capazes de alcançar o nó sorvedouro a fim de entregar seus dados. Nessa etapa, todos os *cluster-heads* devem ser conectados ao sorvedouro através de uma topologia representada como um grafo, no qual todos os nós *cluster-heads* e o sorvedouro correspondem aos vértices. As arestas desse grafo são representadas pelas conexões lógicas existentes entre tais dispositivos. Visando diminuir o número de arestas e manter todos os nós conectados aos seus vizinhos potenciais, foi usado o algoritmo *Delaunay Triangulation* [50]. A seguir, o sorvedouro aplica o algoritmo de *Dijkstra* [54] no grafo gerado com intuito de estabelecer um caminho ótimo de cada *cluster-head* até o sorvedouro. Essa otimização é feita considerando o número de saltos entre cada CH e o sorvedouro. Ao final da execução de tal algoritmo, o sorvedouro possui uma topologia em forma de árvore para a rede.

Deve-se ressaltar que a fase de montagem dessa árvore é um processo realizado em curto espaço de tempo, considerando os tamanhos de rede estudados, conforme pode ser constatado em [37].

Após tal processo de montagem, o nó sorvedouro conhece a topologia da rede, a árvore de escoamento que interliga o sorvedouro a todos os CHs, e também a composição de cada *cluster*. Em seguida, tal informação estrutural deve ser enviada para todos os nós. A informação é enviada através da ***mensagem de Configuração***, conforme ilustrado na Figura 5(1), transmitida em *broadcast* e em alta potência. Desse modo, tanto os nós mais próximos

quanto os mais distantes podem “escutar” a mensagem. A quantidade de energia despendida em tal transmissão não é um problema já que temos um nó sorvedouro com abundância de energia. Do ponto de vista dos nós sensores, a recepção dessa mensagem também não é um problema, visto que é recebida uma única vez e envolve apenas gastos de energia com o circuito eletrônico de recepção de cada nó, que independe da distância ao transmissor, e com o processamento, que depende apenas do tamanho da mensagem que, por sua vez, é linearmente proporcional ao número de nós. Na ***mensagem de Configuração***, o nó sorvedouro envia a estrutura da rede formatada de modo que todos os nós que estão recebendo sejam capazes de descobrir se são nós comuns ou se são *cluster-heads*. No primeiro caso, o nó tem como avaliar quem é o seu CH. No segundo caso, com uma pequena filtragem da mensagem recebida, o nó tem como enumerar (i) o seu conjunto de nós filhos, (ii) o próximo CH (CH pai) no caminho até o sorvedouro e (iii) o(s) CH(s) (CHs filhos) que o têm como próximo passo da rota até o nó sorvedouro.

Portanto, em tal topologia, um determinado nó sensor sempre envia informações para o seu CH. Esse, por sua vez, pode encaminhá-las diretamente para o nó sorvedouro ou, em caso de um *cluster-head* ainda muito distante, para o próximo CH da rota até o sorvedouro.

É importante frisar uma diferença entre o protocolo proposto e o protocolo LEACH. No LEACH, todos os *cluster-heads* se comunicam diretamente com o nó sorvedouro. Em grandes redes de sensores esse tipo de comunicação consome elevadas quantidades de energia. Além disso, considerando que qualquer nó pode ser eleito *cluster-head*, o LEACH assume que todos os nós são capazes de alcançar o sorvedouro.

É bom ressaltar também que os CHs não ficam todo o tempo ligados. Na verdade, assim como os nós sensores dos seus respectivos *clusters*, os CHs possuem um acordo com o nó sorvedouro para periodicamente acordarem de forma a receber ordens. Tal acordo pode ser feito, por exemplo, a partir do recebimento da ***mensagem de Configuração*** (Figura 5(1)). Desse modo, o sorvedouro tem a garantia de saber quando os CHs estarão efetivamente ligados para que ele possa futuramente enviar uma ***mensagem de Interesse*** (Figura 5(3))**.**

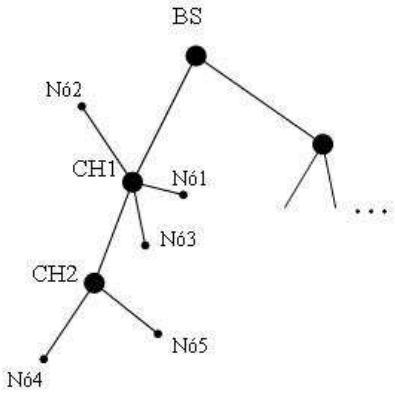

Figura 4- Exemplo de Topologia

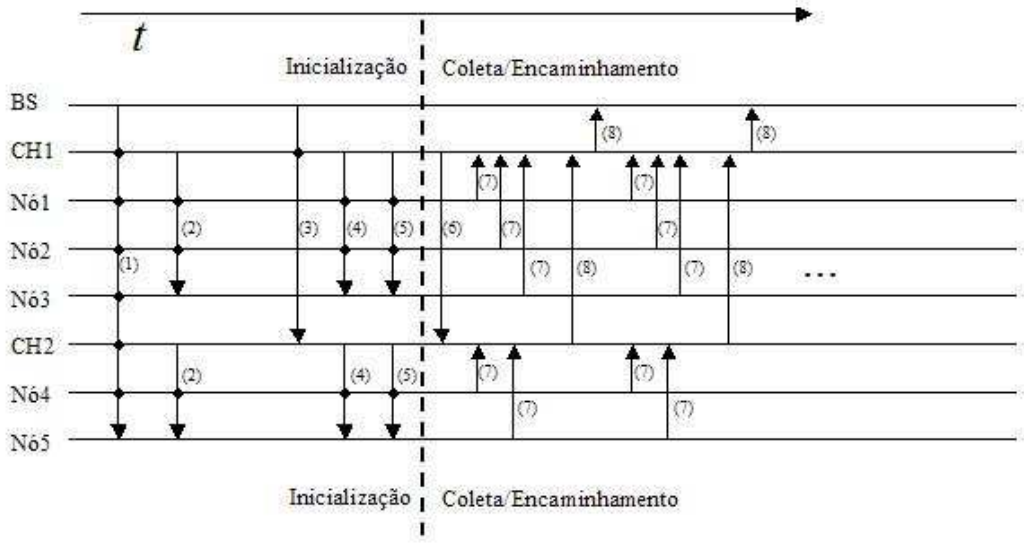

Figura 5 - Fase de Inicialização e Fase de Coleta/Encaminhamento

A Figura 5 mostra um diagrama que ilustra a seqüência de trocas de mensagens entre os nós ao longo do tempo *t*, que aumenta da esquerda para a direita. Cada linha horizontal representa um nó, conforme indicado na legenda do lado esquerdo da figura. Os envios de mensagens são representados por setas partindo do nó remetente em direção ao nó destino. Tais setas, em alguns casos, representam transmissões para múltiplos destinatários. Nesse caso, uma única seta é usada e os pontos de interseção, entre ela e as retas que representam os nós destinatários, são marcados com um pequeno circulo preto, indicando que o nó também é destinatário da mensagem. Adicionalmente, cada seta apresenta uma legenda numérica

indicando qual o código da mensagem representada, através de referências à Tabela 1. Vale ressaltar que esta descrição vale também para a Figura 6.

3.3.4 Períodos de Atenção do *Cluster*

A próxima etapa do processo de inicialização consiste no envio da ***mensagem de Aviso***, conforme ilustrado na Figura 5(2). Esse envio é feito por cada *cluster-head* para os seus filhos informando-os que se desliguem e acordem periodicamente, permanecendo acordados por um curto espaço de tempo (*período de atenção do cluster*). O objetivo desses períodos de atenção é fazer com que os nós membros tenham como receber "ordens" dos seus respectivos CH´s. Essa etapa é também chamada de *espera econômica* e o seu objetivo é manter os nós em um modo de economia de energia e ainda assim, serem capazes de receber ordens do seu CH.

3.3.5 Cronograma TDMA e Recepção de Interesses

Durante a etapa de *espera econômica*, a aplicação do usuário pode eventualmente enviar um interesse para o sorvedouro. Tal interesse contém basicamente as seguintes informações: tipo de sensor; intervalo de aquisição; duração do interesse; área geográfica de interesse e valor limite. Ao receber tal interesse, o nó sorvedouro calcula a distância entre ele e o CH mais distante que satisfaz ao interesse e prepara uma ***mensagem de Interesse*** (Figura 5(3)), para ser enviada com potência suficiente para atingir todos os CHs desejados.

Em seguida, o nó sorvedouro envia em *broadcast* uma ***mensagem de Interesse***, conforme mostrado na Figura 5(3), ordenando que os CH's iniciem suas atividades. A partir do momento em que os CHs recebem a ***mensagem de Interesse***, cada CH aguarda o próximo *período de atenção do cluster* e envia uma mensagem ***Não Durmam*** em *broadcast* para o seu *cluster*, como ilustrado na Figura 5(4), ordenando que os nós permaneçam ligados continuamente. Quando um nó sensor recebe tal mensagem, passa a ficar continuamente ativo, aguardando uma ***mensagem de Ativação*** (Figura 5(5))***,*** para efetivamente iniciar o monitoramento do meio e transmissão das informações monitoradas para o seu CH.

Existe um controle baseado em *ACKs* mantido no sorvedouro visando garantir que todos os CHs efetivamente recebam a mensagem de interesse. Quando um *cluster-head* recebe uma ***mensagem de Interesse***, ele verifica em sua tabela de nós filhos se alguns dos seus filhos enquadram-se no perfil do interesse desejado. Neste caso, o CH em questão, aguarda o próximo período de atenção do seu *cluster* e envia uma ***mensagem de Ativação*** em *broadcast* para os nós do seu *cluster*. Tal mensagem contém uma lista com os nós localizados na área

especificada no interesse recebido, assim como o respectivo *slot* de tempo que cada nó poderá usar para transmitir seus dados (cronograma TDMA), conforme ilustrado na Figura 5(5).

Na montagem do cronograma TDMA são reservados *slots* de transmissão para cada um dos nós sensores do *cluster* que atendem ao interesse recebido, assim como um *slot* para a transmissão de encaminhamento do próprio CH e um *slot* para a recepção de informações oriundas de cada um dos *cluster-heads* filhos.

Ao receber a ***mensagem de Ativação***, cada nó verifica se o seu próprio identificador se encontra na lista e, em caso positivo, o nó *entra na fase de Coleta/Encaminhamento*. A partir de tal momento, o nó deve começar a sensoriar o meio e enviar suas informações para o CH segundo a periodicidade definida no interesse recebido e também respeitando o seu *slot* de tempo para envio de dados. Em aplicações orientadas a eventos, os sensores só enviam informações quando o valor detectado é maior que o valor limite especificado. Porém, mesmo que não ocorram eventos, periodicamente são enviados pacotes vazios para manter o CH informado sobre a situação corrente de cada nó.

As transmissões internas dentro de cada *cluster* são efetuadas usando-se um código de espalhamento (CDMA) definido pelo CH, o qual é diferente do código usado em *clusters* vizinhos a fim de evitar interferência entre transmissões de *clusters* distintos. Tal esquema CDMA para transmissão é similar ao adotado pelo protocolo LEACH [27].

As transmissões entre *cluster-heads* são feitas respeitando-se o cronograma TDMA estabelecido pelo seu *cluster-head* pai, ou respeitando-se o cronograma TDMA estabelecido no próprio *cluster*. Tal escolha depende apenas da abordagem de sincronização adotada, conforme mostrado na seção 3.3.10.

3.3.6 Encaminhamento dos Dados Coletados e Agregação de Informações

Inicialmente, nessa seção, são apresentados os diferentes tipos de nós e seus mecanismos de encaminhamento. Em seguida, é descrito o funcionamento desta fase.

Considerando a comunicação *multi-hop*, adotada pelo protocolo proposto, faz-se necessário definir os mecanismos usados pelos nós para o encaminhamento da informação para seus vizinhos. Para tal, os nós assumem diferentes papéis: nó **folha**, nó **CH folha** e nó **CH interno**.

Os nós do tipo **folha** apenas enviam as suas informações coletadas para os seus respectivos nós CH; eles não executam nenhum outro mecanismo de encaminhamento. Os **nós CH folha** não possuem nós CH como filhos. Este tipo de nó tem a responsabilidade de

encaminhar as informações enviadas por seus nós filhos (do tipo **folha**) para o próximo nó CH segundo a sua tabela de encaminhamento.

Os nós do tipo **CH interno** possuem filhos que podem ser nós do tipo **folha** ou CHs. Esses CHs, por sua vez, podem ser do tipo **nó CH folha** ou **nó CH interno**. Os nós do tipo **CH interno** são responsáveis por encaminhar tanto as informações enviadas por um outro nó CH quanto as informações enviadas pelos nós do tipo **folha** do seu próprio *cluster*.

O protocolo proposto gerencia o processo de encaminhamento usando filas de saída. Existe uma fila para cada nó. Para nós do tipo **CH folha**, a fila armazena apenas pacotes recebidos dos seus nós do tipo **folha**. Para nós do tipo **CH interno**, a fila armazena tanto pacotes recebidos dos nós do tipo **folha** do seu próprio *cluster* como pacotes recebidos dos seus CHs filhos.

O funcionamento desta fase é descrito a seguir. Cada nó do *cluster* aguarda o momento certo para fazer a sua transmissão, considerando o cronograma TDMA, e envia as informações sensoriadas para o seu CH. Esse, por sua vez, tem o papel de receber tais informações e encaminhá-las através do próximo CH de sua rota até o nó sorvedouro. Tal processo de encaminhamento se repete até que as informações atinjam o sorvedouro, podendo ainda se repetir por várias vezes se considerarmos redes de nós espalhados por grandes regiões.

Vale ressaltar que o processo de encaminhamento de um determinado CH só é iniciado quando ele recebe a mensagem ***Ativação Backbone*** do seu CH pai, conforme ilustrado no diagrama da Figura 5(6). Tal mensagem é enviada por todos os CHs para os seus CHs filhos no momento em que recebem a ***mensagem de Interesse***, oriunda do nó sorvedouro. A fase de inicialização é ilustrada na Figura 5 em forma de um diagrama de troca de mensagens ao longo do tempo. Tal diagrama baseia-se na topologia ilustrada na Figura 4. Além disso, o diagrama também define a troca de mensagens na fase de coleta e encaminhamento.

Quanto à agregação de informações durante a etapa de Coleta/Encaminhamento, pode-se dizer que eventos detectados por nós vizinhos (no mesmo *cluster*) podem gerar redundâncias causadas por informações de conteúdo similar. Se fosse tirada uma fotografia da fila de um CH, poder-se-iam ver vários pacotes enfileirados originados no próprio *cluster*. Identificando-se quais pacotes na fila foram gerados localmente, o CH pode aplicar métodos de agregação (ex: média) sobre tais pacotes visando transformá-los em um único pacote, obtendo considerável ganho de energia.

A idéia do protocolo proposto é permitir que o usuário final da rede, através da submissão de interesses para o sorvedouro, possa escolher se deseja ou não que as

informações sejam agregadas e o tipo de função de agregação a ser aplicado. Entretanto, tais funções de agregação não foram implementadas.

3.3.7 Rotação dos *Cluster-heads*

Apesar do processo de formação da estrutura global da rede ser centralizado no nó sorvedouro, o protocolo possui uma etapa periódica onde é feita uma rotação dos *cluster-heads* com o intuito de distribuir uniformemente o consumo de energia entre os nós da rede.

O método adotado para rotação dos *cluster-heads* é semelhante ao utilizado pelo protocolo LEACH[27]. Porém, o conjunto de nós sensores de cada *cluster* é mantido por toda a vida da rede, alternando apenas o nó que assumirá o papel de *cluster-head*. Tal propriedade garante que o número desejado de *clusters* da rede seja sempre mantido constante. No LEACH, devido à componente aleatória que é incluída na decisão, o número de *clusters* ao longo da vida útil da rede pode variar.

O *cluster-head* e os nós de seu *cluster* combinam um instante inicial e uma periodicidade bem definida. Com essa periodicidade (TDMA), de tempos em tempos, todos os nós do *cluster* enviam a sua quantidade residual de energia juntamente com as informações coletadas para o *cluster-head* que, por sua vez, escolhe o que tem maior energia residual e o elege como novo nó *cluster-head*. O nó que acabou de deixar de ser *cluster-head,* chamado de *cluster-head anterior,* deve enviar para o novo *cluster-head* a tabela atual de encaminhamento. O *cluster-head anterior* também deve avisar a todos os *cluster-heads* que o tinham como próximo passo, que o próximo passo mudou. Uma outra obrigação do *cluster-head* que acabou de perder a liderança é enviar um aviso para o sorvedouro dizendo quem é o novo *cluster-head* do seu *cluster*. Este envio é feito usando o esquema *multi-hop* da rede e é fundamental porque o sorvedouro precisa manter uma lista de todos os *cluster-heads* da rede para poder submeter os interesses. A partir deste momento, os nós que não foram eleitos estão aguardando um aviso do novo *cluster-head* contendo, inclusive, o novo cronograma TDMA do *cluster*.

Além da tabela de encaminhamento, o *cluster-head* que acabou de perder a liderança para outro nó do *cluster* também tem a obrigação de enviar a sua tabela de interesses corrente.

Um parâmetro de configuração da fase de rotação que merece destaque é o intervalo entre as fases de rotação. Possíveis valores para tal parâmetro são mostrados no Capítulo 4.

Como dito anteriormente, a execução da fase de rotação é iniciada simultaneamente em todos os nós, sejam *cluster-heads* ou nós puramente sensores do *cluster* em questão. Quando

o nó em questão não é um *cluster-head*, o seu papel inicial nesta fase é simplesmente ficar aguardando a mensagem ***Aviso novo ch*** (Figura 6(11)), informando quem é o novo *cluster-head* do seu *cluster*. Tal aviso pode, inclusive, informar que o novo *cluster-head* é o próprio nó que está recebendo a mensagem. Neste caso, o aviso é considerado uma notificação de que o nó deve assumir a posição de novo CH do seu *cluster*.

Para efetivamente implementar o processo de rotação, o *cluster-head* deve avaliar a sua tabela de energias residuais e escolher o nó com maior energia. Em seguida, tal *cluster-head* deve mandar três mensagens. A primeira, ***Aviso Novo CH Pai***, é enviada conforme ilustrado na Figura 6(9), para os seus *cluster-heads* filhos contendo o identificador do novo *cluster-head* que eles devem passar a considerar como pai. Em seguida, o *cluster-head* entra em um estado de espera, aguardando a recepção dessa mesma mensagem (***Aviso Novo CH Pai***) oriunda de seu *cluster-head* pai. Após a recepção de tal mensagem, a segunda mensagem, ***Aviso Novo CH Filho*** (Figura 6(10)), é enviada por cada *cluster-head* ao seu *cluster-head* pai, informando-o quem ele elegeu para novo CH. O envio dessa segunda mensagem é necessário para que o *cluster-head* pai possa incluir o *cluster-head* eleito em seu conjunto de *cluster-heads* filhos, para manter a estrutura em árvore de *clusters* conectada. A terceira mensagem, chamada de ***Aviso Novo CH*** (Figura 6(11)), é enviada em *broadcast* para os membros do seu próprio *cluster,* informando a todos quem é o CH que estará substituindo-o. Quando cada membro do *cluster* recebe tal mensagem, podem ocorrer duas situações. Na primeira, o nó simplesmente passa a conhecer quem é o novo CH de seu *cluster* e envia uma mensagem ***Ping CH*** (Figura 6(13)) para ele. Na segunda, o nó que está recebendo a mensagem é o novo CH. Neste caso, já assumindo o papel de novo CH do seu *cluster*, o nó entra em um estado onde fica aguardando mensagens ***Ping CH*** dos seus filhos. Cada mensagem ***Ping CH*** recebida caracteriza a inclusão do nó remetente no *cluster*.

O diagrama de mensagens trocadas entre os nós ao longo do tempo durante a fase de rotação está ilustrado na Figura 6. Além disso, tal diagrama se baseia na topologia exposta na Figura 4.

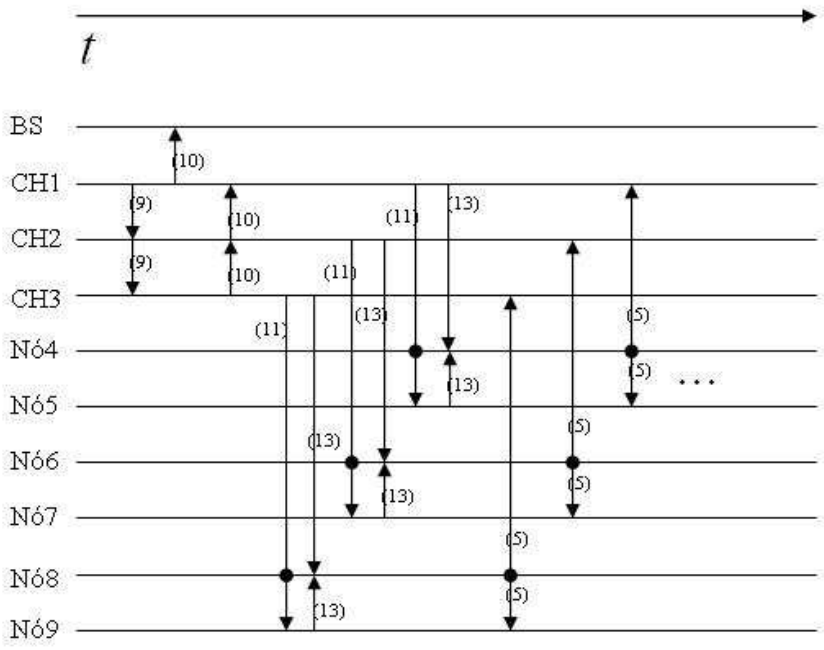

Figura 6 - Troca de Mensagens (Fase de Rotação)

A partir deste momento, a fase de rotação está completa e cada nó puramente sensor já sabe quem é o seu novo *cluster-head*. Tal *cluster-head*, por sua vez, já sabe quem são seus novos CHs filhos, quem é o seu novo CH pai, bem como quem são os nós sensores do seu *cluster*. Adicionalmente, transmissões interrompidas de sensores ativados durante essa fase de rotação são imediatamente retomadas ao fim da fase. Em outras palavras, os CHs passam a receber novamente mensagens ***Dado*** (Figura 5(7)) dos nós sensores do seu *cluster*, e encaminhá-las rumo ao nó sorvedouro através de mensagens ***Dado BackBone*** (Figura 5(8)) enviadas para os seus respectivos CHs pai.

A Tabela 1**,** a seguir, enumera os tipos de mensagens do protocolo de roteamento proposto que são trocadas entre os nós de uma rede. Além disso, é mostrado para cada tipo de mensagem, o tipo de remetente e o tipo de destinatário que se aplica. Adicionalmente, a Figura 7 mostra uma topologia antes da fase de rotação, e a Figura 8 apresenta a mesma topologia após a execução da fase de rotação. Nesse caso, os *cluster-heads* que eram os nós 1, 2 e 3, elegem respectivamente como novos *cluster-heads* os nós 4, 6 e 8.

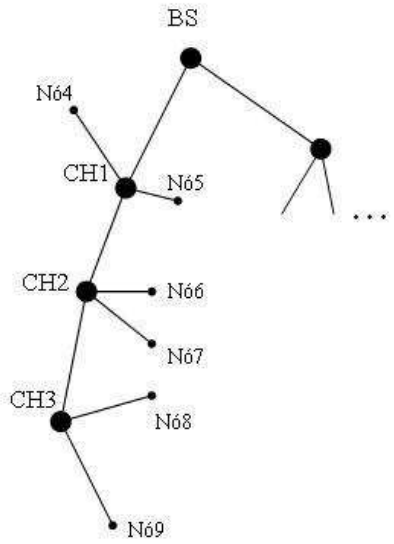

Figura 7 – Topologia Antes da Rotação

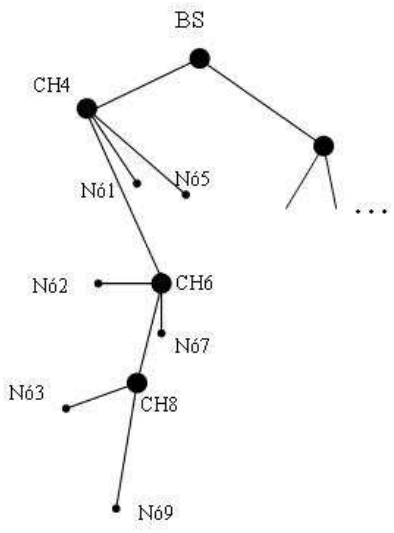

Figura 8 – Topologia Depois da rotação

Em relação as mensagens enumeradas na Tabela 1, é possível dividi-las em dois tipos. O primeiro tipo, incluindo a mensagem de código 7 (mensagem ***Dado***) e a mensagem de código 8 (mensagem ***Dado Backbone***), consiste nas mensagens para tráfego de informações coletadas propriamente ditas. O segundo tipo, incluindo todas as outras mensagens, consiste nas mensagens de controle.

Para simplificar a implementação das simulações e para tornar a comparação mais justa criando mensagens de tamanho aproximadamente igual ao dos protocolos comparados (ex: *Directed Diffusion*), foi adotado um tamanho fixo de 64 *bytes* para as mensagens do primeiro tipo e um tamanho variável para as mensagens de controle dependendo do código da mensagem. Conforme ilustra a Figura 9, em todas as mensagens o primeiro byte é reservado para o código da mensagem em questão sendo seguido de informações variáveis dependendo do da mensagem. Vale ressaltar que, em tal figura, as partes de cada mensagem ressaltadas com um preenchimento mais escuro simbolizam a ausência de conteúdo (servindo para completar os 60 bytes mínimos de tamanho de pacote) existindo alguns casos de mensagens de controle que possuem apenas o código da própria mensagem sem nenhum conteúdo.

Código (1 *byte*)	**Conteúdo da Mensagem**	
1	**[Definição da topologia - N bytes - (N proporcional ao tamanho da rede)]**	
2		
3	**Código CH destino(2 bytes)**	**Interesse (16 bytes – 8 bytes por coordenada)**
4		
5	**Cronograma TDMA- 2 x N *bytes* (N = número de nós do *cluster*)**	
6		
7	**Dado (4 bytes)**	**Energia Residual (4 *bytes*)**
8	**Dado (4 bytes)**	
9	**Código novo CH pai(2 bytes)**	
10	**Código novo CH filho(2 bytes)**	
11	**Código novo CH (2 bytes)**	
12		
13		
14	**Código CH pai(2 bytes)**	**Códigos CHs filhos(2 x N bytes, N = número de CHs filhos)**

Figura 9 - Formato e Conteúdo das Mensagens

Tabela 1 - Tipos de mensagens trocadas entre os nós

Código	Mensagem	Origem	Destino
1	***Mensagem de Configuração***	Sorvedouro	Todos os nós
2	***Mensagem de Aviso***	CH	Nós sensores
3	***Mensagem de Interesse***	Sorvedouro	CH
4	***Não Durmam***	CH	Nós sensores
5	***Mensagem de Ativação***	CH	Nós sensores
6	***Ativação Backbone***	CH	CH filho
7	***Dado***	Nós Sensores	CH
8	***Dado BackBone***	CH	CH pai ou Sorve.
9	***Aviso Novo Ch Pai***	CH	CH filho
10	***Aviso novo Ch filho***	CH filho	CH pai
11	***Aviso novo Ch***	CH	Nós sensores
12	***Reservado para uso futuro***		
13	***Ping Ch***	Nós Sensores	CH
14	***Info Reserva***	CH	CH Reserva

3.3.8 - Adaptação para Múltiplos sorvedouros

Na etapa de inicialização do protocolo proposto, o nó sorvedouro, munido das posições geográficas de cada nó da rede, tem o papel de montar uma topologia inicial e depois divulgar tal topologia. Para viabilizar o envio de informações para múltiplos sorvedouros faz-se necessário efetuar algumas alterações nessa etapa, que inicialmente considerava apenas um sorvedouro.

Para tal, faz-se com que cada um dos sorvedouros envie um sinal em alta potência para atingir todos os nós da rede. Portanto, cada nó sensor da rede recebe N sinais, sendo N o número de *sorvedouros* da rede. Em seguida, o nó sensor deve escolher o *sorvedouro* cujo sinal chegou com maior potência, ou seja, o *sorvedouro* mais próximo considerando que todos os *sorvedouros* enviaram o *beacon* usando a mesma potência.

Após os sensores terem escolhido seu sorvedouro e esse tomar ciência disso, inicia-se a fase de inicialização do protocolo, onde cada nó sensor envia as suas coordenadas para o seu respectivo sorvedouro através de *flooding*. A única diferença em relação à proposta original do protocolo, é que os nós incluem no pacote o sorvedouro escolhido além das próprias coordenadas. Ao final desta fase de *flooding*, todos os sorvedouros terão recebido as coordenadas de todos os nós da rede, podendo selecionar apenas aquelas que eram efetivamente destinadas a eles.

A partir desse momento, pode-se dizer que a rede está devidamente particionada e que cada sorvedouro pode executar a montagem e divulgação da topologia de sua sub-região segundo a abordagem padrão, isto é, segundo a abordagem que usa um único sorvedouro.

3.4 Aspectos específicos do protocolo proposto

Nesta seção são descritos aspectos específicos do protocolo proposto. Dentre tais aspectos podem-se destacar a solução de tolerância a falhas adotada (Seção 3.4.1), o mecanismo de atribuição de códigos CDMA para os nós da rede (Seção 3.4.2) e os possíveis métodos de sincronização entre as transmissões de encaminhamento efetuadas pelos *cluster-heads* (Seção 3.4.3).

3.4.1 Mecanismo de Tolerância a Falhas

Para avaliar plenamente o desempenho de um protocolo para redes de sensores sem fio, deve-se considerar a possibilidade de falhas nos nós. Como já foi dito, tais falhas podem ocorrer de maneira repentina, devido a causas externas, ou por esgotamento de energia, levando um ou mais nós ao fim de sua vida útil. Os nós que falham, repentinamente, sejam eles próximos entre si ou em posições completamente aleatórias, podem ocasionar desde pequenas perdas de dados coletados até a imprecisão na informação fornecida pela rede. Essas falhas podem provocar também situações de partição da rede, caracterizadas pela existência de regiões descobertas.

Para o protocolo hierárquico proposto são considerados dois tipos principais de falhas nos nós. No primeiro tipo, a falha ocorre em um determinado nó que esteja desempenhando apenas o papel de nó sensor. No segundo, as falhas ocorrem nos nós que estão desempenhando o papel de *cluster-head* de um determinado *cluster*. Pode-se afirmar que as falhas do primeiro tipo não devem apresentar impactos significativos no desempenho do protocolo, exceto na questão da precisão da informação, pois tal tipo de falha pode levar uma região a ter uma redução de cobertura. Nesse caso, como exemplo, um objeto móvel cujo movimento estivesse sendo monitorado pela rede, encontrar-se-ia fora de alcance quando este passasse pela região de abrangência do nó que falhou. Entretanto, quando um *cluster-head* falha, toda a região coberta por esse *cluster* fica descoberta e, conseqüentemente, para agravar a situação, todas as regiões cobertas por seus *clusters* filhos também ficam descobertas.

Considerando os tipos de falha citados, pode-se afirmar que o de maior impacto é aquele que apresenta falhas em nós desempenhando o papel de *cluster-head*.

Com o intuito de contornar ou mesmo solucionar os problemas resultantes desse tipo de falha, pode-se fazer uso de alguma redundância em termos de nós. Para tal, é criado um novo papel na rede. Trata-se do *nó reserva*, responsável por desempenhar as funções de *cluster-head* quando o *cluster-head* titular falha. Com a criação desse novo papel, cada *cluster-head*, após a primeira rotação, elege um nó reserva para o seu *cluster*. Tal nó é escolhido dentre todos os nós sensores do *cluster* segundo um dos critérios explicados a seguir nessa seção. Esse nó fica a maior parte do tempo dormindo e acordando apenas quando o *cluster-head* titular do seu *cluster* transmite. Portanto, quando o *nó reserva* percebe que o *cluster-head* titular não está transmitindo, simplesmente assume o papel de *cluster-head* do seu *cluster*.

Uma das possíveis maneiras de se selecionar o nó reserva é o método onde o *cluster-head* simplesmente sorteia um dos nós do seu *cluster* para desempenhar tal papel, não levando em conta nenhuma informação dos sensores como, por exemplo, sua energia residual. A adoção de um nó reserva para solucionar problemas de falhas em redes não é uma novidade. Entretanto, o critério de seleção desse nó reserva proposto nesse trabalho é diferenciado e é descrito a seguir.

Para o critério de seleção do nó reserva, é calculada a média M da energia dos nós do *cluster*. Em seguida, a lista de candidatos a *nó reserva* é formada pelos nós que possuem um nível de energia superior a M. Da lista de candidatos, é escolhido para *nó reserva* aquele que apresentar o menor nível de energia dessa lista. Esse critério é útil, pois permite selecionar para *nó reserva* o nó com menor nível de energia dentre aqueles que ainda possuem quantidade razoável de energia caso precisem assumir o papel de *cluster-head*. Tal escolha é efetuada tendo em vista que o papel de nó reserva consome pouca energia em comparação ao papel de *cluster-head*, por exemplo.

Esse critério de seleção de nó reserva proposto é disparado na fase de rotação dos *cluster-heads*. Em cada fase de rotação, os *cluster-heads* titulares elegem quem será o próximo *cluster-head* e quem será o próximo *cluster-head* reserva. Para tal, o processo de seleção de ambos os nós, *cluster-head* e seu nó reserva, faz uso da tabela de energias disponível no módulo Controlador, sub-módulo de Encaminhamento, conforme explicado na Seção 3.2.

O nó reserva é notificado através da própria ***Mensagem de Ativação*** enviada pelo seu *cluster-head* em *broadcast* para todo o *cluster*. Para tal, é ativado nessa mensagem apenas um *bit* de sinalização do campo referente ao nó escolhido. Assim, tal *bit* é ignorado por todos os nós do *cluster*, exceto pelo nó efetivamente escolhido. A partir desse momento, o nó já sabe que foi eleito como *cluster-head* reserva.

Adicionalmente, uma outra mensagem (***Info Reserva)*** é enviada para o novo nó reserva pelo *cluster-head* titular. Tal mensagem contém informações que serão necessárias caso o nó reserva precise assumir o papel de *cluster-head* titular. Entre tais informações, destaca-se a de estabelecimento de sincronização entre o nó reserva e o *cluster-head* titular, permitindo, assim, o monitoramento da atividade do *cluster-head* titular para que em caso de sua falha, o *cluster-head* reserva possa assumir a liderança do seu *cluster* de maneira transparente para os outros nós membros do *cluster*.

Outras informações de destaque são os dados contidos na mensagem ***Info Reserva*** relativos ao mecanismo de roteamento. Em outras palavras, tal mensagem também informa ao

nó reserva quem é seu *cluster-head* pai e quem são os *cluster-heads* filhos. O código CDMA adotado pelo *cluster-head* pai também é informado pela mensagem. Desse modo, o nó reserva é capaz de efetivamente escutar as transmissões feitas pelo seu *cluster-head* destinadas ao *cluster-head* pai, já que tais transmissões são codificadas usando tal código CDMA.

Caso o nó reserva tenha que assumir a liderança do *cluster*, não ocorrerá uma partição da rede pois ele saberá quem é o seu *cluster-head* pai e quem são seus *cluster-heads* filhos. Além disso, esse nó reserva mudará o seu código CDMA para o código que estava sendo usado pelo *cluster-head* que falhou. Isto é necessário para que o nó reserva seja capaz de receber as informações que forem encaminhadas pelos seus *cluster-heads* filhos e pelos nós sensores do seu próprio *cluster*.

3.4.2 Utilização do protocolo CDMA

Considerando limitações no *hardware* adotado nos nós sensores, que possuem um rádio *half-duplex* conforme descrito na próxima seção, para um determinado nó ser capaz de filtrar mensagens codificadas segundo um único código CDMA por vez, faz-se necessária a definição de algumas regras para que não ocorram casos onde um ou mais nós fiquem incomunicáveis. Tais casos poderiam ocorrer, por exemplo, se um determinado nó estivesse aguardando uma mensagem codificada segundo um código X, e o remetente de tal mensagem desconhecesse o valor de X ou simplesmente adotasse um valor errado na codificação.

Portanto, inicialmente, após a instalação dos nós no campo de sensoriamento, todos os nós sensores iniciam a sua operação configurando a sua interface de rede para admitir a recepção de informações segundo um código CDMA padrão, igual para todos e pré-definido. A partir desse momento, todos os nós passam a ser acessíveis pelo sorvedouro, viabilizando a transmissão da ***Mensagem de Configuração***, por parte do nó sorvedouro, que define a formação dos *clusters*. Após o recebimento de tal mensagem, cada nó conhecerá o seu papel na rede, ou seja, se é um nó sensor comum ou se é um CH.

No primeiro caso, quando o nó é um sensor, o nó também passa a conhecer quem é o seu *cluster-head*. Em seguida, esse nó sensor deve configurar a sua interface para admitir mensagens codificadas através do código CDMA *broadcast* do seu *cluster*, que é definido a partir do identificador do seu *cluster-head*. Desse modo, todos os nós de um mesmo *cluster* usam o mesmo código CDMA para recepção, viabilizando o envio de mensagens por parte do CH para todo o *cluster* como se houvesse uma espécie de canal *broadcast* interno por *cluster*.

No segundo caso, quando o nó em questão é um *cluster-head*, o código CDMA usado para recepção é definido a partir do seu próprio identificador. Com isso, todos os nós de um determinado *cluster*, incluindo o *cluster-head*, adotam um mesmo código CDMA para recepção, reforçando o conceito de canal *broadcast* interno do *cluster*.

A comunicação *inter-cluster* é viabilizada porque cada CH está usando um código CDMA distinto para recepção baseado em seu próprio identificador, como já foi dito. Além disso, cada CH conhece o identificador do seu CH pai, quando da recepção da ***Mensagem de Configuração.***

A cada fase de rotação, os códigos CDMA adotados por cada nó da rede mudam, visto que ao final de cada fase de rotação, um novo CH assume o papel de líder e, como já foi dito, o código CDMA de um *cluster* baseia-se no identificador do seu *cluster-head*.

Para o caso dos nós puramente sensores, pode-se dizer que eles também mudam seu código CDMA a cada fase de rotação. Tal mudança é feita após a recepção da mensagem ***Aviso Novo Ch***, enviada pelo seu *cluster-head*, no fim da fase de rotação. Neste momento, o nó puramente sensor muda o seu código CDMA de recepção de acordo com o identificador do novo líder, mantendo a sua conectividade dentro do *cluster*.

Em relação ao nó tipo *cluster-head*, após eleger o novo líder e divulgar tal informação para os seus CH´s filhos (através da mensagem ***Aviso Novo Ch Pai***), para o seu CH pai (através da mensagem ***Aviso Novo Ch Filho***) e para os nós do seu *cluster* (através da mensagem ***Aviso Novo CH***), ele muda o seu código CDMA para o código baseado no identificador do novo líder. A partir deste momento, ele tem sua conectividade garantida dentro do seu *cluster* através do novo *cluster-head*, recém eleito. Isto pode ser afirmado pois ele conhece o código do novo *cluster-head*, estando apto a transmitir informações para tal *cluster-head*. Além disso, o código de recepção adotado por ele faz com que esteja conectado ao canal interno de *broadcast* do *cluster*, tornando-o apto a receber informações destinadas ao seu *cluster*.

3.4.3 Limitações de *Hardware* e Mecanismos de Sincronização

O nó *cluster-head* em um *cluster* de um protocolo hierárquico como o proposto nesta dissertação possui um subconjunto de tarefas que podem ser executadas simultaneamente, tais como: (i) a recepção das informações coletadas pelos nós membros de seu *cluster*, (ii) a recepção das informações que estão sendo encaminhadas pelos *cluster-heads* filhos e (iii) o envio de todas as informações obtidas nos itens (i) e (ii) para o *cluster-head* pai. Assim, um

cluster-head executa tanto tarefas de transmissão para o seu *cluster-head* pai como tarefas de recepção de informações dos seus *cluster-heads* filhos e dos seus nós sensores filhos.

Situações de recepções e transmissões simultâneas provocam a perda de informações, pois o rádio dos sensores é *half-duplex* e, portanto, incapaz de receber informações durante uma transmissão. O mecanismo de recepção dos nós, sejam eles apenas sensores ou *cluster-heads*, permite a recepção de informações codificadas segundo um único código (CDMA), como já foi descrito. Esse mecanismo de recepção é desabilitado toda vez que uma transmissão é iniciada.

A limitação de *hardware* descrita pode ser contornada através da adoção de estratégias de sincronização entre os nós da rede. Esta seção apresenta três alternativas para a sincronização entre transmissões e recepções na rede, a fim de contornar a limitação física do rádio usado pelos sensores.

A primeira alternativa é apresentada apenas para efeito ilustrativo. Considerando que se trata de uma opção que demanda a existência de uma sincronização única entre todos os nós da rede, trata-se de uma opção pouco viável, tendo em vista as características das RSSFs já mencionadas. Tal alternativa de sincronização visa alinhar todos os cronogramas TDMA de todos os *clusters* de modo que não ocorra, em hipótese alguma, a situação onde um *cluster-head* transmite para um segundo *cluster-head* exatamente no momento em que esse segundo *cluster-head* está fazendo a sua própria transmissão de encaminhamento. A Figura 10 ilustra o alinhamento entre todos os cronogramas. Nessa figura, são mostrados os cronogramas TDMA de cada *cluster-head* da árvore de CHs onde são reservados *slots* para transmissão (TX) do próprio CH, recepção de informações originadas em cada um dos *cluster-heads* filhos (RX CHn) e recepção de informações originadas em cada um dos nós do seu próprio *cluster* (RXn).

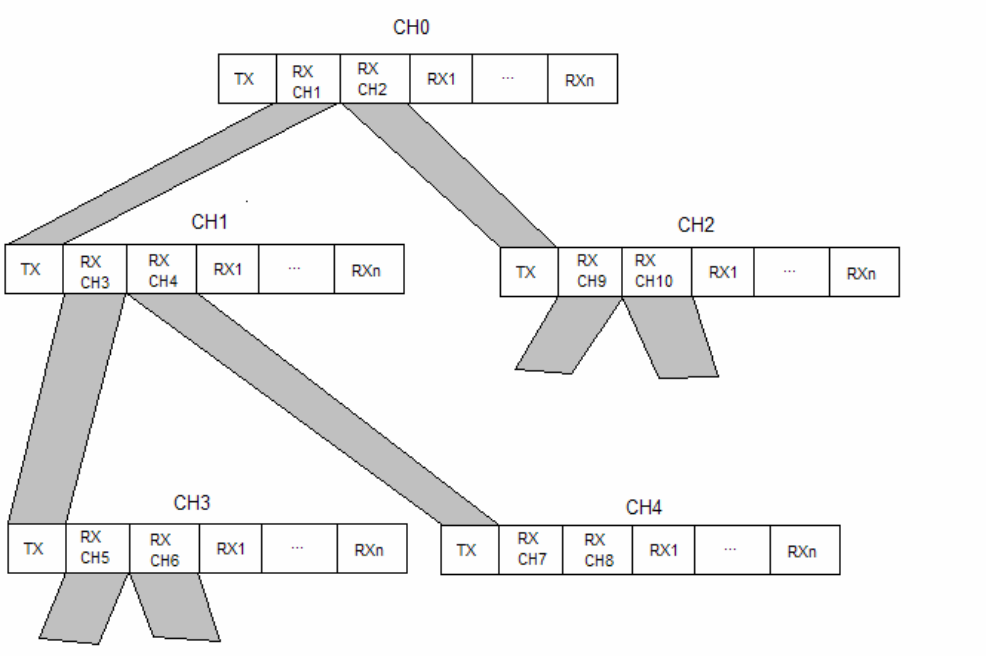

Figura 10 - Sincronização global entre todos os nós da rede

A segunda abordagem, como mostra a Figura 11, seguindo a mesma nomenclatura da Figura 10,trata da sincronização do *cluster-head* com seu próprio *cluster*. Diferentemente das soluções propostas em outros trabalhos [34,51], onde é imprescindível a existência de uma sincronização global, esse método utiliza um esquema de cronograma TDMA independente para cada *cluster*. Tal cronograma é definido pelos *cluster-heads* e não leva em consideração nenhuma informação externa ao *cluster*. Cada *cluster-head* cria um cronograma onde é reservado um *slot* para a sua transmissão de encaminhamento para o *cluster-head* pai, um *slot* para a recepção das informações de cada um dos nós do seu *cluster* e, finalmente, um *slot* para a recepção de informações de cada *cluster-head* filho.

Esse método requer sincronização local apenas entre os nós do próprio *cluster*. Dessa forma, a transmissão de informações efetuada pelo *cluster-head* é realizada na hora programada, ignorando o cronograma TDMA no seu *cluster* pai. No momento dessa transmissão, pode acontecer uma das situações descritas a seguir.

a) O *slot* alocado para a transmissão de informações do *cluster-head* para seu *cluster-head* pai (no cronograma do *cluster* em questão) coincide com o *slot* reservado para a transmissão do *cluster-head* pai (no cronograma do *cluster* pai). Nesse caso, a informação é perdida devido à limitação física já mencionada.

b) O *slot* alocado para a transmissão de informações do *cluster-head* para seu *cluster-head* pai (cronograma do *cluster* em questão) coincide exatamente com o *slot* de tempo reservado pelo *cluster-head* pai (cronograma do *cluster* pai) para a recepção dessa transmissão. Nesse caso, ocorrerá uma transmissão bem sucedida, já que tanto o *cluster-head* pai como qualquer nó do seu *cluster* certamente não estarão transmitindo. Este é o caso ideal.

c) A transmissão de informações do *cluster-head* para o *cluster-head* pai (no cronograma do *cluster* em questão) pode não coincidir nem com o caso (a) e nem com (b). Nesse caso, a transmissão coincide com um *slot* reservado pelo *cluster-head* pai (no cronograma do *cluster* pai) para a transmissão de um dos nós sensores que estão operando no *cluster* pai. Portanto, considerando que ambas as transmissões tem como destino o *cluster-head* pai e, conseqüentemente, estão usando uma mesma codificação CDMA, ocorre interferência entre elas. Podendo ser tratada, por exemplo, como uma disputa ao meio no âmbito do protocolo CSMA, se tal protocolo estiver sendo usado.

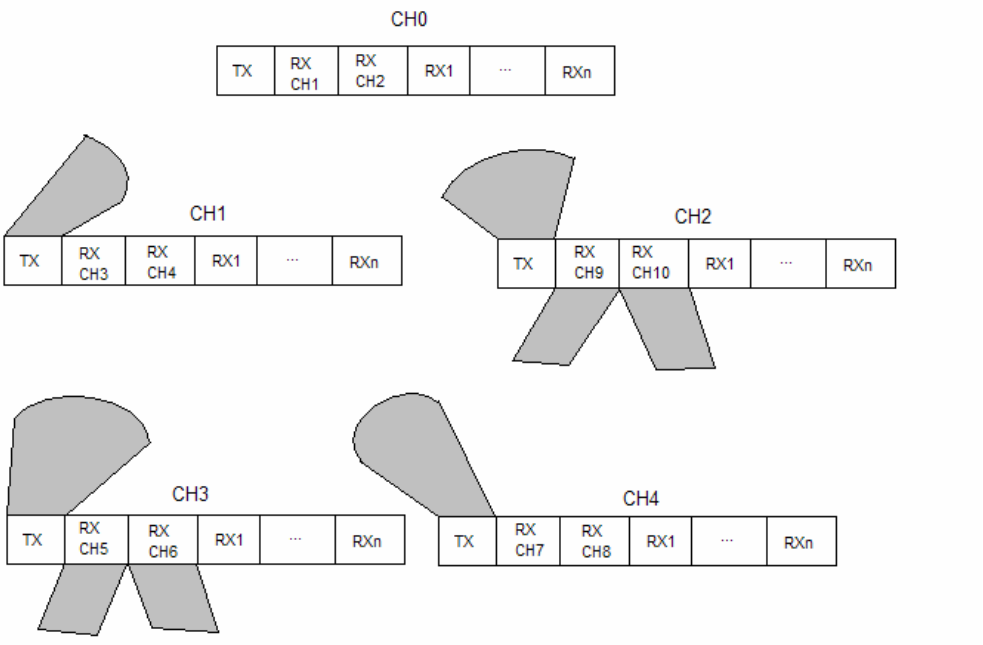

Figura 11 - Sincronização com o próprio *Cluster*

A terceira abordagem, como mostra a Figura 12, trata da sincronização do *cluster-head* com o *cluster-head* pai. Nessa abordagem, os nós de um determinado *cluster* precisam estar sincronizados apenas entre si e esse sincronismo pode ser estabelecido pelo seu *cluster-head*, da mesma maneira como é realizado no método anterior. Porém, apesar de o *cluster-head* estabelecer o sincronismo interno do seu *cluster*, ele não leva em conta as transmissões

internas do seu *cluster* para definir o momento de efetuar a sua transmissão de encaminhamento. Ao invés disso, o *cluster-head* sincroniza a sua transmissão de encaminhamento com o *cluster-head* pai. Tal procedimento é efetuado de forma que o *cluster-head* em questão seja capaz de encaminhar as suas informações para o seu *cluster-head* pai exatamente no *slot* que foi reservado para esse propósito pelo *cluster-head* pai, não importando o que esteja acontecendo no interior do seu próprio *cluster*.

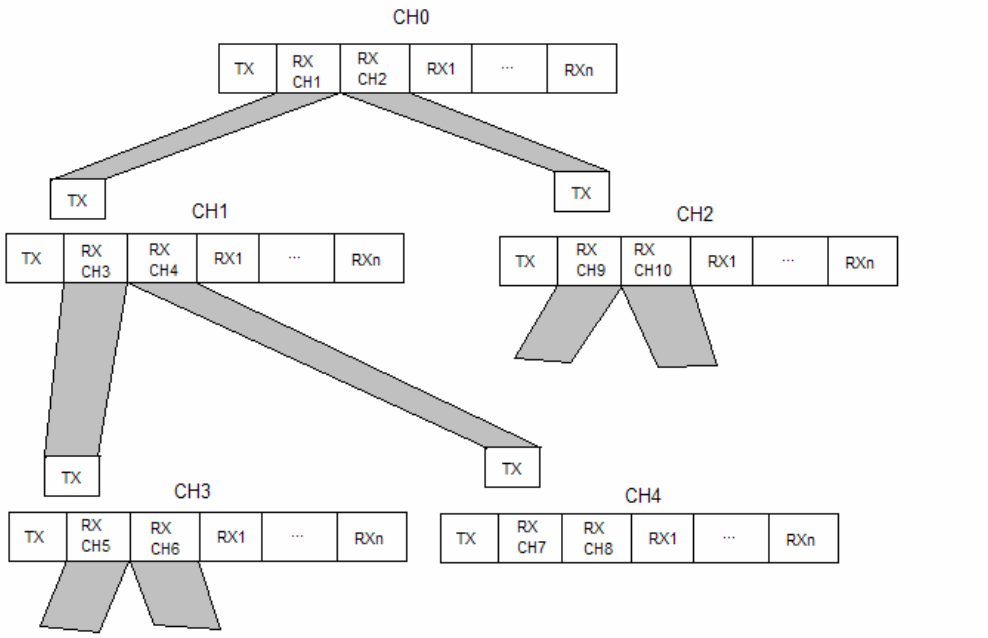

Figura 12 - Sincronização com o *Cluster* pai

Através desse método, pode-se garantir que um determinado *cluster-head* não tentará encaminhar informações para um segundo *cluster-head* durante a fase de transmissão desse segundo *cluster-head* .

Quanto à transmissão dos nós puramente sensores para seus respectivos *cluster-heads*, usando esse método, podem ocorrer apenas duas situações**:** (i) a transmissão ocorre durante uma transmissão de encaminhamento do *cluster-head* em questão para o *cluster-head* pai, havendo uma perda de informação devido à limitação física mencionada; ou (ii) a transmissão de encaminhamento não atrapalha a recepção da informação oriunda de um nó sensor, ou seja, a situação (i) não ocorre.

3.5 Considerações Finais

Neste capítulo, o protocolo proposto foi detalhadamente descrito. Além disso, questões como métodos alternativos de sincronização e detalhes sobre a utilização do protocolo CDMA foram enunciadas e esclarecidas.

Quanto às principais contribuições do protocolo proposto, pode-se destacar a topologia hierárquica diferenciada adotada, o esquema híbrido de coleta de dados e o método diferenciado para seleção do *cluster-head* reserva. A topologia diferenciada, hierárquica e baseada em *clusters* é caracterizada pela formação de um backbone de *cluster-heads*, viabilizando uma comunicação de curto alcance em múltiplos saltos. Com isso, é possível eliminar o vínculo existente (no protocolo LEACH) entre o diâmetro máximo da rede e o alcance do rádio disponível nos nós.

A abordagem híbrida de coleta de informações permite ao protocolo proposto operar tanto em modo orientado a interesses quanto em modo periódico, aumentando a gama de possíveis aplicações.

Finalmente, a criação do nó reserva provê um mecanismo de tolerância a falhas através de um método de eleição diferenciado, levando em conta o consumo energia para prolongar a vida útil da rede.

Capítulo 4 Simulações e Análise dos Resultados

Este capítulo apresenta os principais resultados de simulações realizadas com o objetivo de comprovar a eficácia do protocolo proposto nesta dissertação.

A próxima seção, 4.1, descreve a ferramenta de simulação *ns-2* [39] usada para verificar o correto funcionamento do protocolo proposto. Esta ferramenta foi empregada em todas as simulações descritas neste capítulo. Adicionalmente, são enumerados e descritos os componentes do simulador que foram usados ou alterados para acomodar de maneira realista as simulações em questão. A seção 4.2 define as métricas usadas nos diferentes cenários. A seção 4.3 apresenta os diferentes cenários de rede utilizados na simulação. A seção 4.4 apresenta a análise dos principais resultados obtidos nas simulações propriamente ditas. A seção 4.5 finaliza o capítulo, destacando seus pontos de maior relevância.

4.1 Ambiente de Simulação

O *ns-2* [39] é um simulador de redes orientado a objetos, baseado em eventos discretos. Esse simulador foi desenvolvido em duas linguagens: C++ e *OTcl*.

O simulador é formado por vários componentes escritos em C++, os quais podem ser combinados para formar topologias de rede específicas. Os componentes incluem, entre outros, nós, aplicações, agentes, protocolos de roteamento, filas, enlaces, pacotes e temporizadores.

A linguagem OTcl é usada como *interface* para a criação de cenários de simulação. Para cada classe C++, existe uma classe correspondente em *OTcl*. Assim, aquele que pretende executar uma simulação, usa comandos *OTcl* para combinar os componentes desejados e formar o seu *script* de simulação. A versão 2.1b9a do *ns-2* foi utilizada no desenvolvimento dos *scripts* de simulação. Além disso, extensões [40] ao *ns-2* feitas no MIT para o projeto *μAMPS* foram incorporadas e, por sua vez, fazem uso das extensões [41] feitas ao *ns-2* pela *Carnegie Mellon University*. Tais extensões ao núcleo do simulador incluem suporte a nós móveis, camadas MAC e modelos de propagação no canal.

A Figura 13 mostra o modelo de um nó padrão que foi adotado no ambiente de simulação. Nesse modelo, a Aplicação tem o papel de criar pacotes de dados e os enviar para um Agente que pode efetuar funções da camada de rede ou transporte e enviar a informação para o Coletor de Estatísticas. Esse elemento da pilha tem o papel de coletar estatísticas sobre os pacotes que são então enviados e recebidos. Desse modo, o Coletor repassa o pacote para o

Conector que, por sua vez, repassa-o para a Camada de Enlace para os processamentos peculiares a esta camada. Após um pequeno intervalo de tempo, a informação é repassada para a Fila onde permanece aguardando a sua vez de ser transmitida. Quando um pacote é removido da Fila, é enviado para a camada MAC, onde protocolos de acesso ao meio são executados. Finalmente, o pacote é enviado para a *Interface* de Rede, onde a potência correta de transmissão é calculada e o pacote é enviado através do Canal. O Canal, por sua vez, envia uma cópia do pacote para cada nó conectado a ele. Os pacotes são então recebidos por cada *Interface* de Rede de cada nó e "sobem" através dos componentes MAC, Camada de Enlace, Conector, Coletor e Agente. O Agente, finalmente, desempacota o dado e envia uma notificação de chegada de pacote para a Aplicação destino.

Adicionalmente, as extensões [40] ao *ns-2* adotadas implementam um mecanismo de monitoramento de recursos onde torna-se possível, por exemplo, monitorar a quantidade de energia residual do nó no nível da aplicação. Tal monitoramento é obtido através do acréscimo dos conceitos de **Recurso** e **Gerenciador de Recursos** ao nó padrão previamente definido, conforme mostra a Figura 14. O **Gerenciador de Recursos** provê uma *interface* de comunicação entre a Aplicação e cada recurso individualmente. Um **Recurso** pode ser qualquer coisa que necessite ser monitorada. No caso das simulações executadas, o recurso monitorado foi a energia, conforme mostra o diagrama da Figura 14.

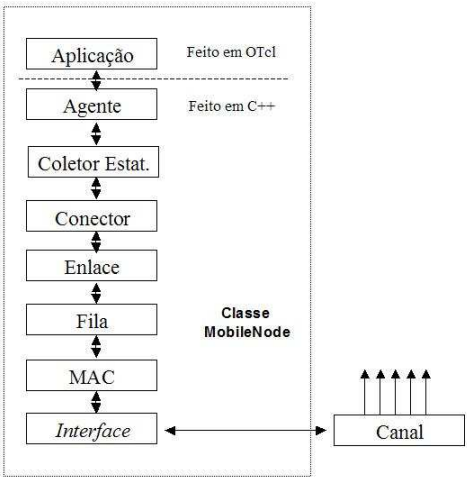

Figura 13 - Nó padrão adotado no *ns-2*

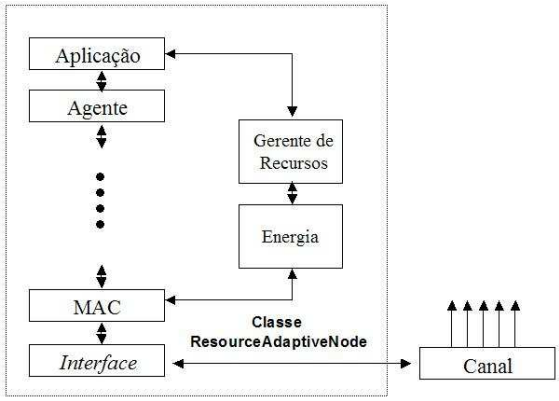

Figura 14 - Gerência de energia do nó padrão

Considerando o nó padrão adotado na simulação, como é mostrado na Figura 13, a Interface de Rede é responsável pela implementação das funções da camada física. Ao receber um quadro da camada MAC, ela configura uma potência de transmissão baseando-se numa aproximação da distância até o destino, subtrai da energia residual a energia necessária para

efetuar a transmissão, e envia o pacote para o Canal. Caso o nó tenha esgotado a sua energia após a transmissão de um pacote, ele é removido do Canal e não interfere mais no ambiente de simulação, sendo dada como perdida qualquer informação enviada para ele.

Quanto à recepção de dados provenientes da camada física, duas situações podem ocorrer. Na primeira situação, considere o caso em que o nó destino de uma transmissão está "dormindo". Esse mesmo nó destino detecta a recepção de informações oriundas do Canal através de sua *Interface* de Rede. Como o nó está "dormindo", a *Interface* de Rede descarta o pacote considerando que nós neste estado são incapazes de transmitir ou receber informações. Portanto, quando um nó está dormindo, também não há consumo de energia pois o nó simplesmente ignora qualquer transmissão sendo feita ao seu redor. Assim, fica como responsabilidade dos protocolos das camadas superiores fazer com que um nó destino de uma transmissão sempre esteja ativo durante tal transmissão.

Na segunda situação, considere o caso em que o nó destino de uma transmissão está ativo. Ao detectar uma recepção, a *Interface* de Rede do nó determina a potência do sinal recebido. Se essa potência for inferior a um limite de detecção ($P_{r\text{-}detect}$), o pacote é descartado, pois o nó não seria capaz de detectar tal transmissão na prática. Se a potência do sinal detectado for superior ao limite de detecção, porém, inferior ao limite de recepção ($P_{r\text{-}thresh}$), o pacote é marcado como defeituoso e passado para a camada MAC. Nesse caso, o pacote não pode ser sumariamente descartado já que afetaria a recepção de outros pacotes sendo transmitidos ao mesmo tempo. Finalmente, caso a potência do sinal detectado seja superior ao limite de recepção, o pacote é dado como recebido com sucesso e é passado para a camada MAC.

Quanto a camada MAC adotada nesta dissertação, pode-se dizer que a escolha do protocolo depende do cenário adotado. Para tal camada, foram adotados os protocolos CSMA, TDMA e CDMA, conforme mencionado no capítulo anterior.

Os protocolos CSMA e CDMA são implementados como extensões [40] ao simulador, ou seja, são programados como classes C++ compiladas juntamente com o simulador *ns*-2 por questões de desempenho. Para o protocolo TDMA, foi usado um esquema implementado no nível da aplicação que faz com que um determinado nó só transmita informações no seu devido *slot* de tempo.

Quanto ao modelo de energia adotado, ilustrado na Figura 15, foi adotado o mesmo modelo usado em [40], onde o processo de transmissão apresenta dois pontos de dissipação de energia. O primeiro ponto de dissipação é o próprio circuito eletrônico do rádio e o segundo é o amplificador de potência usado nas transmissões. Ambos os pontos de dissipação

apresentam um consumo de energia proporcional ao tamanho em bits (*k*) da mensagem transmitida. Entretanto, apenas o amplificador de potência apresenta um consumo proporcional à distância até o destino da transmissão. Para o caso do processo de recepção, o único ponto de dissipação é o circuito eletrônico do rádio que tem o consumo proporcional ao tamanho em bits (k) da mensagem recebida.

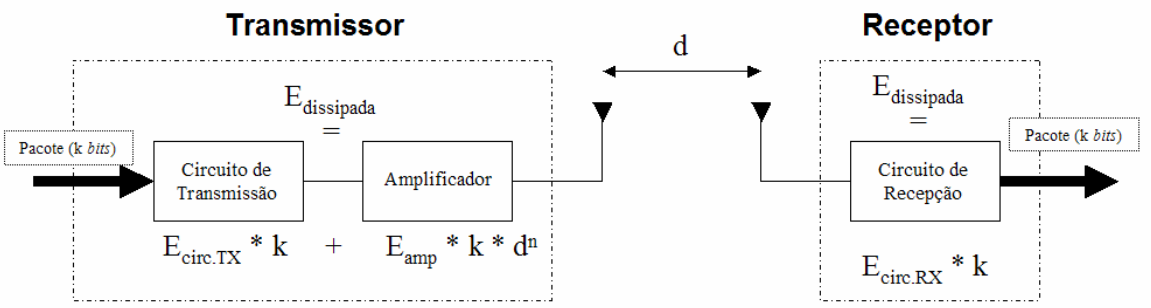

Figura 15 - Modelo de energia adotado

Adicionalmente, considerando que os nós são capazes de estimar a distância dos remetentes através da potência do sinal recebido, é possível regular a potência do sinal transmitido de modo a alcançar o nó alvo sem desperdiçar energia.

Na próxima seção, são enumeradas e descritas as métricas que foram adotadas para avaliar o desempenho do protocolo proposto nesta dissertação.

4.2 Métricas

A avaliação do desempenho do protocolo proposto foi feita através da análise de algumas métricas escolhidas visando viabilizar comparações com outras soluções encontradas na literatura [26,27] para o problema da disseminação de dados em RSSF. Foram utilizadas quatro diferentes métricas que são descritas a seguir.

- ***Cobertura***

A primeira métrica utilizada é a ***cobertura***. Tal métrica indica o quanto a região de monitoramento está coberta pela rede de sensores sem fio. Trata-se de uma métrica viável de ser obtida, pois cada pacote de informação enviado pelo nó sensor contém as coordenadas do nó remetente. Por outro lado, o nó sorvedouro possui um mapa em sua memória e toda vez que um pacote é recebido o sorvedouro marca o ponto referente às coordenadas contidas no pacote recebido. Cada ponto marcado no mapa é associado a um temporizador que expira após T segundos, ativando o procedimento de exclusão da marca referente ao ponto pelo nó

sorvedouro. Tal ponto poderá ser remarcado quando da recepção de um novo pacote em cujo conteúdo são encontradas as mesmas coordenadas. Assim, o nó sorvedouro possui um mapa em sua memória capaz de lhe dizer em determinado instante, quais são as áreas cobertas na rede.

O mecanismo tolerante a falhas proposto nesse trabalho utiliza essa métrica para fornecer a informação de quais células permanecem ou não cobertas após a falha de alguns nós. Uma célula descoberta é uma célula que não possui nenhum ponto marcado na memória do sorvedouro. Para tal, uma célula é definida como sendo uma região quadrada cuja área é equivalente a um percentual da área total de abrangência da rede. Além disso, as dimensões de tal quadrado são definidas de modo que o comprimento da sua diagonal não exceda o alcance máximo da sensibilidade do sensor de cada nó. Ou seja, se o sensor adotado nos nós é capaz de detectar a ocorrência de eventos a (no máximo) *y* metros de distância, então a diagonal da célula deve ser igual ou menor que *y*. Assim, a área total é dividida em um número *x* de células.

A métrica cobertura será útil na comparação dos mecanismos tolerantes a falhas propostos. É evidente que os mecanismos capazes de promover um número médio maior de células cobertas ao longo do tempo podem ser considerados mais eficientes se comparados aos outros. Além disso, a cobertura é considerada em inúmeros trabalhos como um parâmetro de QoS para as aplicações [38].

- ***Atraso médio* fim a fim**

O ***atraso médio* fim a fim** indica o tempo médio que uma unidade de informação leva para sair do nó sensor e chegar até o sorvedouro. Sua importância é clara em aplicações cuja notificação da ocorrência de eventos sensoriados precisa ser realizada em tempo real ou com um atraso que não ultrapasse um determinado limiar. Portanto, para tais tipos de aplicações, os protocolos mais adequados são aqueles que apresentam um menor atraso médio.

- ***Energia média dissipada***

A próxima métrica utilizada é a ***energia média dissipada***. Tal métrica indica a quantidade média de energia gasta por nó da rede para permitir que cada unidade de informação alcance o sorvedouro. Esta métrica tem uma grande importância considerando que eficiência em termos de energia é um dos fatores de maior impacto na vida útil das redes de sensores. Além disso, tal métrica reflete a razão entre a quantidade de energia dissipada por nó da rede e a quantidade de eventos recebidos no sorvedouro. Portanto, assim como em [26], para calcular tal razão, a quantidade total de energia dissipada em uma dada rodada de

simulação é dividida pelo total de nós da rede, e em seguida tal resultado é dividido pela quantidade total de pacotes efetivamente recebidos no sorvedouro.

- **Taxa de perda**

A métrica ***taxa de perda*** reflete a razão entre o número de pacotes entregues para o sorvedouro e o número de pacotes gerados pelos sensores da rede. Esta métrica indica a quantidade de informações que estão sendo enviadas e por algum motivo não estão sendo efetivamente entregues ao sorvedouro. Tal métrica tem um papel importante na avaliação dos resultados dos mecanismos tolerantes a falhas propostos, visto que indica se as falhas geradas pelo modelo de falhas estão provocando algum impacto na rede em termos de perda de informações. Além disso, o uso dessa métrica permitirá a comparação dos diferentes mecanismos propostos.

Adicionalmente, um conjunto de parâmetros foi selecionado de modo que pudessem ser alterados em diferentes rodadas de simulação. Na próxima seção, além de uma descrição detalhada dos cenários de simulação, são descritos também os parâmetros que foram variados entre as rodadas de simulação.

4.3 Cenários de Simulação

Nesta seção, são apresentados os cenários de simulação adotados assim como as justificativas para a adoção de cada um. A palavra cenário tem como objetivo abranger algumas características do ambiente de monitoramento, incluindo: tamanho e forma da região, coordenadas de cada nó da rede em tal região, número de *clusters*, número de sorvedouros, coordenadas de cada um dos sorvedouros, método de sincronização adotado, intensidade de tráfego, ocorrência ou não de falhas e, finalmente, número total de nós envolvidos na simulação.

Inicialmente, foi decidido usar 5 valores distintos para o número total de nós na rede. Os valores são 50, 100, 150, 200 e 250 nós. Estas quantidades de nós foram adotadas para facilitar a comparação com resultados de simulação obtidos em outros trabalhos como, por exemplo, LEACH e *Directed Diffusion*. Para o cenário contendo 50 nós, adotou-se uma região em forma de quadrado de 160m x 160m. Quanto à quantidade de nós desempenhando o papel de *cluster-head*, da mesma forma como foi feito em [27], adotou-se um valor de 6% do total de nós. Além disso, para os cenários com mais nós, as dimensões da região foram proporcionalmente aumentadas para não haver diferenças drásticas de densidade entre cada cenário. Para as simulações em cenários orientados a eventos, foram gerados eventos a uma

taxa de 2 eventos por segundo e os nós fontes foram espalhados por uma área correspondente a 43% de toda a área de observação. Finalmente, os valores exibidos nos gráficos são resultados de uma média obtida através de N rodadas de simulação para cada elemento da rede, onde N é suficiente para termos intervalos de confiança de 95%, sendo tais intervalos de largura menor ou igual a 20% do intervalo total exposto no gráfico. A Tabela 2 apresenta as quantidades de nós e as dimensões correspondentes da região.

Tabela 2 - Quantidade de nós e dimensões da região

Quantidade de nós	Dimensões da região (quadrado)
50	160x160 m
100	230x230 m
150	277x277 m
200	320x320 m
250	360x360 m

Em relação ao posicionamento de cada nó na rede, foram geradas coordenadas aleatórias uma única vez para cada tamanho de rede (quantidade de nós), como mostrado na Figura 16 e Figura 20. Além disso, tais figuras também mostram as topologias (em um dado instante de simulação) em forma de árvore de CHs, geradas pela execução do algoritmo de *Delaunay* e *Dijkstra*, considerando apenas um nó sorvedouro. Feito isso, as cinco distribuições geradas, uma para cada tamanho de rede, foram usadas ao longo de todo o processo de simulação. Portanto, como mencionado em seções anteriores, assume-se que não há mobilidade por parte de nenhum dos nós da rede.

Outras questões importantes em termos de cenários de simulação para avaliação de desempenho, são: a adoção de um ou mais sorvedouros, a abordagem de sincronização entre *cluster-heads* adotada para as transmissões de encaminhamento e a adoção ou não de um mecanismo de tolerância a falha. Para tal, essas questões são descritas nas próximas sub-seções.

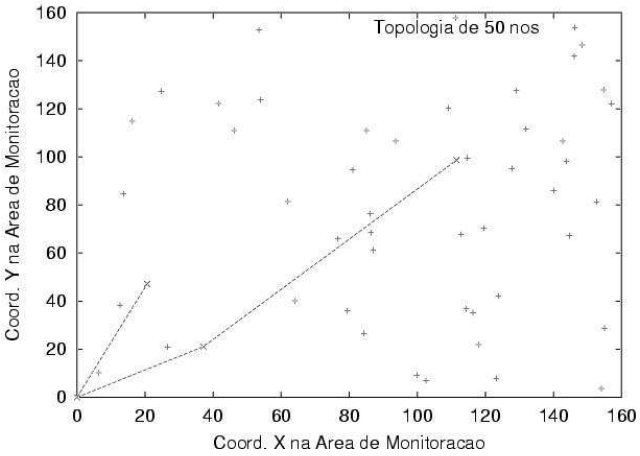

Figura 16 - Topologia de rede com 50 nós

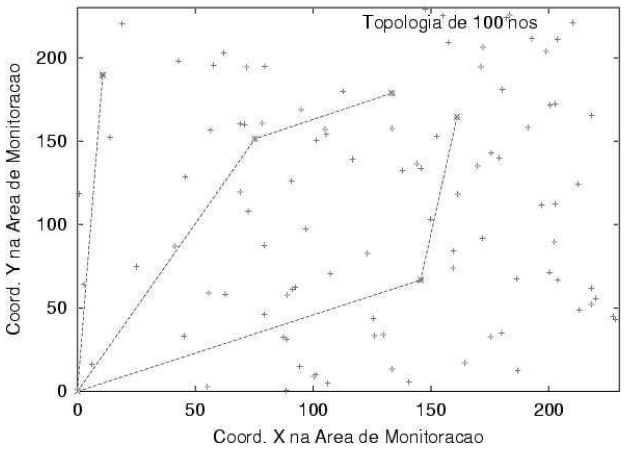

Figura 17 - Topologia de rede com 100 nós

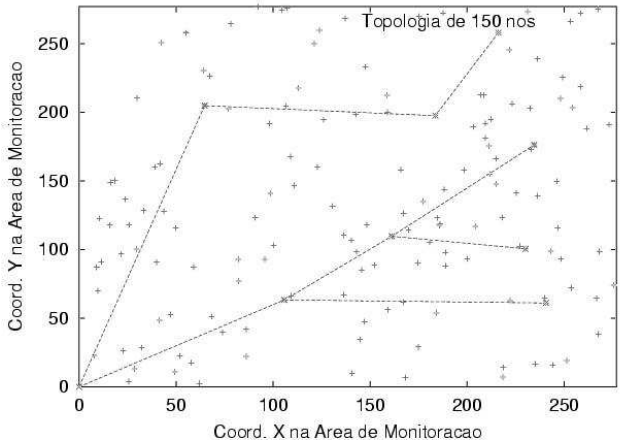

Figura 18 - Topologia de rede com 150 nós

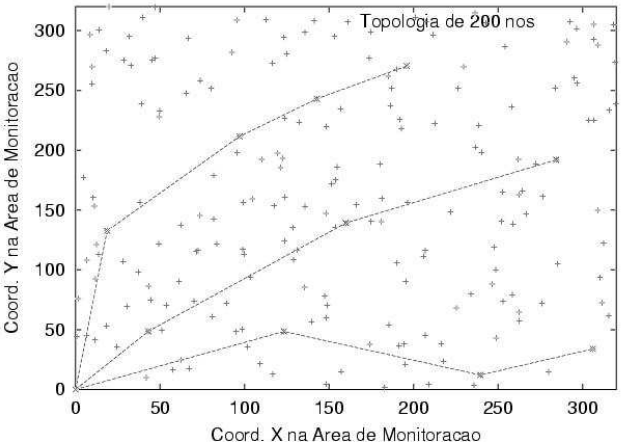

Figura 19 - Topologia de rede com 200 nós

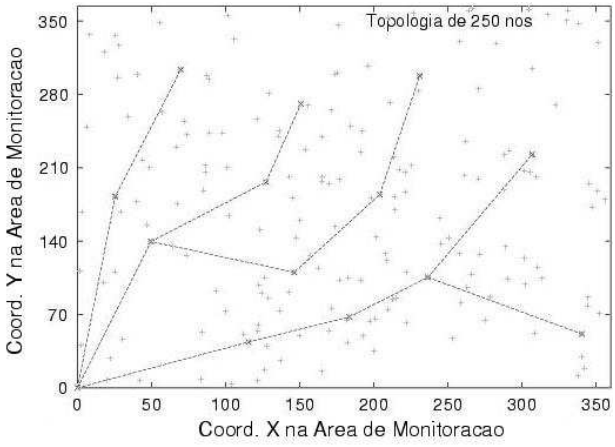

Figura 20 - Topologia de rede com 250 nós

4.3.1 Cenários com um *sorvedouro* ou Múltiplos *sorvedouros*

A grande maioria dos cenários de rede de sensores encontrados na literatura, conforme mencionado no Capítulo 3, adotam um único nó como sorvedouro, para onde todas as informações sensoriadas convergem. Logo, considera-se nesta dissertação também a adoção de 1 nó sorvedouro. Entretanto, na Seção 3.3.7 foi proposta uma adaptação ao protocolo proposto de modo a permitir o uso de múltiplos *sorvedouros*, visando avaliar o comportamento da rede na existência de vários destinos para as informações coletadas.

Quanto ao posicionamento de cada sorvedouro na região monitorada, são utilizadas as disposições ilustradas na Figura 21, a seguir, onde cada círculo indica a posição de um sorvedouro dentro da região de monitoramento.

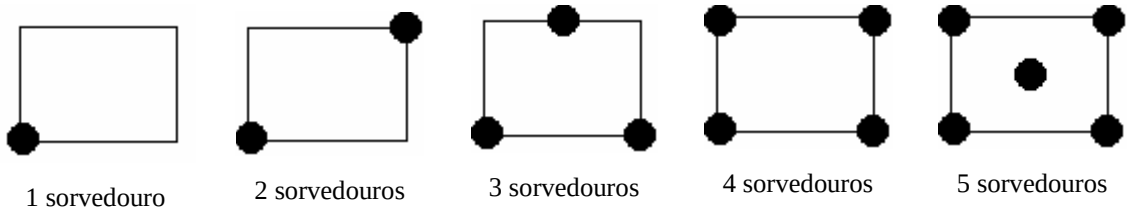

Figura 21- Posição dos sorvedouros em cada cenário

Visando avaliar se tais distribuições de sorvedouros são efetivamente uniformes ao longo da área de monitoração, foi executado o processo descrito a seguir.

Um *grid* de nós uniformemente distribuídos ocupando toda a área quadrada foi criado e a distância média de cada um dos nós do *grid* até o sorvedouro mais próximo foi computada. Assumindo que a distribuição de sorvedouros seja uniforme e que a área da região quadrada e o número de nós do *grid* tenham o mesmo valor para todas as cinco configurações (número de sorvedouros), a distância média deve ser inversamente proporcional ao número de sorvedouros. Entretanto, conforme mostra a Figura 22, o cenário de 4 sorvedouros apresenta uma distância média entre os nós e o sorvedouro mais próximo maior do que o cenário de 3 sorvedouros, portanto a distribuição não está rigorosamente uniforme como esperado. Portanto, conclusões tiradas a partir dos resultados expostos na seção 4.4.7 (Avaliação dos resultados com múltiplos sorvedouros), devem levar tal questão em consideração.

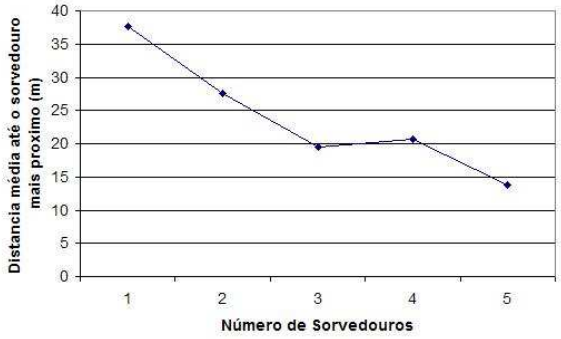

Figura 22- Distância Média até o sorvedouro mais próximo X Número de sorvedouros

4.3.2 Cenários com Sincronização Local ou Remota

Esta seção descreve dois tipos de cenários de simulação diferentes em relação ao método de sincronização adotado para as transmissões de encaminhamento. A fim de contornar a limitação física do rádio usado nos sensores, é adotado um método diferente para tal sincronização em cada um dos cenários. A limitação física do rádio, conforme descrita no Capítulo 3, diz respeito à incapacidade de um nó sensor receber informações durante uma transmissão. No primeiro tipo de cenário, o método de sincronização adotado, denominado ***sincronização local***, trata da sincronização do *cluster-head* com seu próprio *cluster*, isto é, o *cluster-head* sincroniza a sua transmissão de encaminhamento com o cronograma TDMA do seu próprio *cluster*. No segundo, onde é adotado um método de sincronização denominado ***sincronização remota***, as transmissões de encaminhamento de um determinado *cluster-head* são feitas respeitando-se o cronograma TDMA do *cluster* pai.

Com o intuito de viabilizar a comparação entre as duas soluções, serão apresentados na Seção 4.4 resultados de rodadas de simulação considerando tanto cenários adotando o método de *sincronização local* como cenários adotando o método de *sincronização remota*.

4.3.3 Cenários com falhas e *Cluster-head* reserva

Para avaliar o mecanismo de tolerância a falhas do protocolo proposto nesta dissertação, fez-se necessária a criação de diferentes tipos de cenários de simulação. O primeiro ponto diz respeito à ativação ou não do gerador de falhas. Desse modo, alguns cenários simulam a ocorrência de falhas nos nós, enquanto outros não. Um segundo ponto importante diz respeito ao uso do mecanismo tolerante a falhas propriamente dito. Com isso, são criados tanto cenários de simulação utilizando o mecanismo de *cluster-head* reserva, como cenários que

não usam tal mecanismo de tolerância a falhas. Além disso, esses últimos cenários são combinados com a ativação ou não do gerador de falhas.

Finalmente, para avaliar todas as funcionalidades do protocolo em termos de tolerância a falhas, foi necessário criar mais duas opções para os cenários de simulação. Na primeira, é adotado o método aleatório para a seleção do nó reserva. Na segunda, é adotado o método diferenciado. Assim, cria-se um conjunto de cenários de simulação que permitem a avaliação de vários resultados, conforme mostrado na Seção 4.4.

Concluindo esta seção, vale ressaltar que o gerador de falhas adotado faz com que, em um dado instante da simulação, todos os *cluster-heads* tenham a mesma probabilidade de falhar. Além disso, como já foi dito, falhas são geradas apenas em nós desempenhando o papel de *cluster-head*.

4.3.4 Geração de eventos

Quanto ao mecanismo de geração de eventos na rede, que é igual em todas as rodadas de simulação, adotou-se um esquema onde cada nó sensor da rede observa a ocorrência de um novo evento a cada *x* segundos, sendo *x* uma variável aleatória uniforme que pode assumir valores entre 0.5 e *t*. Nesse caso, *t* é um parâmetro variado entre os cenários para viabilizar uma análise detalhada do comportamento do protocolo diante de diferentes cargas de tráfego na rede. Isso é possível, pois o nível de sobrecarga em termos de tráfego na rede é inversamente proporcional ao valor de *t*.

4.4 Resultados

Nesta seção, são mostrados os resultados obtidos ao longo de todo o processo de avaliação da proposta. Tal avaliação foi realizada através de simulações, análises e comparação de resultados.

As simulações foram divididas em 2 fases. Na primeira fase (Seções 4.4.1-4.4.6), são considerados cenários que utilizam apenas um sorvedouro e na segunda fase (Seção 4.4.7) são considerados cenários com vários sorvedouros.

A primeira fase da avaliação, por sua vez, foi dividida em cinco etapas. O objetivo da primeira etapa (Seção 4.4.1) é apresentar resultados em termos de consumo de energia por parte de cada nó da rede, de um modo geral, assim como mostrar a uniformidade de tal consumo de energia ao longo do tempo. A segunda etapa (Seção 4.4.2), com o intuito de

avaliar a eficiência do protocolo proposto em termos de energia, compara os resultados obtidos na simulação do protocolo proposto com os resultados obtidos em [26], na avaliação do protocolo *Directed Diffusion*.

A terceira etapa (Seção 4.4.3), também com o intuito de avaliar a eficiência do protocolo proposto em termos de energia, compara os resultados obtidos na simulação do protocolo proposto com os resultados obtidos em [27], na avaliação do protocolo LEACH.

Os resultados relativos aos diferentes mecanismos de sincronização propostos são mostrados na quarta etapa (Seção 4.4.4). Finalmente, a quinta etapa (Seção 4.4.5) mostra os resultados obtidos com o mecanismo de tolerância a falhas proposto. Adicionalmente, a Seção 4.5.6 mostra um resultado obtido com medições feitas com o intuito de encontrar um valor ideal para o intervalo entre as fases de rotação do protocolo.

4.4.1 Consumo de energia

Nesta primeira etapa, é mostrado o consumo de energia por parte de cada nó da rede ao longo de sua vida útil. O resultado do consumo é mostrado em forma de gráfico tridimensional, onde os eixos x e y representam as coordenadas de cada nó no ambiente de monitoração, e o eixo z mostra o nível de energia de cada nó em cada momento. Vale ressaltar que cada gráfico representa uma fotografia dos níveis de energia em determinado instante da vida da rede e o nível inicial de energia de todos os nós é de 2J.

Conforme ilustrado na Figura 23, foi usado o cenário de 50 nós e 3 *clusters*. Tal figura, bidimensional, mostra o posicionamento inicial de cada um dos *cluster-heads* para que se possa observar nos gráficos tridimensionais (Figura 24 a Figura 30) um consumo de energia mais acentuado em tais regiões. Porém, observa-se nessas figuras (Figura 24 a Figura 30) que o consumo de energia ocorre de maneira uniforme ao longo da rede.

Deve-se observar que as saliências nos gráficos mostrados (Figura 24 a Figura 30) ocorrem nas regiões onde encontram-se os 3 *cluster-heads* iniciais, sendo causadas pelo consumo inicial de energia desses *cluster-heads* durante a primeira rodada de coleta, e são mantidas até o fim da simulação. O fato de tais saliências existirem e se manterem bem definidas até o fim da simulação pode ser visto como mais uma prova da uniformidade do consumo de energia provido pelo mecanismo de rotação dos *cluster-heads* após a primeira rodada de coleta. Tal uniformidade é importante pois indica que não ocorrem mortes prematuras de alguns dos nós, o que poderia levar a partições na rede.

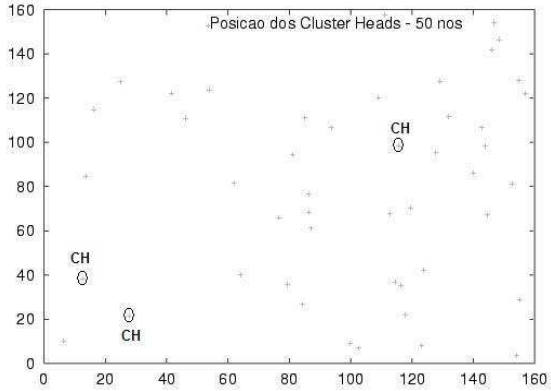

Figura 23 - Posição dos *cluster-heads* para a rede de 50 nós

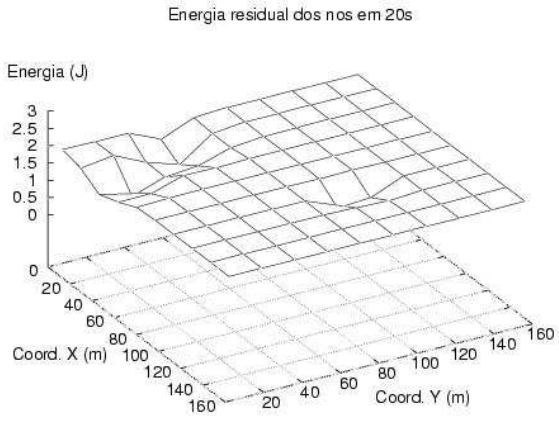

Figura 24 - Nível de energia de cada nó em t=20s

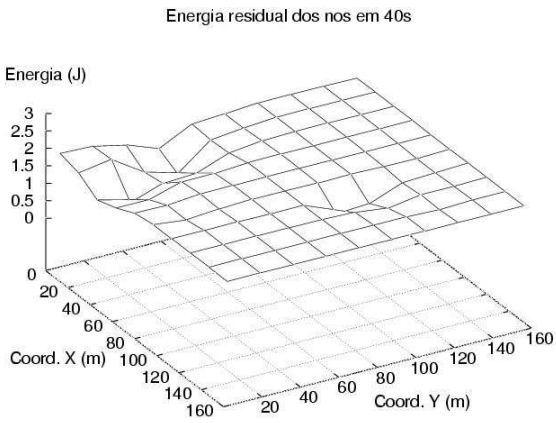

Figura 25 - Nível de energia de cada nó em t=40s

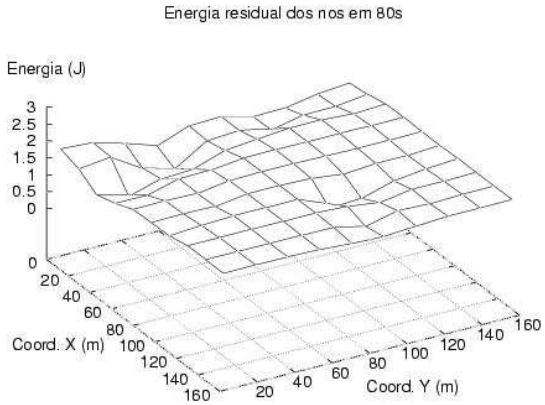

Figura 26 - Nível de energia de cada nó em t=80s

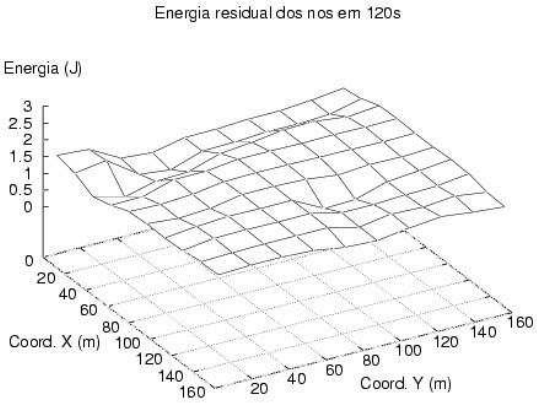

Figura 27 - Nível de energia de cada nó em t=120s

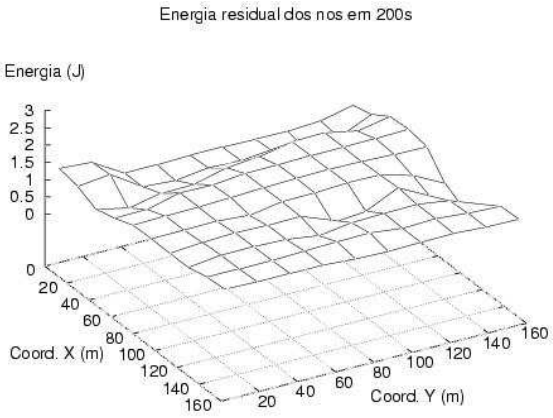

Figura 28 - Nível de energia de cada nó em t=200s

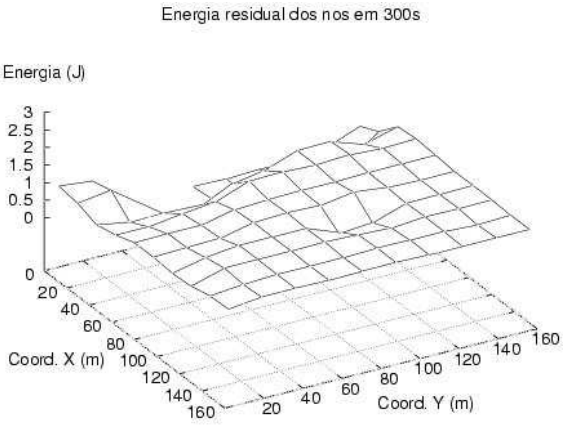

Figura 29 - Nível de energia de cada nó em t=300s

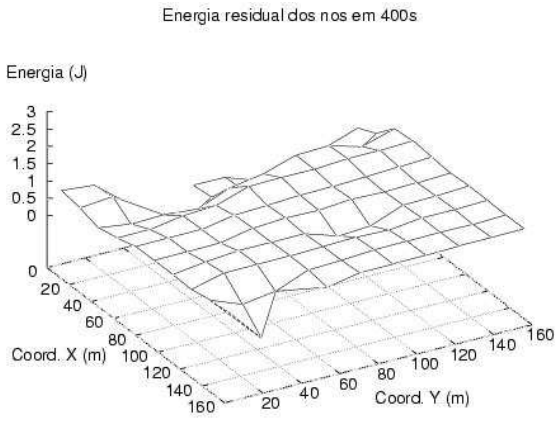

Figura 30 - Nível de energia de cada nó em t = 400s

4.4.2 Comparação com um protocolo de topologia plana

Nesta segunda etapa, os resultados obtidos na simulação do protocolo proposto são comparados com aqueles obtidos na simulação do protocolo *Directed Diffusion* em [26], visando mostrar que o presente trabalho propõe uma solução mais eficiente em termos de energia para redes de sensores que adotam um método orientado interesses. O modelo de energia adotado é o mesmo que foi adotado em [26], onde um nó gasta 35mW nos períodos de ociosidade, 395mW para receber informação através do seu circuito de rádio e 600mW para transmitir informações.

Nesta etapa, foram analisadas duas das mesmas métricas usadas em [26]: o **atraso médio** na entrega da informação e a **energia média dissipada.** Os gráficos da Figura 31 e da Figura 32 mostram os resultados das duas métricas nos cinco (5) tamanhos de rede considerados, comparando o protocolo proposto com o protocolo *Directed Diffusion*.

O gráfico da Figura 32 mostra a **energia média dissipada**. De acordo com esse gráfico, pode-se perceber que o protocolo proposto apresenta um ganho médio considerável em termos de energia. Esse ganho é atribuído às diferenças entre os métodos de roteamento adotados pelos protocolos. O protocolo proposto faz com que o sorvedouro envie informações, por exemplo, de ativação, para todos os nós através dos *cluster-heads* que, por sua vez, enviam uma única mensagem em *broadcast* para todo o *cluster* visando economizar a energia que seria gasta com transmissões. Além disso, todas as mensagens de controle que são enviadas por um *cluster-head* para o seu *cluster*, são feitas uma única vez, em *broadcast*. Já o *Directed Diffusion* gasta energia enviando tais informações de controle através de um método de inundação que ativa todos os nós da rede.

Quanto ao **atraso médio**, observa-se no gráfico da Figura 31 um atraso médio maior no protocolo proposto quando comparado ao *Directed Diffusion*. Este aumento no atraso ocorre pois a comunicação entre *cluster-heads* é implementada usando-se um esquema TDMA. Esta implementação do módulo de comunicação *inter-cluster* implica uma espera por parte de cada *cluster-head* pelo seu respectivo *slot* antes de transmitir para o próximo *cluster-head*.

Adicionalmente, deve ser mencionado o fato de que nesta etapa, as simulações estavam sendo feitas usando apenas um nó *sorvedouro*. É fácil notar que tendo apenas um sorvedouro, toda a informação coletada ao longo da rede tem que convergir para esse mesmo sorvedouro, criando um atraso médio realmente maior. No *Directed Diffusion*, as simulações foram executadas em redes contendo 5 nós sorvedouros. Entretanto, na seção 4.4.7, serão apresentados gráficos de simulações onde foi usado mais de um nó sorvedouro com o intuito de mostrar o comportamento do protocolo proposto diante desse tipo de cenário.

4.4.3 Comparação com um protocolo hierárquico

A execução das simulações desta terceira etapa tem por objetivo avaliar os ganhos de energia por se adotar um mecanismo de comunicação *multi-hop* entre os *cluster-heads* ao invés de usar sempre transmissão direta para o sorvedouro. Mais uma vez, tais comparações são feitas usando apenas um sorvedouro pois o protocolo LEACH não prevê a utilização de múltiplos sorvedouros.

Em outras palavras, os resultados obtidos na simulação do protocolo proposto foram comparados com aqueles obtidos na simulação do protocolo LEACH, visando mostrar que o presente trabalho propõe uma solução eficiente em termos de energia para redes de sensores que adotam um esquema hierárquico e periódico para o envio das informações. Assim, o intuito desta comparação foi observar o comportamento do protocolo proposto em um cenário onde é necessário fazer uma monitoração contínua. Esse cenário é o mesmo onde o protocolo LEACH se aplica de maneira eficiente, ou seja, trata-se de um cenário onde todos os nós da rede estão ativos e continuamente detectando novos eventos para transmití-los para os seus *cluster-heads*.

O gráfico da Figura 33 mostra a energia (J) total dissipada pelos protocolos (proposto e LEACH) em um intervalo de simulação de 100s. O modelo de energia usado nessa comparação foi o mesmo usado em [27], na avaliação do protocolo LEACH. De acordo com o gráfico da Figura 33, pode-se perceber que os nós da rede gastaram até 5 vezes menos energia quando executando o protocolo proposto. Esse ganho pode ser explicado considerando que o

protocolo proposto faz com que os *cluster-heads* repassem suas informações para *cluster-heads* vizinhos ao invés de enviar diretamente para o distante nó sorvedouro, como é feito no LEACH.

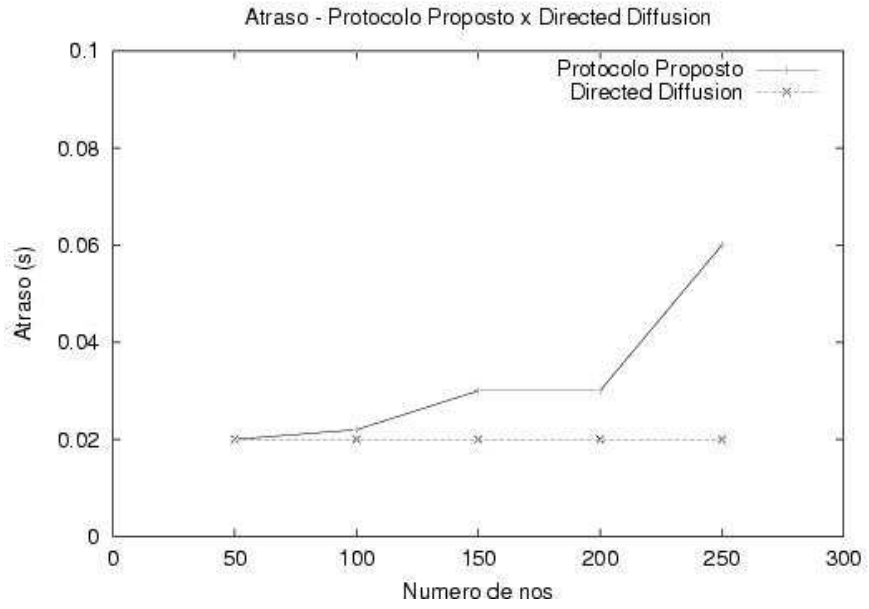

Figura 31 - Atraso(s) - Proposto x DD

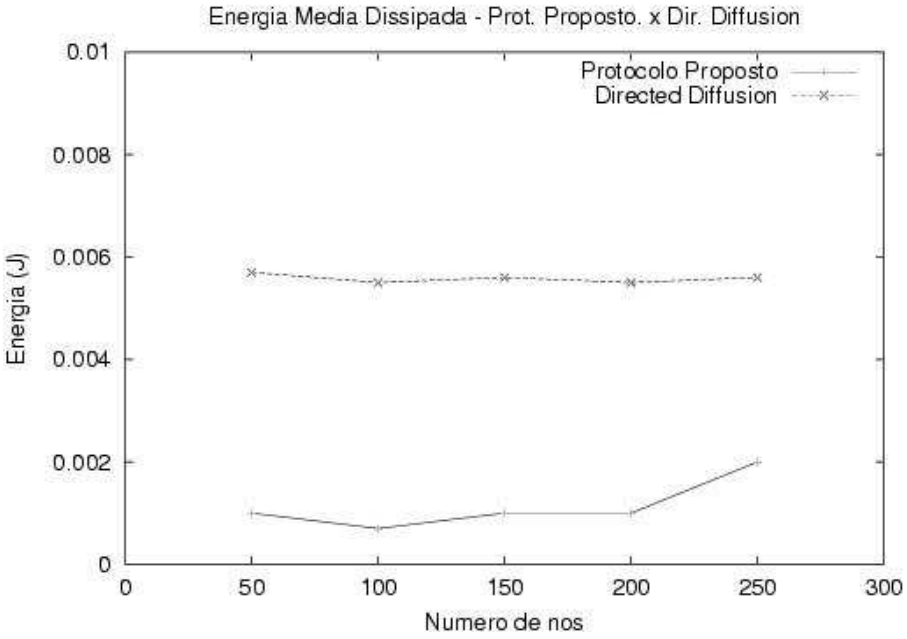

Figura 32 - Energia Média (J) - Proposto x DD

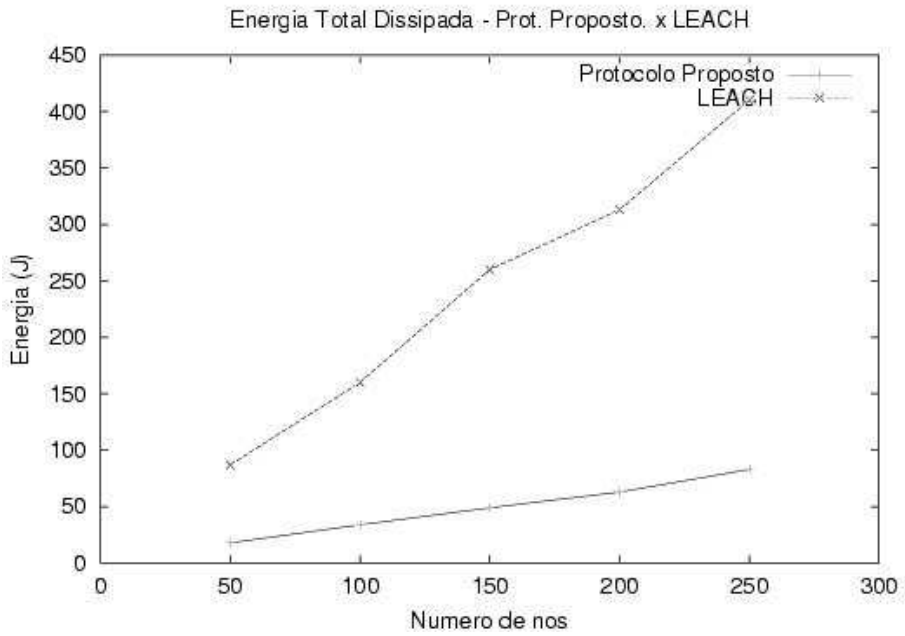

Figura 33- Energia Total(J) – Proposto x LEACH

4.4.4 Avaliação dos métodos de Sincronização

O objetivo desta quarta etapa é avaliar e comparar os possíveis métodos de sincronização entre transmissões e recepções na rede, a fim de contornar a limitação física do rádio usado pelos sensores, conforme explicado no Capítulo 3. Os dois métodos de sincronização simulados, considerando as transmissões de encaminhamento de um determinado *cluster-head*, são a sincronização com o cronograma TDMA do *cluster* pai (**sincronização remota**) e a sincronização com o cronograma TDMA do próprio *cluster* (**sincronização local**).

Dentre os resultados obtidos nas comparações entre os dois métodos, o de maior relevância está relacionado à taxa de perda de dados. Conforme ilustrado na Figura 34, deve-se observar que na sincronização local não ocorrem perdas de informações coletadas no próprio *cluster*, visto que a transmissão de encaminhamento do *cluster-head* jamais coincide com uma recepção de informações vindas dos nós puramente sensores do seu *cluster*. Assim, a taxa de perda é maior na sincronização remota pois o *cluster-head*, ao encaminhar periodicamente as informações para o próximo *cluster-head*, despreza o cronograma interno do seu *cluster*. Logo, as chances de sua transmissão de encaminhamento coincidir com a recepção de um pacote vindo de um nó puramente sensor de seu *cluster* são grandes, o que causa uma maior perda de pacotes.

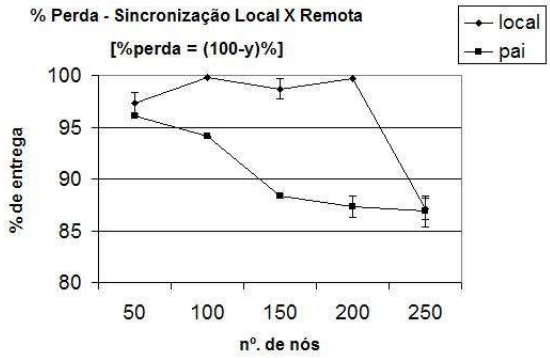

Figura 34 - % Perda - Sincronização Local X Remota

4.4.5 Avaliação do mecanismo de Tolerância a Falhas

O objetivo dessa quinta etapa é mostrar a importância de se adotar um nó reserva se comparada com a sua não adoção, considerando-se as métricas mencionadas anteriormente. Além disso, é avaliado como as duas alternativas para a seleção do nó reserva se comportam com diferentes tamanhos de redes e/ou tráfegos mais intensos.

Nos próximos parágrafos, os resultados obtidos com a adoção de cada uma das diferentes abordagens são apresentados e analisados.

- **Nó reserva com esquema de eleição aleatório**

Inicialmente, são apresentados os resultados obtidos para as 4 métricas, comparando-se situações onde é usado nó reserva com situações onde tal mecanismo não foi usado. Além disso, as simulações foram executadas usando o método aleatório de seleção do nó reserva.

Os resultados referentes à métrica ***taxa de perda*** para uma bateria de simulações são mostrados na Figura 35. Nessas simulações, foi variado o tamanho da rede (número de nós) dentre os 5 valores considerados (50,100,150,200,250). Além disso, foram adotados dois padrões de tráfego distintos, t=5 e t=30, tanto em cenários usando o nó reserva quanto em cenários sem o nó reserva.

Analisando tais resultados, observa-se que a adoção do nó reserva promove uma menor taxa de perda para todos os tamanhos de rede, independente do padrão de tráfego em questão. Adicionalmente, pode-se observar que o protocolo não se comporta bem diante de uma sobrecarga (t=5) de tráfego na rede, com ou sem adoção do nó reserva. Nesse caso, as taxas

de perda ficam elevadas devido ao crescimento acelerado das filas de saída dos nós *cluster-head*, resultando em descarte de pacotes. Esse crescimento das filas ocorre porque o número de recepções é maior que o número transmissões, pois o cronograma TDMA de cada *cluster-head* apresenta vários *slots* para recepção (um para cada nó do *cluster*) e apenas um *slot* para transmissão. Além disso, esta sobrecarga se agrava para os nós mais próximos do sorvedouro.

A combinação do crescimento acelerado das filas com o momento da rotação dos *cluster-heads* resulta em taxas de perda elevadas, pois as informações remanescentes em suas filas são perdidas no momento da rotação. Portanto, com o intuito de evitar perdas provocadas por sobrecarga, recomenda-se que o limite inferior para o parâmetro t, responsável pela geração de tráfego, seja 30. Ou seja, ambientes de monitoração que apresentam um padrão de ocorrência de eventos análogo a t menor que 30, não são adequados para o uso do protocolo proposto.

Já os resultados referentes à métrica ***energia média dissipada*** para uma bateria de simulações são ilustrados no gráfico da Figura 36. Nessas simulações, foi variado o tamanho da rede entre os 5 valores, e o padrão de tráfego foi gerado para 3 valores distintos, t=5,t=15 e t=30, tanto em cenários usando o nó reserva como em cenários sem o nó reserva. Analisando os resultados, nota-se que o uso do nó reserva traz uma grande economia em termos de energia nas redes que possuem um número mais elevado de nós. Em outras palavras, esse benefício é mais acentuado na simulação de redes de 250 nós usando nó reserva. Observa-se que em redes de 250 nós, os nós gastaram uma quantidade média de energia bem inferior aos cenários que não adotaram o nó reserva. Tal melhoria de desempenho fica ainda mais acentuada quando a rede é submetida a padrões de tráfego mais intensos, podendo ser parcialmente atribuída à maneira como é calculada a métrica energia média dissipada, onde a energia total dissipada em uma rodada é dividida pelo número de pacotes recebidos no sorvedouro e em seguida pelo número total de nós.

Quanto à métrica **cobertura**, constata-se, adicionalmente, através da análise dos resultados ilustrados na Figura 37, que a adoção de um mecanismo tolerante a falha, isto é, a presença de um nó reserva, implica um aumento de cobertura para todos os tamanhos de rede em valores superiores a 90% da região monitorada. Logo, a ausência de um mecanismo de contingência para cada *cluster* pode ser catastrófica em cenários onde podem ocorrer falhas em *cluster-heads*. Nesse caso, quando um *cluster-head* falha, além da área monitorada pelo seu *cluster* ficar descoberta, todas as áreas monitoradas por *clusters* filhos também ficam descobertas.

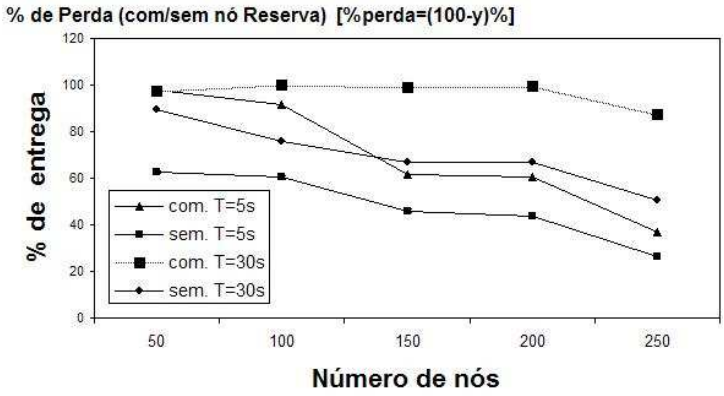

Figura 35 - % de perda (com / sem) nó reserva

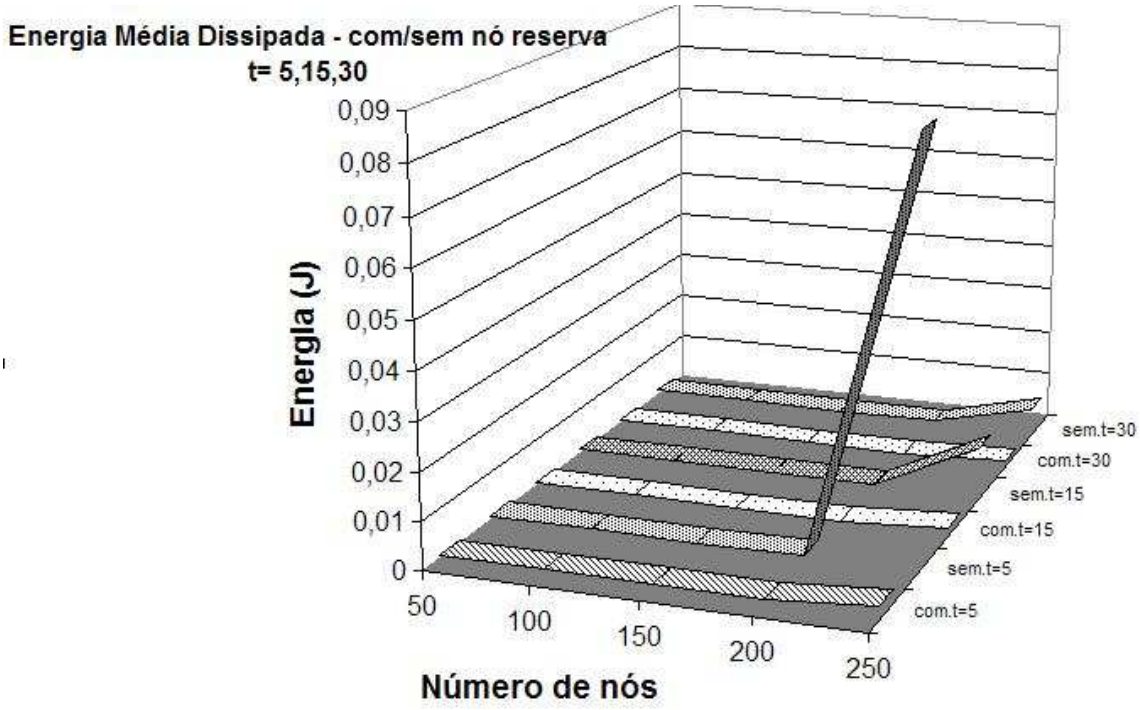

Figura 36 - Energia Média Dissipada - com/sem CH reserva

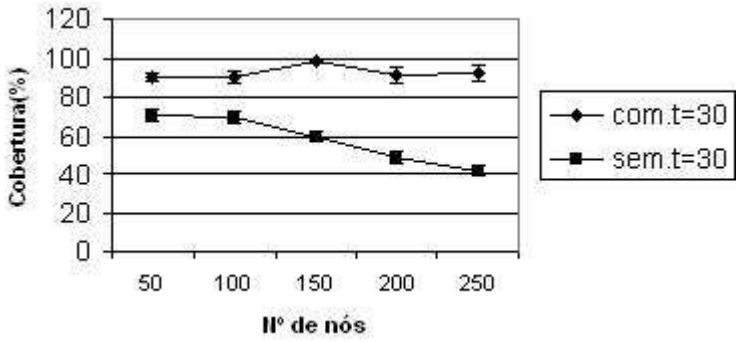

Figura 37 - % de Cobertura (com/sem nó reserva) X Número de nós

Vale ressaltar que o uso do mecanismo tolerante a falhas promove uma cobertura mais acentuada na simulação de redes com 150 nós. Considerando que as coordenadas de cada nó foram geradas aleatoriamente, uma distribuição de determinado tamanho pode apresentar uma qualidade melhor em termos de espalhamento do que outra. Portanto, essa é uma possível causa para a melhor cobertura observada para a rede com 150 nós.

- **Nó reserva com esquema de eleição diferenciado**

Nos próximos parágrafos, são descritas as simulações que contemplaram a utilização do método diferenciado para a seleção do nó reserva. O intuito desta abordagem é que a seleção diferenciada traga uma distribuição do gasto de energia mais uniforme, comparada com a seleção aleatória, considerando que só são selecionados para nó reserva aqueles que estão com seu nível de energia baixo dentre aqueles que estão acima da média. Assim, os nós selecionados ganham a oportunidade de economizar energia enquanto estão desempenhando tal papel, já que na função de nó reserva o nó fica apenas monitorando as transmissões do *cluster-head* titular, sem realizar transmissões.

As simulações que geraram os resultados expostos no gráfico da Figura 38, foram executadas para uma rede de 50 nós, tendo uma carga de tráfego equivalente a t=30 e tendo todos os nós da rede ativos simultaneamente. Em tal gráfico é mostrado o nível médio de energia dos nós da rede ao longo do tempo. Adicionalmente, é mostrado o desvio padrão de cada uma das médias calculadas. Nesse gráfico, observa-se que o nível médio de energia da rede, começando em 2J, decresce ao longo do tempo, como era de se esperar. Além disso, o nível de energia é similar para ambos os métodos de seleção do nó reserva, aleatório ou diferenciado. Entretanto, nota-se que o desvio padrão da energia média é crescente para o método aleatório e mantém-se relativamente constante para o método diferenciado. Portanto, considerando que quanto maior o desvio padrão, menos uniforme é o gasto de energia, pode-se considerar que o método diferenciado faz com que a rede apresente um consumo mais uniforme de energia. O fato da rede apresentar um consumo mais uniforme de energia é desejável, como já foi dito, para evitar a morte prematura de nós críticos para a aplicação e a partição da rede.

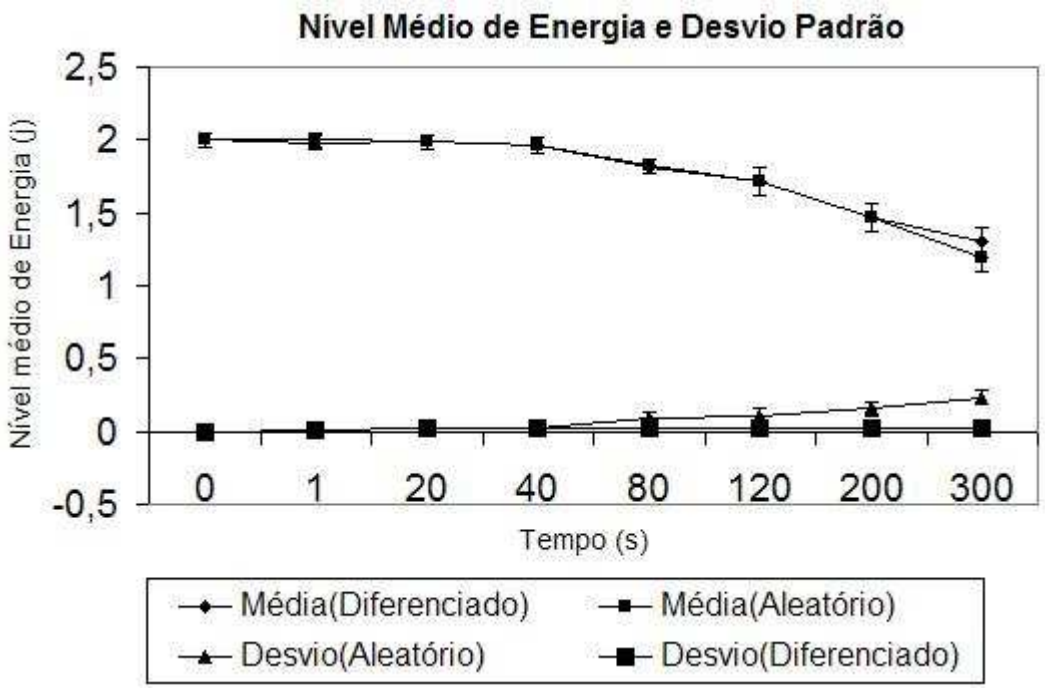

Figura 38- Nível Médio da energia dos nós e desvio padrão

4.4.6 Intervalo Entre as Fases de Rotação

No início do processo de simulação do protocolo proposto, percebeu-se que o tempo adotado como intervalo entre as fases de rotação poderia influenciar no desempenho do protocolo. Visando encontrar um valor ideal para um determinado cenário de rede, foram feitas avaliações para vários intervalos. Os resultados, apesar de não apontarem um valor ótimo para tal medida, considerando-se quaisquer cenários, servem para indicar a existência de tal valor ótimo para cada cenário de rede. Neste caso, foi adotado o cenário de 50 nós descrito na Seção 4.3. Conforme é mostrado na Figura 39, as simulações que foram executadas adotando o intervalo entre cada fase de rotação, em torno de 10 e 12 segundos, apresentaram melhores resultados, visto que a adoção de tal intervalo adiou ao máximo a ocorrência da primeira morte de um nó na rede. Tais resultados estão expressos em termos do instante da primeira morte de um nó causada pelo fim de seus recursos de energia.

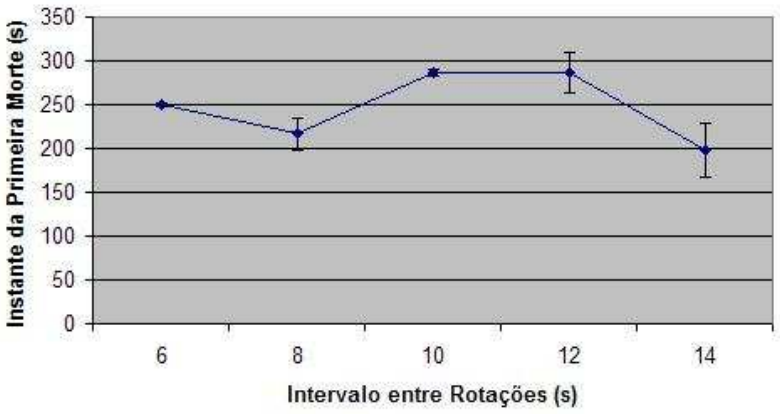

Figura 39- Intervalo entre Rotações(s) x Instante da primeira morte

4.4.7 Avaliação dos Resultados em Cenários com Múltiplos Sorvedouros

Considerando agora cenários contendo múltiplos sorvedouros, foram executadas rodadas de simulação em redes de um até cinco sorvedouros, variando também o tamanho da rede em número de nós de 100 até 250 nós. O motivo pelo qual, neste caso, o menor tamanho de rede simulado foi 100 ao invés de 50, como tem sido feito em todas as simulações desta dissertação, é que foi adotado como padrão um valor de 5% do total de nós para desempenhar o papel de *cluster-head*. Assim, para a rede de tamanho 50, teríamos apenas 3 *cluster-heads*. Como foi criada uma restrição, a nível de simulação, de que cada sorvedouro deve possuir pelo menos um nó *cluster-head* raiz de uma árvore de *cluster-heads* associado a ele, por exemplo, em um cenário com 50 nós e x sorvedouros, onde $x > 3$, não teríamos *cluster-heads* suficientes para simular. Portanto, decidiu-se simular apenas redes com 100 ou mais nós, onde temos sempre pelo menos 5 *cluster-heads*.

A seguir (Figura 40, Figura 41 e Figura 42) são mostrados os resultados obtidos com tais simulações. As métricas monitoradas durante as simulações foram as mesmas dos resultados expostos anteriormente, ou seja, **atraso médio**, **energia média dissipada** e **taxa de perda**. Os gráficos para cada tamanho de rede/métrica foram gerados separadamente para facilitar a visualização conforme mostram as figuras a seguir.

Observando os gráficos, pode-se notar, segundo a Figura 41, que quanto maior o número de sorvedouros, independente do tamanho da rede, menor o **gasto médio de energia** para fazer as informações chegarem até um dos sorvedouros. Tal conclusão é explicada considerando que os sorvedouros estão espalhados de maneira aproximadamente uniforme pelo espaço de monitoramento e que quanto maior o número de sorvedouros maiores as chances de haver um sorvedouro próximo de um determinado nó. Além disso, quanto maior a

chance de haver um sorvedouro próximo de um determinado nó, maior é a chance de evitar que o nó tenha que enviar as informações coletadas para sorvedouros distantes através da comunicação *multi-hop*.

Um outro resultado que traz conclusões intuitivas, é mostrado na Figura 40, onde são mostradas as curvas de **atraso** na entrega da informação. Observando tais curvas, constata-se também que quanto maior o número de sorvedouros, menor é o atraso. Tais resultados são interessantes para mostrar que, assim como na métrica de energia, quanto maior o número de sorvedouros, maiores são as chances de haver um sorvedouro mais próximo de um determinado nó. Conseqüentemente, maiores são as chances desse nó entregar as suas informações ao sorvedouro em um número menor número de saltos, resultando assim, em um menor tempo de entrega. Além disso, esse atraso também é claramente proporcional ao tamanho da rede, conforme pode ser observado na Figura 40. Esse resultado pode ser justificado considerando que quanto maior o número de nós da rede, mais níveis possui sua árvore de *cluster-heads* e, conseqüentemente, maior é o número de saltos que cada pacote de informação tem que percorrer, aumentando o seu tempo total para entrega, a cada salto.

Uma outra avaliação importante é a avaliação da **taxa de perda**. Tal métrica reflete o nível de perda de informações na rede. Tais perdas podem ser causadas por falhas temporárias ou definitivas de um ou mais nós, por sobrecarga de tráfego na rede, ou até mesmo por limitações físicas no *hardware* do nó. Esse último caso pode ocorrer porque o *hardware* do nó não é capaz de receber informações durante uma transmissão, isto é, ocorrerão perdas toda vez que um *cluster-head* estiver transmitindo uma informação para o seu *cluster-head* pai e, por exemplo, o seu *cluster-head* filho ou um nó sensor do seu *cluster* lhe enviar informação.

Portanto, o gráfico da Figura 42 mostra as curvas de perda para 3 tamanhos de rede (100,150,200). Observando-se tal gráfico, pode-se perceber que quanto maior o número de nós, em média, há uma tendência para ter-se uma maior taxa de perda, independendo do número de sorvedouros. Além disso, é possível notar também que o aumento do número de sorvedouros contribui, em média, para diminuir a taxa de perda. Isso pode ser explicado considerando que quanto maior o número de sorvedouros uniformemente distribuídos em uma região, menores são as árvores de *cluster-heads* associadas que possuem cada um dos sorvedouros como raiz. Desse modo, cada *cluster-head* tende a ter menos filhos e, conseqüentemente, receber menos transmissões sujeitas a perdas provocadas pela limitação de *hardware* já mencionada. Uma outra justificativa para as altas taxas de perda mostradas na Figura 42 pode ser dada considerando que foi adotado um padrão de tráfego intenso (t = 5) em todas as rodadas de simulação que geraram esse resultado. Adicionalmente, confirmando

tais constatações, pode-se perceber através desse gráfico, que a rede com menor número de nós analisada (100) apresentou o melhor desempenho em termos de perda de pacotes e a rede com maior número de nós (200) apresentou o pior desempenho, independente do número de sorvedouros.

Finalmente, ainda em relação a essas curvas (Figura 40,Figura 41 e Figura 42), deve-se observar que a distribuição dos sorvedouros na área de monitoração não foi rigorosamente uniforme conforme explicado na seção 4.3.1, onde são descritos os cenários com múltiplos sorvedouros. Assim, padrões como, por exemplo, aquele que indica (Figura 40) uma tendência de queda no atraso para cenários específicos de três e cinco sorvedouros, podem ser justificados usando o argumento de que tais distribuições de sorvedouros foram geometricamente privilegiadas.

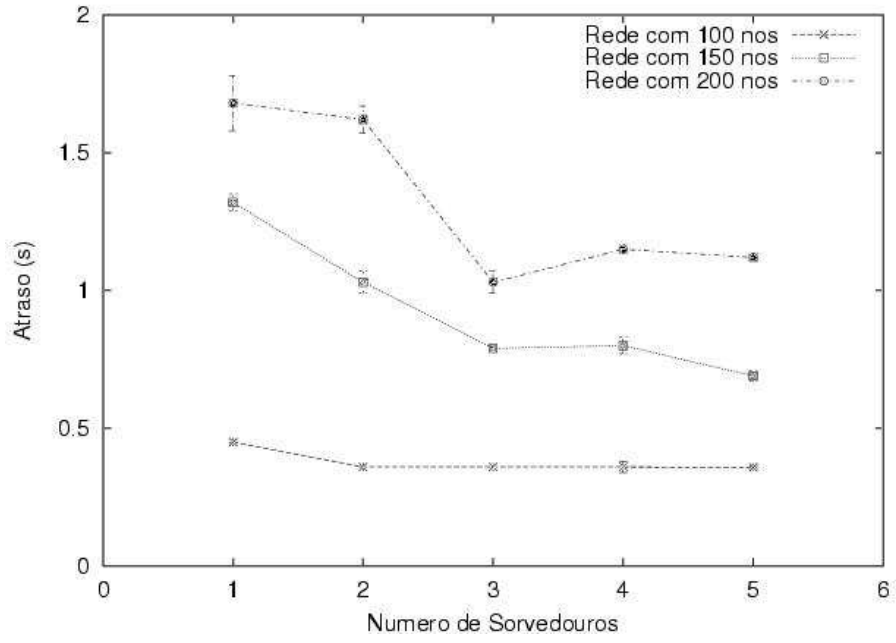

Figura 40- Atraso (s) - 100,150 e 200 nós, [1-5] Sorvedouros

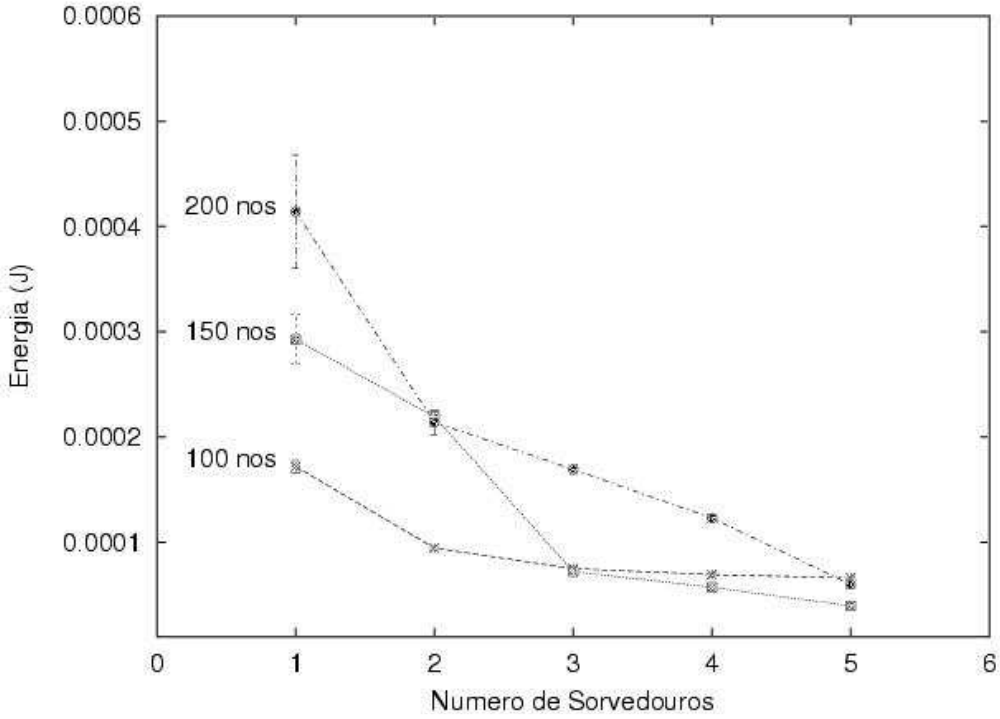

Figura 41- Energia Média Dissipada(J) - 100,150,200 nós, [1-5] Sorvedouros

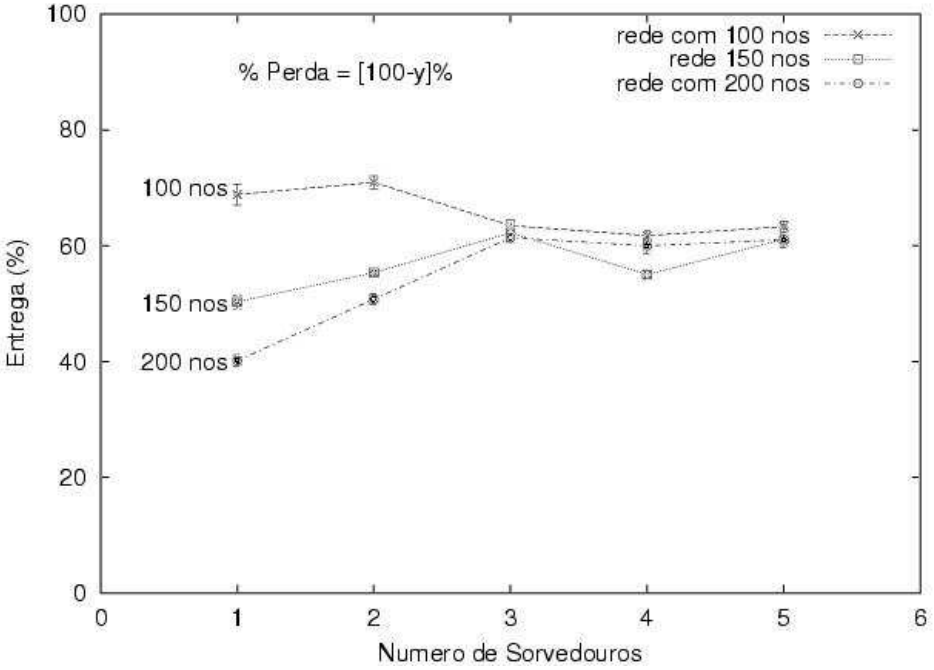

Figura 42 - Taxa de Perda (%) - 100,150,200 nós [1-5] Sorvedouros

4.5 Considerações Finais

As simulações efetuadas atestaram a eficácia do protocolo proposto em termos de energia quando comparado aos protocolos LEACH e *Directed Diffusion*, apesar de ter-se percebido um atraso superior ao do protocolo *Directed Diffusion*. Além disso o consumo mostrou-se uniformemente distribuído nos nós da rede, constatando-se também que, para redes maiores (250 nós) sob alto tráfego, ocorreu um pico de consumo de energia nos cenários em que o mecanismo de recuperação de falhas estava desativado. Tal mecanismo, quando ativado, proporcionou ainda um aumento na cobertura média da rede e uma redução na taxa de perdas de pacotes.

Para cenários com múltiplos sorvedouros, assumindo que os sorvedouros estão uniformemente distribuídos na área de monitoração, constatou-se que o consumo médio de energia e o atraso médio na entrega de dados são inversamente proporcionais ao número de sorvedouros, independente do tamanho da rede.

Finalmente, foi observado que o protocolo proposto não se mostra adequado para cenários de monitoração onde a quantidade de tráfego gerada é mais intensa do que (t=30), havendo uma excessiva perda de pacotes.

Capítulo 5 Conclusão e Trabalhos Futuros

Entre os principais objetivos do protocolo proposto, pode-se destacar: (i) a melhoria em termos de consumo de energia em relação ao protocolo LEACH, (ii) a eliminação da restrição imposta pelo LEACH de que o nó sorvedouro tem que estar nos raios de alcance de todos os nós, (iii) a criação de um mecanismo para recuperação de falhas, e (iv) a criação de um protocolo híbrido quanto ao método de coleta de dados, capaz de atender tanto às aplicações orientadas a eventos quanto às aplicações periódicas.

O protocolo LEACH, ao empregar a comunicação direta entre os *cluster-heads* e o sorvedouro, gera gastos excessivos de energia. Assim, baseado nessa constatação, adotou-se no protocolo proposto, uma topologia hierárquica diferenciada, que consiste na formação de um *backbone* de *cluster-heads*, sem a necessidade de sincronização global, para encaminhar informações dos seus *clusters* através de múltiplos saltos, em direção ao sorvedouro. Desse modo, além de ter sido criada a possibilidade de se economizar energia explorando as transmissões de curto alcance, foi possível também eliminar o vínculo existente entre o limite máximo para o tamanho da área de monitoração e o alcance do rádio. A eliminação de tal vínculo viabilizou o uso do protocolo proposto para monitorar áreas de diâmetros maiores que o alcance do rádio.

Em termos de tolerância a falhas, criou-se um mecanismo que consiste na eleição de um nó *cluster-head* reserva para cada *cluster*, explorando as altas densidades de nós típicas nas RSSFs. O nó reserva tem a função de substituir o *cluster-head* corrente em caso de falha. Esse esquema se mostrou eficaz para a redução da taxa de perda de dados, haja vista que a recuperação de falhas nos *cluster-heads* é realizada imediatamente, não necessitando esperar um novo ciclo de reorganização topológica da rede. Um outro aspecto importante desse esquema de recuperação de falhas é o método usado na eleição dos *cluster-heads* reservas. Dois foram os métodos adotados: método aleatório e método diferenciado. O método aleatório, como o próprio nome indica, consiste em uma eleição aleatória, tendo como candidatos todos os nós do *cluster* em questão. O método diferenciado considera apenas como candidatos os nós com nível residual de energia acima da média e, entre esses, o escolhido é aquele com menor quantidade de energia. Assim, caso não ocorra falha de um *cluster-head* durante a fase de coleta / encaminhamento, o nó reserva preserva sua energia, pois permanece boa parte desse tempo em modo de baixo consumo.

O protocolo proposto, devido à natureza híbrida, isto é, adequado tanto para aplicações orientadas a eventos quanto para aplicações periódicas, aumentou a gama de possíveis aplicações, permitindo, entre outras coisas, que apenas os nós localizados nas áreas de interesse sejam mantidos ligados, consumindo energia em suas tarefas de monitoração. Quanto aos outros nós, localizados fora dessas áreas, o protocolo permite que sejam mantidos em um modo de baixo consumo, aumentando a vida útil da rede.

Embora na literatura tenham sido propostos outros protocolos [29,30,51] visando melhorias sobre o protocolo LEACH, a solução proposta no presente trabalho mostrou-se diferenciada (melhor) em relação as demais, considerando que apresentou características capazes de suprir algumas das principais limitações criadas ou mantidas pelas soluções alternativas. Dentre tais limitações, pode-se destacar a inadequação no atendimento a aplicações orientadas a eventos e periódicas (Híbrido) em um mesmo protocolo [51], a permanência do vínculo entre o tamanho máximo da área de interesse e o alcance do rádio [30,51], e a inexistência de um mecanismo de tolerância a falhas mais eficiente [29], assim como um maior compromisso entre o retardo de entrega dos dados e o consumo de energia [51].

Após as modificações e inclusões no código do simulador *ns-2* e posterior validação, as simulações do protocolo proposto possibilitaram tecer comparações com os resultados obtidos na simulação de outros protocolos. Além disso, foram confrontados os resultados obtidos em diferentes cenários, ajustando os parâmetros de execução para encontrar as melhores configurações de cada cenário. Isto permitiu também que pontos negativos fossem identificados. Dentre os principais resultados obtidos, destacam-se aqueles relacionados às seguintes métricas: energia média dissipada, cobertura e taxa de perda.

Em termos de energia, constatou-se que o protocolo proposto apresenta um menor consumo quando comparado aos protocolos LEACH e *Directed Diffusion*, confirmando a eficácia da topologia hierárquica diferenciada como estratégia de organização da rede, assim como o benefício trazido pelo esquema híbrido adotado para disseminação de dados na rede. Como exemplo, pode-se citar uma redução na energia média dissipada pelo protocolo proposto em relação ao *Directed Diffusion* de até 60% em aplicações orientadas a interesse. Quanto à comparação realizada do protocolo proposto com o protocolo LEACH, foram obtidos ganhos em termos de energia total dissipada onde o protocolo proposto gastou até quatro vezes menos energia, dependendo do tamanho da rede.

Em termos de cobertura da rede e taxa de perdas de dados, obteve-se ganhos relevantes através da adoção do mecanismo de tolerância a falhas pelo protocolo proposto quando

comparado com a sua não adoção. Por exemplo, foram obtidos aumentos de até 100% na cobertura média da rede. Além disso, tal mecanismo, independente do processo de reorganização periódico da rede, também proporcionou taxas de perda de dados menores em até 50%, visto que foi criada uma solução de contingência imediata para os *clusters* em caso de falha em seus *cluster-heads*, não gerando um período de indisponibilidade no *cluster* em questão até a fase de rotação seguinte, como ocorre no protocolo LEACH.

Entre as limitações identificadas no protocolo proposto, destaca-se aquela que diz respeito ao fato de tal protocolo, da maneira como foi implementado, com *slots* TDMA de tamanho fixo para toda a rede, não ser recomendado para ambientes de tráfego intenso. Em outras palavras, foi constatado que o protocolo proposto apresenta uma alta taxa de perda de dados em situações de sobrecarga de tráfego na rede conforme resultados das simulações. Tais situações de alta intensidade de tráfego podem ser esperadas, por exemplo, quando uma área de interesse estiver submetida à ocorrência de fenômenos ambientais que gerem uma grande quantidade de dados (eventos) a serem transmitidos para o nó sorvedouro. Nesse caso, a grande quantidade de tráfego gerada provoca a perda de dados, pois há um crescimento acelerado das filas de transmissão dos *cluster-heads* devido à exaustão da banda dos enlaces referentes ao *backbone* de *cluster-heads*. O emprego de diversos sorvedouros e de agregação de dados poder ser o primeiro passo para contornar essa limitação. Outra solução seria a implantação de um *backbone* de *cluster-heads* reservas para o balanceamento do tráfego em situações de sobrecarga, sendo formado por uma topologia disjunta em relação ao *backbone* principal. Uma última sugestão para resolver esse problema seria a adoção de um esquema de *slots* TDMA de tamanhos e/ou quantidades variáveis, onde seriam alocados dinamicamente *slots* maiores (ou em maior quantidade) para as áreas que estivessem apresentando maior intensidade de tráfego, aumentando a banda disponível nos enlaces localizados nessas áreas de sobrecarga.

Um outro aspecto negativo da solução proposta diz respeito à complexidade dos nós sensores, vale ressaltar que, para o funcionamento correto do protocolo, é necessário que seja incorporado aos nós um mecanismo de localização geográfica, pois a informação de posicionamento é fundamental no processo de agrupamento (*clustering*) durante a fase de inicialização.

Quanto aos trabalhos futuros, o protocolo proposto apresenta alguns pontos que ainda podem ser aprimorados ou previstos. Dentre eles, pode-se destacar o aprimoramento dos mecanismos de tolerância a falhas, de organização dos *clusters* e de rotação dos *cluster-heads*, além de ser prevista a mobilidade dos sorvedouros.

Em relação ao mecanismo de tolerância a falhas proposto, pode-se dizer que o seu funcionamento limita-se à detecção de falhas ocorridas nos nós *cluster-head* propriamente ditos, não considerando falhas de comunicação entre os *cluster-heads* e seus respectivos *cluster-head* pais, por exemplo. Essas falhas também precisam ser tratadas, pois, assim como as falhas dos nós *cluster-head*, criam situações de partição na rede, podendo ser causadas, por exemplo, por obstáculos móveis localizados na área de interesse. Uma solução para amenizar esse problema seria a utilização do *backbone* de *cluster-heads* reserva sugerido anteriormente para o problema da sobrecarga de tráfego.

Em relação à solução que foi proposta para a organização dos clusters, onde cada *cluster* mantém a sua composição original em termos de nós até o fim da vida útil da rede, fez-se com que uma característica potencialmente útil do protocolo LEACH deixasse de ser usada. Essa característica diz respeito à componente aleatória que o LEACH introduz para reorganizar periodicamente a composição de cada *cluster*, fazendo com que a rede "esqueça" periodicamente qualquer informação sobre a topologia anterior e continue seu funcionamento como se tivesse acabado de ser instalada. Esse esquema pode ser intuitivamente considerado ideal no que se refere à uniformização do consumo de energia dos nós da rede. Desse modo, sugere-se que seja adicionada e avaliada uma nova funcionalidade em relação ao protocolo proposto. Essa funcionalidade deveria permitir um *reset* periódico da rede, fazendo com que o protocolo reinicie o seu funcionamento a partir da fase inicial de configuração. Com isso, seria possível avaliar se a retirada do fator aleatório trouxe algum impacto no consumo de energia dos nós.

Quanto ao mecanismo de rotação dos cluster-heads adotado no protocolo proposto, que consiste em um processo de eleição centralizado no *cluster-head* corrente, sugere-se uma solução distribuída, assim como é feito no protocolo LEACH, visando prover maior robustez em relação às falhas que podem ocorrer no *cluster-head* corrente, durante o seu processo de eleição. Isso pode trazer benefícios visto que, no protocolo proposto, esse tipo de falha leva o *cluster* em questão a uma situação de isolamento definitivo. Uma solução alternativa seria a criação de um processo local de elegibilidade do *cluster-head* (envolvendo somente os nós do *cluster*) disparado após um tempo pré-definido a partir do início da fase de rotação. Então, durante o processo de eleição, caso o nó *cluster-head* corrente falhe, não anunciando o novo *cluster-head*, os outros nós do *cluster* iniciam o processo de elegibilidade de maneira distribuída para definir o novo *cluster-head*. Adicionalmente, para aumentar ainda mais o grau de resiliência, o processo de eleição do novo *cluster-head* poderia ser feito paralelamente

pelo *cluster-head* reserva corrente, neste caso, o isolamento do *cluster* só ocorreria quando da falha simultânea do *cluster-head* corrente e do seu reserva.

Quanto à mobilidade dos nós sorvedouros, que estava fora do escopo do presente trabalho, sugere-se a criação de um mecanismo que permita uma checagem periódica da alcançabilidade entre os nós filhos do sorvedouro na árvore de *cluster-heads* e o sorvedouro. Usando tal mecanismo, caso fosse detectado que o sorvedouro moveu-se e tal comunicação foi inviabilizada, o protocolo deveria ser reiniciado. Isto inevitavelmente impactaria de forma negativa o desempenho do protocolo devido a um maior consumo de energia e uma maior perda de dados.

Finalmente, devido à incompatibilidade nas metodologias de avaliação de desempenho e cenários de simulação, não foi possível comparar o desempenho da presente proposta com aqueles obtidos pelas propostas que tentaram melhorar o protocolo LEACH (ICA, PEGASIS, TEEN). Entretanto, analisando qualitativamente, é fácil concluir que, por exemplo, o protocolo ICA, apesar dos ganhos auferidos em termos de energia, tende a apresentar retardos na entrega de dados maiores do que os obtidos com o protocolo proposto. Então, uma linha de estudo para ampliar a avaliação da presente proposta seria implementar e validar esses protocolos no *ns-2* utilizando os mesmos critérios de avaliação e cenários de simulação. Uma outra questão que não foi abordada ao longo do presente trabalho bem como nos protocolos avaliados, diz respeito à implementação da fase de organização inicial da rede, na qual o nó sorvedouro obtém informações de posicionamento e quantidade de energia para definir a estrutura global da rede. Ainda durante essa fase, são realizados os procedimentos para garantir a alcançabilidade entre nós de *clusters* adjacentes que é vital para o correto funcionamento do protocolo proposto. Então, duas tarefas podem ser consideradas como trabalhos futuros para se ter uma situação mais realista da aplicabilidade do protocolo proposto. A primeira é a medição da energia gasta durante essa fase e a segunda é verificar, através de simulação, o impacto da densidade de nós em função do raio de alcance do rádio na garantia da alcançabilidade.

Referências

1. Estrin, D., Girod, L., Pottie, G. e Srivastava, M. *Instrumenting the world with wireless sensor networks*, In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 2001) Salt Lake City, Utah, Maio 2001.

2. Bulusu, N., Estrin, D., Girod, L. e Heidemann, J. *Scalable Coordination for wireless sensor networks: Self-Configuring Localization Systems*, In Proceedings of the Sixth International Symposium on Communication Theory and Applications (ISCTA 2001), Ambleside, Lake District UK, Julho 2001.

3. Sohrabi, K., Manriquez, B., Pottie, G. *Near-ground wideband channel measurements.* In Proceedings of the 49th Vehicular Technology Conference (Houston, Maio 16-20). IEEE, New York, 1999, pp. 571-574.

4. Shih, E., Cho, S., Ickes, N., Min, R., Sinha, A., Wang, A., and Chandrakasan, A., *Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks.* In Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (Rome, Italy). MobiCom '01. ACM Press, New York, NY, pp. 272-287, Julho 2001.

5. Kahn, J. M., Katz, R. H. e Pister, K. S. J. *Next century challenges: mobile networking for Smart Dust.* In International Conference on Mobile Computing and Networking (MobiCom '99), pp. 271-278, Agosto 1999.

6. Sohrabi, K.,Gao, J., Ailawadhi, V., Pottie, G. *Protocols for self-organization of a wireless sensor network*, IEEE Personal Communications, Outubro 2000, pp. 16–27.

7. Karvonen H, Pomalaza-Ráez C. *Coding for energy efficient multihop wireless sensor networks*. In: Proc. Nordic Radio Symposium 2004 / Finnish Wireless Communications.

8. Elson, J., e Estrin, D. *Random, Ephemeral Transaction Identifiers in Dynamic Sensor Networks,* Proceedings st International Conference on Distributed Computing Systems (ICDCS-21), Phoenix, Arizona, Abril 2001.

9. Akyildiz, I.F., *Wireless Sensor Networks: a survey*, Computer Networks (Elsevier), Março 2002.

10. Estrin, D., Govindan, R., Heidemann, J. e Kumar, S. *Next century challenges: scalable coordination in sensor networks*, ACM MobiCom'99, Washingtion, USA, Agosto, 1999, pp. 263–270.

11. Perrig, A., Szewczyk, R., Wen, V., Culler, D. e Tygar, J.D. *SPINS: security protocols for sensor networks*, Proceedingsof ACM MobiCom'01, Rome, Italy, Julho, 2001, pp. 189–199.

12. Savvides, A., Han, C. e Srivastava, M. *Dynamic fine-grained localization in ad-hoc networks of sensors*, Proceedings ofACM MobiCom'01, Rome, Italy, Julho 2001, pp. 166–179.

13. C. Shen, C. Srisathapornphat e C. Jaikaeo, Sensor information networking architecture and applications, IEEE Personal Communications, Agosto 2001, pp. 52–59.

14. Suzuki H. e Tobagi, F.A. *Fast bandwidth reservation scheme with multi-link and multi-path routing in ATM networks.* In Proceedings of the IEEE Infocom, 1992.

15. Zappala, D. *Alternate path routing for multicast.* In Proceedings of the IEEE Infocom, March 2000.

16. Ishida, K., Kakuda, Y. e Kikuno, T. *A routing protocol for finding two node-disjoint paths in computer networks*. In International Conference on Network Protocols, pp. 340 – 347, Nov 1992.

17. Maxemchuk, N.F. *Dispersity routing in high-speed networks*. In Computer Networks and ISDN systems, volume 25, pages 645–661, Janeiro, 1993.

18. Moy, J. *The OSPF specification*. In RFC 1583, Outubro, 1994.

19. Zaumen, W.T. e Garcia-Luna-Aceves, J.J. *Shortest multipath routing using generalized diffusing computations.* In Proceedings of the IEEE Infocom, Março, 1998.

20. Banerjea, A. *Simulation Study of the Capacity Effects of Dispersity Routing for Fault Tolerant real-Time Channels.* In ACM Computer Communications Review, volume 26, pages 194–205. ACM Press, Outubro, 1996.

21. Park, V. D. e Corson. M. S. *A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks*. In Proceedings of INFOCOM 97, pages 1405–1413, Abril 1997.

22. Ganesan, D., Govindan, R., Shenker, S. e Estrin, D. *Highly resilient, energy efficient multipath routing in wireless sensor networks.* ACM Mobile Comput. and Commun. Review, 5(4), pp. 11-25, Outubro, 2001.

23. Akkaya, K. e Younis, M. *Energy-aware Delay-Constrained Routing in Wireless Sensor Networks*, International Journal of Communication Systems,Volume 17, Issue 6 , pp. 663-687, Agosto, 2004

24. Ye, F., Luo, H.,Cheng, J., Lu, S. e Zhang, L. *A Two tier Data Dissemination Model for Large-scale Wireless Sensor Networks*, ACM/IEEE Mobicom 2002, pp. 148-159, Setembro, 2002.

25. Kim, H. S., Minimum-Energy Asynchronous Dissemination to Mobile Sinks in Wireless Sensor Networks, ACM SenSys, Los Angeles, CA, Novembro, 2003.

26. Intanagonwiwat, C., Govindan, R. e Estrin, D., *Directed diffusion: a scalable and robust communication paradigm for sensor networks*, Proceedings of ACM MobiCom '00, Boston, MA, Agosto, 2000, pp. 56-67.

27. Kulik, J., Heinzelman, W. R. e Balakrishnan, H., *Negotiation-based protocols for disseminating information in wireless sensor networks*, Wireless Networks, Volume: 8, pp. 169-185, Março, 2002.

28. Heinzelman, W., Chandrakasan, A. e Balakrishnan, H., *Energy-Efficient Communication Protocol for Wireless Mi-crosensor Networks*, Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS '00), Janeiro, 2000.

29. Lindsey, S., Raghavendra, C., *PEGASIS: Power-Efficient Gathering in Sensor Information Systems*, IEEE Aerospace Conference Proceedings, 2002, Vol. 3, pp. 1125-1130.

30. Manjeshwar, A. e Agarwal, D. P., *TEEN: a routing protocol for enhanced efficiency in wireless sensor networks*, In 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, Abril 2001.

31. Manjeshwar, A. e Agarwal, D. P., *APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks*, Parallel and Distributed Processing Symposium., Proceedings International, IPDPS 2002, pp. 195-202.

32. Subramanian, L. e Katz, R. H., *An Architecture for Building Self Configurable Systems*, In the Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing, Boston, MA, August 2000.

33. Demirbas, M., Nolte, T., Arora, A. e Lynch, N., *A Hierarchy-based Fault-local Stabilizing Algorithm for Tracking in Sensor Networks*, 8th International Conference on Principles of Distributed Systems (OPODIS), Dezembro, 2004.

34. Gupta, G., e Younis, M., *Fault-Tolerant Clustering of Wireless Sensor Networks. Wireless Communications and Networking* (WCNC 2003). IEEE , Volume: 3 , 16-20 Março, 2003.

35. Kirkpatrick, S., Gelatt Jr, C. D. e Vecchi, M. P., *Optimization by Simulated Annealing*, Science, 220, 4598, 671-680, 1983.

36. Singh, S. e Tripathi, S., *A time-slotted-CDMA architecture and adaptive resource allocation method for connections with diverse QoS guarantees*, Wireless Networks, Volume: 9, p.479-494, Setembro 2003

37. Efrat, A. *et al*. (1998), *Fly Cheaply: On the Minimum Fuel-Consumption Problem*, Procs. of the Symposium on Computational Geometry, pp.143-145.

38. Frolik, J., *QoS Control for Random Access Wireless Sensor Networks*. In Proceedings of the IEEE WCNC2004, Atlanta, USA, Mar, 2004.

39. NS-2 (The Network Simulator version 2). Disponível em: http://www.isi.edu/nsnam/ns/

40. Heinzelman, W. *et al*, *The MIT uAMPS ns Code Extensions*, Disponível em: http://www-mtl.mit.edu/research/icsystems/uamps/leach

41. *The CMU Monarch´s Project Wireless and Mobility Extensions to ns.*, Disponível em: http://www.monarch.cs.cmu.edu/

42. Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, S. e Rubenstein, D., *Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet.* In Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X) (San Jose, CA, USA, Oct. 2002), ACM Press, pp. 96–107.

43. Wang, H., Estrin, D. e Girod, L., *Preprocessing in a tiered sensor network for habitat monitoring.* EURASIP JASP special issue of sensor networks, volume 4, pp. 392-401, Março, 2003.

44. Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D. e Anderson, J., *Wireless sensor networks for habitat monitoring*. In ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02), Atlanta, GA, Setembro, 2002.

45. Maroti, M., Simon, G., Ledeczi, A. e Sztipanovits, J., *Shooter Localization in Urban Terrain*, Computer, Vol. 37, No. 8, August 2004, pp. 60-61.

46. Burrell, J., Brooke, T. e Beckwith, R., *Vineyard Computing: Sensor Networks in Agricultural Production*, IEEE Pervasive Computing, Vol. 3, No. 1, Março 2004, pp. 38-45.

47. Leoncini, M., Resta, G., Santi, P., *Analysis of a Wireless Sensor Dropping Problem in Wide-Area Environmental Monitoring*, The Fourth International Conference on Information Processing in Sensor Networks (IPSN'05), Abril, 2005.

48. Gao, J., *Analysis of Energy Consumption for Ad Hoc Wireless Sensor Networks Using a Bit-Meter-per-Joule Metric*, IPN Progress Report 42-150. Agosto, 2002.

49. Lin, E., Rabaey, J. e Wolisz, A., *Power-Efficient Rendez-vous Schemes for Dense Wireless Sensor Networks*, In *Proc. of ICC 2004 Paris*, Paris,France, Junho 2004.

50. Aurenhammer, F. (1991), "Voronoi Diagrams - A Survey Of A Fundamental Geometric Data Structure", *ACM Computing Surveys* 23, pp.345-405.

51. Habib, E. *et al*, "ICA: Um Novo Algoritmo de Roteamento para Redes de Sensores". Simpósio Brasileiro de Redes de Computadores, 2004.

52. Hoesel, L., Chatterjea, S. e Havinga, P., *An Energy Efficient Medium Access Protocol for Wireless Sensor Networks*, ProRisc, Holanda, Novembro, 2003

53. Lazos, L. et al, *SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks*, ACM Transactions on Sensor Networks (TOSN), Volume: 1, pp.73-100, Agosto, 2005

54. Cormen, T., Leiserson C. e Rivest R., *(1990) Introduction to Algorithms*. MIT Press. ISBN 0-262-03141-8

55. Koushanfar, F., Potkonjak, M. e A. Sangiovanni-Vincentell. *Fault tolerance techniques for wireless ad hoc sensor networks*. Sensors, 2002. Proceedings of IEEE , Volume: 2, Junho, 2002.

56. Papadopoulos, A. e McCann, J., *2004. Towards the Design of an Energy-Efficient, Location-Aware Routing Protocol for Mobile, Ad-hoc Sensor Networks.* In *Proceedings of the Database and Expert Systems Applications, 15th international Workshop on (Dexa'04) - Volume 00* (August 30 - Setembro 03, 2004).

57. Galvão, B., Delicato, F., Pires, P., Carmo, L. e Pirmez, L., "A Flexible Cluster-Based Approach for Architecting Wireless Sensor Networks". International Network Conference (INC2004). Universidade de Plymouth - Inglaterra. Julho/2004.

58. Galvão, B., Delicato, F., Pires, P., Carmo, L. e Pirmez, L., "CLIN - A CLuster-based, INterest-oriented Protocol for Wireless Sensor Networks", Western Multi-Conference - Communication Networks and Distributed Systems, Modelling and Simlation Conference (CNDS 2003). San Diego, CA. Janeiro/2003.