

VITOR BARBOSA CARLOS DE SOUZA

**UMA IMPLEMENTAÇÃO DO PROTOCOLO DE ROTEAMENTO
MULTI-GEO PARA REDES DE SENSORES SEM FIO USANDO
QUADTREES**

Dissertação apresentada à
Universidade Federal de Viçosa, como
parte das exigências do Programa de
Pós-Graduação em Ciência da
Computação, para obtenção do título
de *Magister Scientiae*.

VIÇOSA
MINAS GERAIS – BRASIL
2009

VITOR BARBOSA CARLOS DE SOUZA

**UMA IMPLEMENTAÇÃO DO PROTOCOLO DE ROTEAMENTO
MULTI-GEO PARA REDES DE SENSORES SEM FIO USANDO
QUADTREES**

Dissertação apresentada à
Universidade Federal de Viçosa, como
parte das exigências do Programa de
Pós-Graduação em Ciência da
Computação, para obtenção do título de
Magister Scientiae.

APROVADA: 14 de OUTUBRO de 2009.

Ricardo do Santos Ferreira
(Coorientador)

Carlos de Castro Goulart
(Coorientador)

Raquel Aparecida de Freitas Mini

André Gustavo dos Santos

Mauro Nacif Rocha
(Orientador)

Aos meus pais Sebastião e Edirlene,
orientador Mauro, irmãos Vinícius e
Vladimir, amigos e Suzana.

AGRADECIMENTOS

- Aos meus pais, Sebastião e Edirlene, por toda a paciência e apoio.
- Aos meus irmãos, Vinícius e Vladimir, que sempre foram também grandes amigos.
- À minha companheira Suzana pelo amor, paciência e incentivo.
- Aos colegas do DPI, muitos dos quais se tornaram grandes amigos que levarei por toda minha vida.
- Aos amigos de república, principalmente Vladimir e Eduardo, pelo convívio e também ao Raphael e Luiz pelas hospedagens.
- Ao professor Mauro Nacif Rocha pela orientação neste trabalho e pela alta disponibilidade e paciência.
- Aos meus coorientadores Ricardo dos Santos Ferreira e Carlos de Castro Goulart.
- Aos demais professores e funcionários do DPI, especialmente ao Altino, pela simpatia e pelos serviços prestados.

BIOGRAFIA

Vitor Barbosa Carlos de Souza, filho de Sebastião Carlos de Souza e Edirlene Barbosa Carlos de Souza, brasileiro, nascido em 29 de abril de 1981, no município de Muriaé, no Estado de Minas Gerais.

No ano de 1998, concluiu o ensino médio na Escola São Paulo de Muriaé-MG. Em 2006, concluiu o curso de Ciência da Computação na Universidade Federal de Viçosa, UFV. No ano seguinte, ingressou no programa de pós-graduação em Ciência da Computação do Departamento de Informática da Universidade Federal de Viçosa – UFV, onde obteve o título de mestre, defendendo essa dissertação em outubro de 2009.

SUMÁRIO

LISTA DE TABELAS	vii
LISTA DE FIGURAS	viii
LISTA DE ACRÔNIMOS	x
RESUMO	xi
ABSTRACT	xii
1. Introdução	1
2. Trabalhos Relacionados	5
2.1. Quadtree.....	5
2.2. S-MAC	7
2.3. Simulador Hades	9
2.4. Multi	13
2.4.1. SID (Source-Initiated Dissemination)	13
2.4.2. EF-Tree (Earlier-First Tree)	15
2.4.3. Funcionamento do Multi	16
2.5. Multi-K	17
2.6. Multi-Geo	18
3. Multi-Q: A Versão Modificada do Multi-Geo	22
3.1. Introdução.....	22
3.2. Eleição Dinâmica do Nó Líder	24
3.3. Agrupamentos com Tamanho Desigual.....	27
3.4. Alcance Limitado das Difusões.....	32
3.5. O Problema da Localização.....	34
3.6. Funcionamento do Protocolo Multi-Q	35
4. Avaliação do Multi-Q	38
4.1. Metodologia	38

4.1.1. Introdução	38
4.1.2. O Simulador	39
4.1.3. Parâmetros da Simulação	43
4.2. Resultados.....	46
4.2.1. Definição dos Parâmetros do Protocolo	46
4.2.2. Avaliação do Consumo Energético	50
4.2.3. Avaliação da Taxa de Entrega	52
4.2.4. Avaliação da Latência	54
5. Conclusões e Trabalhos Futuros	56
6. Referências.....	59

LISTA DE TABELAS

2.1 Simuladores existentes e algumas de suas principais características (BECKER, 2007).....	11
4.1 Alguns parâmetros da simulação	45

LISTA DE FIGURAS

1.1	Componentes de um nó sensor (AL-KARAKI; KAMAL, 2004).	3
2.1	Ilustração de uma Quadtree (SAMET, 1984).	6
2.2	Interface do simulador Hades (HENDRICH, 2006).	12
2.3	Conexão entre componentes SimObject, Port e Signal (HENDRICH, 2006).	13
2.4	Comportamento do protocolo SID (FIGUEIREDO; NAKAMURA; LOUREIRO, 2004).	15
2.5	Comportamento do protocolo EF-Tree (FIGUEIREDO; NAKAMURA; LOUREIRO, 2004).	16
2.6	Estrutura de disseminação do protocolo Multi-K em modo proativo (GONÇALVES et al., 2008).	18
2.7	Exemplo de distribuição das regiões virtuais em uma matriz 4x4 (SILVA; GOULART, 2009).	20
3.1	Seleção do nó líder no Multi-Geo e no Multi-Q.....	25
3.2	Divisão de agrupamentos baseada no alcance dos nós.	29
3.3	Divisão de agrupamentos em um mesmo cenário com utilização de agrupamentos de tamanho fixo e variável com o uso de Quadtree.	30
3.4	Divisão de um agrupamento com distribuição de nós não uniforme. ...	31
4.1	Visão esquemática dos componentes implementados.....	41
4.2	Comparação entre as duas implementações da fila de eventos do simulador.....	42

4.3	Tipos de distribuição dos nós: (a) Distribuição uniforme; (b) Distribuição exponencial;	44
4.4	Variação do consumo de acordo com o número máximo de nós por agrupamento utilizando redes com 500 nós.....	47
4.5	Variação do consumo de acordo com o número máximo de nós por agrupamento utilizando redes com 1000 nós.....	48
4.6	Variação do consumo energético total em função do número de nós na rede utilizando diferentes posicionamentos para os nós líderes.	48
4.7	Variação da taxa de entrega de pacotes em função do número total de nós na rede utilizando diferentes posicionamentos para os líderes	49
4.8	Variação do consumo total de energia de acordo com o número total de nós na rede.	50
4.9	Variação do consumo energético total em função da probabilidade de falha dos nós.....	51
4.10	Comparação detalhada entre o consumo energético dos protocolos Multi-Geo, Multi-Geo 2 e Multi-Q.....	51
4.11	Variação da taxa de entrega de pacotes em função do número total de nós na rede.	53
4.12	Variação da taxa de entrega de pacotes em função da probabilidade de falha de cada nó.....	53
4.13	Variação da latência média dos protocolos de acordo com o número total de nós na rede.....	54
4.14	Variação da latência média dos protocolos em função da probabilidade de falha dos nós.	55

LISTA DE ACRÔNIMOS

CTS	Clear to Send
EF-Tree	Earlier-First Tree
GPS	Global Positioning System
Hades	Hamburg Design System
LEACH-C	Low-Energy Adaptive Clustering Hierarchy Centralized
MAC	Medium Access Control
NS-2	Network Simulator 2
QoS	Quality of Service
RSSF	Redes de Sensores Sem Fio
RTS	Request to Send
S-MAC	Sensor MAC
SID	Source-Initiated Dissemination
WAF	Wall Attenuation Factor

RESUMO

SOUZA, Vitor Barbosa Carlos de, M.Sc., Universidade Federal de Viçosa, outubro de 2009. **Uma implementação do protocolo de roteamento Multi-Geo para redes de sensores sem fio usando Quadrees.** Orientador: Mauro Nacif Rocha. Co-Orientadores: Ricardo dos Santos Ferreira e Carlos de Castro Goulart.

Este trabalho apresenta o *Multi-Q*, uma versão modificada do *Multi-Geo*, que é um protocolo adaptativo híbrido para Redes de Sensores Sem Fio (RSSFs) cujo comportamento é baseado no tráfego da rede em um dado momento. Ele usa um comportamento reativo quando a frequência de eventos é baixa e um comportamento proativo quando a frequência de eventos aumenta. A principal diferença entre a nova versão e a anterior é o uso de agrupamentos de diferentes tamanhos de acordo com o número de nós, assim, o tamanho de cada agrupamento é inversamente proporcional ao número de nós em sua área. Outras diferenças são a limitação no alcance das difusões dentro de um agrupamento e a eleição de um novo nó líder quando o atual falha. O objetivo desse protocolo é reduzir o número de mensagens de controle e de dados na rede, que são os principais responsáveis pelo consumo energético. Além disso, ele preserva o esquema de envio multi-salto que é uma outra estratégia efetiva para a economia no consumo energético da rede. Os resultados apresentados ao final deste trabalho mostram que este protocolo obteve uma redução no consumo energético em comparação com versões anteriores, mantendo a taxa de entrega de pacotes. Além desse protocolo, este trabalho também gerou uma ferramenta para simulação de RSSF através da criação de uma extensão para a ferramenta Hades.

ABSTRACT

SOUZA, Vitor Barbosa Carlos de, M.Sc., Universidade Federal de Viçosa, October of 2009. **An implementation of the Multi-Geo routing protocol for wireless sensor networks using Quadrees.** Adviser: Mauro Nacif Rocha. Co-Advisers: Ricardo dos Santos Ferreira and Carlos de Castro Goulart.

This work presents the *Multi-Q*, a modified version of *Multi-Geo*, which is an adaptive hybrid protocol for Wireless Sensor Network (WSN) whose behavior is based on the network traffic in a given time. It is set to a reactive behavior when the frequency of events is low and it is set to proactive behavior when the frequency of events increases. The main difference between the new protocol and the previous one is the use of different *cluster* sizes according to the number of nodes, thus, the size of each *cluster* is inversely proportional to the number of nodes in its area. Other differences are the range limitation of broadcasts inside a *cluster* and the election of a new *clusterhead* when the current one fails. The goal of this protocol is to reduce the number of control and data messages in the network, which are the main responsible for the energy consumption. In addition, it preserves the scheme of multi-hop sending that is another effective strategy for the economy in energy consumption of the network. The results presented at the end of this work show that this protocol obtained a reduction of energy consumption when comparing to earlier versions, maintaining the data packets delivery rate. Besides this protocol, this work also generated a tool for WSN simulation through the implementation of a Hades tool extension.

Capítulo 1 - Introdução

Os avanços na tecnologia móvel permitiram um maior uso de sensores para monitoramento de uma determinada área objetivando a detecção da ocorrência de um ou mais tipos de eventos e o envio de dados relativos a esses eventos para estações base responsáveis pelo processamento desses dados. Os dados gerados por esses sensores são propagados por uma rede de sensores sem fio (RSSF), sendo que, essas redes podem conter centenas ou milhares de nós sensores. Esse tipo de rede pode ter aplicações nas mais diversas áreas.

Em aplicações militares, por exemplo, pode-se detectar a intrusão de determinada área ou escanear um campo em busca de um determinado objeto. Em aplicações civis, pode-se, além disso, monitorar o clima, segurança, detectar condições ambientais e ajudar em resgates após ocorrências de desastres. A criação de uma RSSF pode ajudar em operações de resgate através da localização de sobreviventes, identificação de áreas de risco e determinação da situação geral da área de desastre (AL-KARAKI; KAMAL, 2004).

Assim, na implementação de uma RSSF podem ser usados diversos tipos de sensores, que podem possuir, entre outras, as capacidades de detecção de movimento, som, luz, temperatura, distância e presença. Esses sensores podem ser distribuídos deterministicamente ou não. Um exemplo de distribuição determinística é quando sensores são posicionados manualmente em intervalos de distância aproximadamente iguais com o objetivo de detectar a ocorrência de fumaça no interior de um edifício. Um

exemplo de uma distribuição não determinística é a utilização de um avião para arremessar milhares de nós sensores em uma mata aberta com o objetivo de monitorar distúrbios como incêndio ou alagamento, fazendo com que os nós sejam posicionados de maneira aleatória, podendo, ou não, haver a ocorrência de áreas com maiores quantidades de sensores do que outras.

Cada nó sensor é formado basicamente por uma unidade de sensoriamento, uma unidade de processamento, um transmissor e uma unidade de energia. A figura 1.1 ilustra um diagrama dos componentes de um sensor, incluindo componentes não obrigatórios, como um sistema de localização de posicionamento, um dispositivo de mobilidade e um gerador de energia para o sensor. Essa figura também ilustra a arquitetura de comunicação de uma RSSF, onde alguns nós sensores são distribuídos em uma determinada área, formando uma rede por onde os dados coletados trafegarão até atingir uma estação base externa, que pode ser um nó fixo ou móvel capaz de conectar a rede de sensores à uma infraestrutura existente ou à internet, onde um usuário poderá ter acesso aos dados coletados (AL-KARAKI; KAMAL, 2004). O nó responsável por fazer a comunicação entre a rede de sensores e o meio externo é comumente chamado de nó *sink*.

Para que os dados gerados sejam transportados até a estação base, onde eles podem ser processados, é usado uma rede *ad-hoc* multi-salto e, devido ao grande número de nós sensores necessários por algumas aplicações e aos tipos de aplicações que as redes de sensores devem suportar, os nós sensores devem, muitas vezes, ser pequenos e de baixo custo. Para atingir esses requisitos, os nós sensores tem importantes características como o uso de pequenas baterias, com pouca capacidade de energia, e potência de comunicação limitada.

Para uma rede de sensores sem fio funcionar com essas limitações, ao invés de simplesmente enviar dados brutos, em aplicações típicas, os nós em RSSF realizam, através de métodos computacionais, tomadas de decisões individualmente ou em agrupamentos locais. Através dessas

tomadas de decisões dentro da rede, a comunicação pode ser reduzida, prolongando o tempo de vida da rede (PARK; SAVVIDES; SRIVASTAVA, 2001).

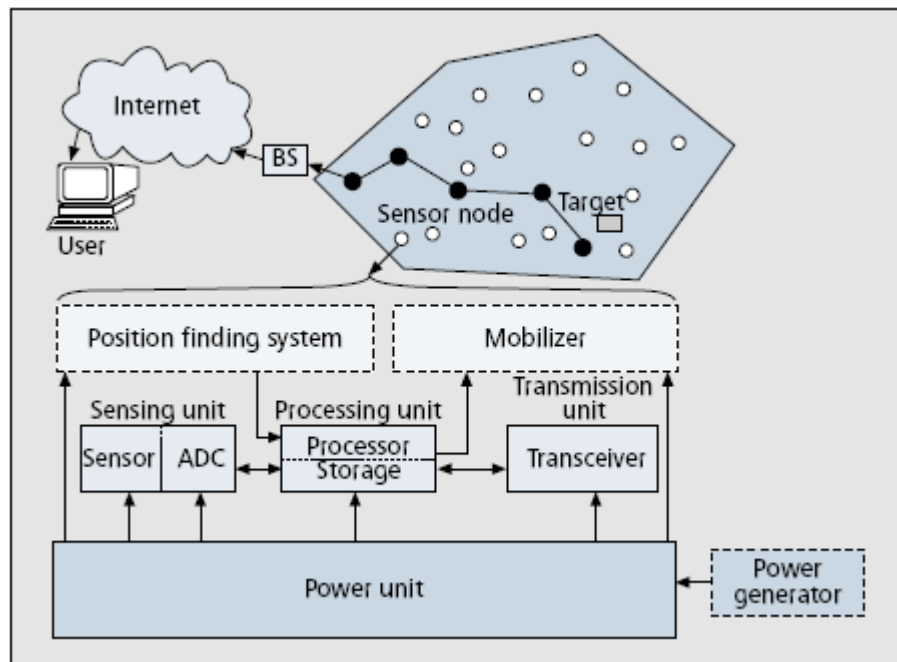


Figura 1.1. Componentes de um nó sensor (AL-KARAKI; KAMAL, 2004).

Essas tomadas de decisões são definidas através do desenvolvimento de protocolos específicos para RSSF, considerando todas as limitações inerentes a esse tipo de rede. Atualmente, existem muitos protocolos de roteamento implementados, cada um com suas vantagens e desvantagens e, a maioria deles são adequados para tipos específicos de redes, permitindo que o protocolo aproveite essas características. Este trabalho é baseado no protocolo *Multi-Geo* (SILVA; GOULART, 2009) e seus predecessores, que serão apresentados no próximo capítulo. A motivação para este trabalho está no fato de que o protocolo *Multi-Geo* apresenta algumas características que limitam o seu desempenho. Sendo assim, pretende-se neste trabalho propor modificações a esse protocolo, de modo a obter um novo protocolo com melhor desempenho, principalmente se tratando do consumo energético dos nós.

Os próximos capítulos deste trabalho estão organizados da seguinte forma. O capítulo 2 apresenta as principais pesquisas relacionadas a esta. A

seguir, é apresentado, no capítulo 3, o protocolo de roteamento proposto por este trabalho, destacando suas principais diferenças em relação aos demais protocolos nos quais ele foi baseado. Posteriormente, o capítulo 4 faz a validação do protocolo, apresentando a metodologia usada e uma sequência de resultados. Finalmente, no capítulo 5, é feita uma conclusão apresentando as vantagens e as limitações desse protocolo.

Capítulo 2 - Trabalhos Relacionados

Este capítulo apresenta a estrutura *Quadtree* (SAMET, 1984), utilizada neste trabalho, além do protocolo *S-MAC* (YE; HEIDEMANN; ESTRIN, 2004), utilizado para controle do acesso ao meio, o Simulador *Hades* e os protocolos de roteamento usados como base para este trabalho. Esses protocolos são o *Multi* (FIGUEIREDO; NAKAMURA; LOUREIRO, 2004) e seus sucessores *Multi-K* (GONÇALVES et al., 2008) e *Multi-Geo* (SILVA; GOULART, 2009), sendo que, seus comportamentos também são descritos assim como as principais mudanças realizadas em relação a seus antecessores e os resultados obtidos por essas mudanças.

2.1 Quadtree

A *Quadtree* (SAMET, 1984) é uma estrutura hierárquica de dados baseada no princípio da decomposição recursiva do espaço. Essa decomposição pode ser realizada, entre outros, sobre pontos, regiões, curvas e superfícies podendo ser feita com o uso de divisões em partes iguais em cada nível (i.e.: decomposição regular em polígonos) ou partes diferentes de acordo com os dados de entrada. A resolução da decomposição (i.e.: o número de vezes que o processo de decomposição é realizado recursivamente) pode ser fixa e definida a priori ou pode ser definida com base nas propriedades dos dados de entrada.

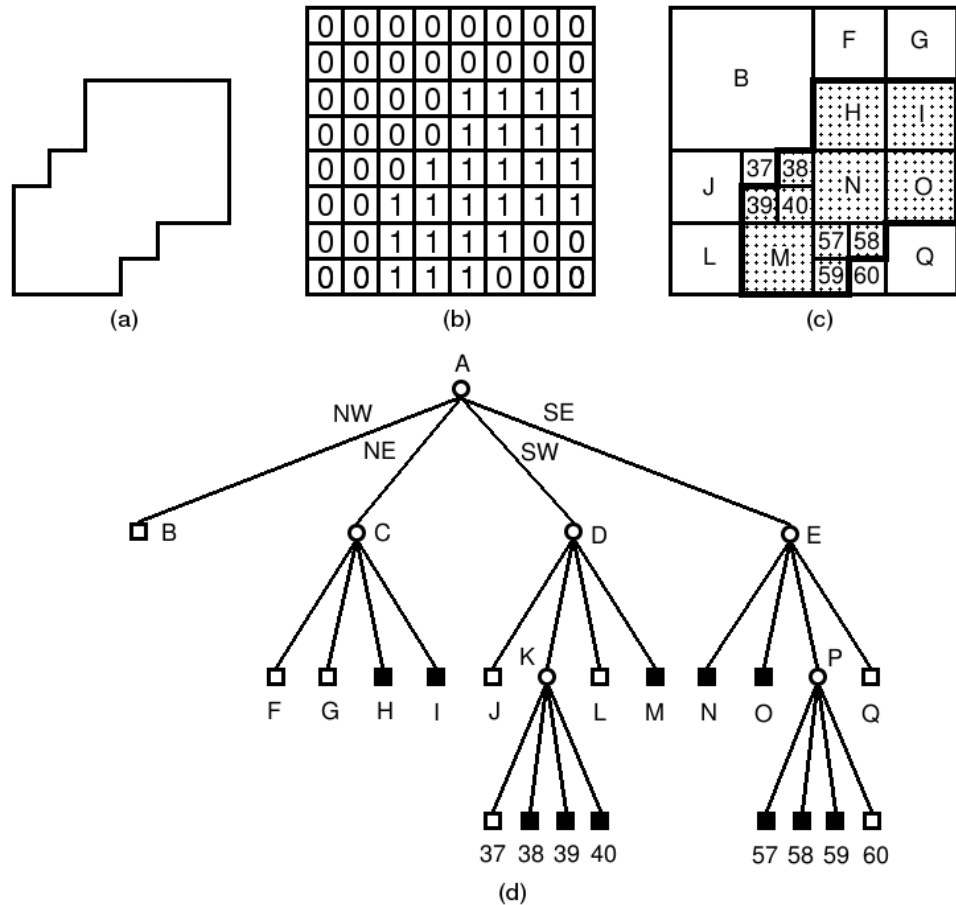


Figura 2.1. Ilustração de uma *Quadtree* (SAMET, 1984).

As estruturas *Quadtree* são geralmente usadas para divisão de espaços bidimensionais através da divisão recursiva de cada região em quatro quadrantes até que um parâmetro pré-estabelecido seja atingido. Dessa forma, uma *Quadtree* pode ser organizada como uma árvore, de modo que, o espaço original é representado pela raiz da árvore, enquanto que os quatro quadrantes são representados por 4 nós filhos. Cada nó filho, por sua vez, pode ser pai de outros 4 nós caso o quadrante representado por ele seja redividido em 4 quadrantes menores.

Desse modo, uma estrutura *Quadtree* pode ser representada por uma árvore onde cada nó pode ser um nó folha ou ter 4 nós filhos. A figura 2.1 mostra uma região aleatória e ilustra como é realizada a sua representação através de uma *Quadtree*. A figura 2.1(a) ilustra a região que deve ser representada pela *Quadtree*; a figura 2.1(b) exibe a representação matricial

da região; a figura 2.1(c) decompõe a região em blocos; e a figura 2.1(d) é a representação da região usando uma *Quadtree*.

2.2 S-MAC

O protocolo *S-MAC* (*Sensor MAC*) (YE; HEIDEMANN; ESTRIN, 2004) é um protocolo de controle do acesso ao meio desenvolvido especificamente para RSSFs com o principal objetivo de redução de consumo energético. Esse protocolo identifica as principais fontes de uso ineficiente de energia e inclui abordagens para tornar esse gasto mais eficiente. As principais fontes de gasto energético ineficiente identificadas foram as seguintes:

- *Colisões*: quando um pacote recebido está corrompido, ele deve ser descartado, sendo necessária a retransmissão do mesmo, aumentando o consumo energético;
- *Idle listening*: ocorre quando o nó permanece consumindo energia para escutar o meio esperando pela recepção de um pacote que não será enviado. Acontece durante períodos de inatividade da rede;
- *Overhearing*: ocorre quando um nó gasta energia para receber pacotes que não são destinados a ele;
- *Overhead* de mensagens de controle: apesar dessas mensagens geralmente serem de tamanho reduzido, a transmissão de muitas mensagens de controle podem representar um gasto de energia significativo.

Como a técnica usada pelo protocolo IEEE 802.11 para evitar colisões é muito eficiente, ou seja, ela consegue evitar grande parte das colisões sem a necessidade do uso de um rádio *full duplex*, o protocolo *S-MAC* usa procedimentos similares incluindo verificação física e virtual do meio de transmissão e a troca de pacotes RTS (*Request To Send*) e CTS (*Clear To Send*) para o problema do terminal escondido.

A verificação física do meio é realizada pela camada física e consiste em escutar o canal para verificar a existência de transmissões. A verificação

virtual do meio usa as mensagens recebidas para deduzir a existência de transmissões entre os vizinhos. Cada pacote enviado possui um campo que indica o tempo restante estimado até o final da transmissão da mensagem, sendo assim, quando um nó recebe um pacote destinado a outro nó, ele consegue saber durante quanto tempo o meio permanecerá ocupado. O meio é considerado livre apenas se as duas verificações (física e virtual) indicarem que ele está livre.

O uso de um campo informando a duração da transmissão da mensagem também permite que o *overhearing* seja evitado, bastando para isso, que o nó sensor durma durante o tempo da transmissão da mensagem após receber um pacote que não é destinado a ele. No protocolo IEEE 802.11, cada nó permanece escutando a todas as transmissões de seus vizinhos e, como resultado, cada nó gasta muita energia recebendo vários pacotes que não são destinados a ele e que serão descartados.

Baseado no protocolo *PAMAS* (SINGH; RAGHAVENDRA, 1998), o protocolo *S-MAC* faz com que os nós entrem no estado *dormindo* após receber um pacote RTS ou CTS destinado a outro nó. Como em muitas aplicações RSSFs esses pacotes costumam ser bem menores que os pacotes de dados, essa prática evita que um nó gaste energia recebendo grandes pacotes de dados que não são destinados a ele.

Para evitar o *overhead* de pacotes de controle, o *S-MAC* divide as grandes mensagens em fragmentos menores e os transmite em rajadas, mas ao contrário do protocolo IEEE 802.11, que também tem suporte a fragmentação de mensagens e usa os pacotes RTS e CTS para reservar o meio apenas para o envio do primeiro fragmento e o primeiro ACK, o *S-MAC* usa apenas um pacote RTS e um CTS para transmitir todos os fragmentos e seus respectivos ACKs. Para cada fragmento transmitido, o nó origem espera receber um ACK do nó destino para enviar o próximo fragmento. Se um ACK não é recebido, o tempo reservado para transmissão será estendido e o último fragmento enviado será retransmitido a seguir.

Como citado anteriormente, todos os tipos de pacotes possuem um campo indicando o tempo restante estimado até o final da transmissão do último fragmento, incluindo o tempo para a recepção de todos os ACKs. Assim, quando o tempo de transmissão é estendido devido a alguma falha e os nós vizinhos acordam durante a transmissão, ou se um nó passa a receber os pacotes contendo os fragmentos no meio da transmissão de uma mensagem, eles poderão examinar o campo que indica o tempo restante até o final da transmissão e dormir novamente.

O pacote ACK enviado após cada fragmento tem o objetivo de evitar o problema do terminal escondido no caso de um nó acordar ou se juntar à vizinhança durante a transmissão. Se esse nó for vizinho apenas do nó que está recebendo os fragmentos, ele não escutará os pacotes sendo enviados, mas também poderá dormir quando escutar o pacote ACK.

Além de dormir ao receber pacotes destinados a outros nós, para diminuir o consumo energético com o *idle listening*, o *S-MAC* faz com que os nós sejam divididos em grupos virtuais de vizinhos que dormem periodicamente. Todos os nós vizinhos pertencentes a um mesmo grupo dormem e acordam ao mesmo tempo. Essa técnica é parecida com o modo *power-save* do protocolo IEEE 802.11, mas a principal diferença é que esse modo do IEEE 802.11 foi desenvolvido para redes de apenas um salto e o protocolo *S-MAC* foi desenvolvido para redes multi-salto (YE; HEIDEMANN; ESTRIN, 2004).

2.3 Simulador Hades

Apesar da disponibilidade de uma grande quantidade de simuladores para RSSFs, os simuladores livres existentes não conseguem alcançar uma boa performance ao realizar simulações com um grande número de sensores e, ao mesmo tempo, oferecer uma rápida curva de aprendizado. Dessa forma, muitos pesquisadores dessa área desenvolvem suas próprias extensões para os simuladores existentes (HEINZELMAN; KULIK; BALAKRISHNAN, 1999) ou até mesmo desenvolvem seus próprios simuladores (WOO; CULLER,

2001) para permitir a simulação e avaliação de variáveis importantes em suas pesquisas.

Um dos simuladores mais utilizados na atualidade é o NS-2, no entanto, além de apresentar uma curva de aprendizado lenta, esse simulador muitas vezes se torna muito lento ao simular redes com muitos nós sensores (XUE et al., 2007). Outros simuladores utilizados para simulação de RSSFs estão listados na tabela 2.1. Essa tabela também exibe algumas de suas características principais como custo, linguagem de programação utilizada, formato da descrição do cenário e disponibilidade de simulação de consumo energético. A tabela completa, com mais resultados e características, está disponível em (BECKER, 2007). Alguns desses simuladores como o OMNeT++, JProWler, Atemu, SensorSim e GloMoSim, entre outros, foram estudados e apresentaram diferentes inconvenientes como descontinuidade do projeto, falta de informações, dificuldade de aprendizado e falta de versões livres.

Dessa forma, neste trabalho optou-se por desenvolver uma alternativa aos simuladores existentes. Essa alternativa teve o objetivo principal de permitir uma implementação fácil dos protocolos utilizados em RSSFs, permitindo também que as simulações fossem executadas rapidamente mesmo em simulações contendo um número elevado de nós sensores.

O simulador *Hades* (HENDRICH, 2000), que foi adaptado para simulações de RSSFs, é um ambiente simples e portátil que permite aos usuários a implementação e a execução de simulações de circuitos digitais de um modo iterativo. Assim, não há a necessidade de escrever arquivos de entrada externos ou usar o ciclo: editar, compilar, simular e analisar (HENDRICH, 2006). Uma outra característica desse simulador é a possibilidade de exibir os resultados do processamento em tempo real, assim, ele dá uma boa aproximação do que é observado em sistemas eletrônicos reais, como ilustrado pela figura 2.2.

Tabela 2.1. Simuladores existentes e algumas de suas principais características (BECKER, 2007)

Simulation Tool	+Tool Extension (Tool Version)	License, Cost	Language of the tool	Language of the models	GUI, API etc.	Scenario description format	Parallel execution	Supports Energy Consumption Research
TOSSIM	tython (1.1.15cvs)	GPL	Java	nesC	GUI: TinyVIZ/ SimDriver	Python script	?	with extension Power TOSSIM
ns-2	BonnMotion, BonnTraffic, SensorSim	GPL	C++, OTcl	C++, OTcl	GUI: nam	OTcl	Pdns	
GloMoSim (Qualnet)	BonnMotion	GloMoSim: Academic, QualNet: Commercial	Parsec (C, C++)		Yes	?	Parsec	
OMNeT++	Sensim, Mobility Framework, NesCT, EWSNSIM	Academic Public License	C++, NED	C++, NED, nesC	GUI: TkEnv	NED	Yes: Parsim	
OPNET	Wireless Module	Commercial / University Program	C, C++	C, C++	GUI	GUI (?)	Yes	
Avrora		GPL	Java		No GUI	Plain Text		
SwarmNet/ Shawn		BSD	C++	C++	No GUI	-	No	
Netwiser		Commercial license or GPL			Eclipse Plugin			
COOJA		Apache	Java	C (or Java)	GUI		No	
GTSNetS		GPL	C++	C++	Qt based animation tool	C++	Yes, libsync library	

Diferente de muitos outros simuladores, onde os modelos devem ser escritos em uma alguma linguagem proprietária específica, o simulador *Hades* permite que os modelos sejam escritos em Java, possibilitando um acesso completo, por parte do desenvolvedor, às facilidades de uma linguagem orientada a objetos com uma rica biblioteca de classes, incluindo acesso à rede, gráficos e *interfaces online*, permite o desenvolvimento de *Web Applets* de modo que as simulações possam ser executadas em um navegador *web*, e além disso, permite o uso de programação orientada a aspectos, que aumenta a modularidade através da separação de requisitos transversais (KICZALES et al., 1997).

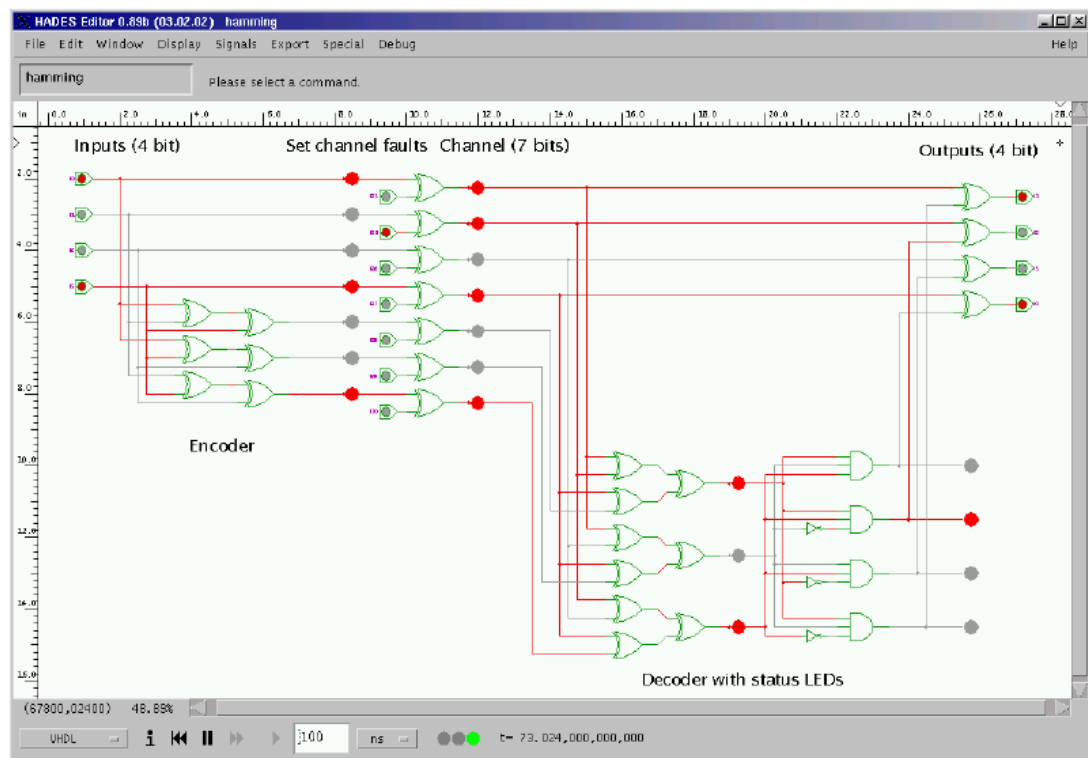


Figura 2.2. Interface do simulador Hades (HENDRICH, 2006).

O desenvolvimento de componentes no *Hades* também pode ser simplificado através da herança das classes base disponíveis, como *SimObject*, *SimEvent* e *Signal*, ou classes complexas, como editores de memória com muitas portas de entrada e/ou saída. A classe *SimObject* é uma classe genérica usada como base para todas as outras classes. Ela possui implementações padrão para quase todos os métodos, de modo que essas implementações são suficientes para simular quase todos os componentes padrões. Assim, geralmente, basta sobrecarregar o construtor da classe e alguns métodos específicos como *elaborate()* e *evaluate()*, sendo que o primeiro é chamado durante a inicialização do componente e o segundo descreve o comportamento do componente ao receber qualquer informação.

A simulação é executada pelo *kernel* de simulação, que interage com os outros componentes da simulação através de objetos do tipo *SimEvent* e, no editor gráfico, os objetos do tipo *Signal* são usados como componentes intermediários para manipular conexões entre os componentes da simulação

(HENDRICH, 2006). A figura 2.3 ilustra algumas dessas classes. O objeto *Signal* é transmitido através do componente *Port* de um *SimObject*. Os artigos (FERREIRA; CARDOSO; NETO, 2004) e (FERREIRA et al., 2005) são exemplos de trabalhos que também criaram extensões para o simulador *Hades*.

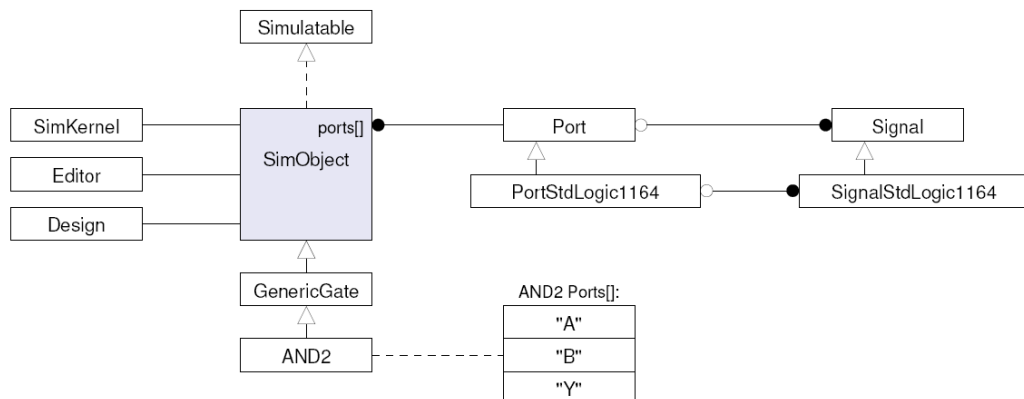


Figura 2.3. Conexão entre componentes *SimObject*, *Port* e *Signal* (HENDRICH, 2006).

2.4 Multi

O protocolo *Multi* (FIGUEIREDO; NAKAMURA; LOUREIRO, 2004) é um protocolo de roteamento adaptativo híbrido que muda seu comportamento de acordo com a quantidade de dados gerados pela rede, podendo operar de dois modos. O primeiro modo de operação tem um comportamento reativo e é usado em períodos de baixa ou nenhuma atividade da rede. O segundo modo tem um comportamento proativo e é usado em situações nas quais a ocorrência de eventos é grande gerando um tráfego intenso. Os dois modos de operação do protocolo *Multi* são apresentados a seguir.

2.4.1 SID (Source-Initiated Dissemination)

O modo de operação *SID* (FIGUEIREDO; NAKAMURA; LOUREIRO, 2004) usa um algoritmo reativo de disseminação de dados, ou seja, a disseminação é iniciada no nó de origem dos dados quando há a

necessidade do envio de pacotes para o nó *sink*. O envio dos dados é feito da seguinte forma:

- inicialmente é necessário construir um caminho por onde os pacotes de dados trafegarão, de modo que esses pacotes possam ser entregues de forma eficiente. Para isso, o nó fonte de dados faz a difusão de um pacote de controle informando a existência de dados a serem enviados. Esse pacote contém o *id* do nó fonte e o *timestamp* do dado gerado;
- cada nó que receber este pacote de controle deverá inserir em uma tabela de rotas as informações contidas no mesmo, além do *id* do nó que fez o envio do pacote, nesse caso, o próprio nó fonte. Em seguida, cada um desses nós deve refazer a difusão do pacote de controle;
- novamente, cada nó que receber o pacote de controle deverá armazenar as informações relativas ao mesmo em sua tabela de rotas, no entanto, apenas os nós que estão recebendo o pacote pela primeira vez farão isso, sendo assim, cada nó, ao receber o pacote, deve verificar se existe em sua tabela de rotas uma entrada com o *id* do nó fonte e o *timestamp* iguais aos contidos no pacote recebido já que essas duas informações funcionam como o *id* do dado originado no nó fonte. Caso o pacote não esteja sendo recebido pela primeira vez, o mesmo é descartado e, em caso negativo, o nó grava essas informações em sua tabela de rotas juntamente com o *id* do nó que realizou o reenvio do pacote;
- dessa forma, várias cópias do pacote original viajarão através da rede até atingir o nó *sink*, como ilustrado na figura 2.4(a). Cada nó intermediário (i.e.: cada nó que recebe esse pacote e refaz a difusão) armazena o *id* do dado que será enviado e o *id* do nó que fez o envio do pacote, assim, este nó é adotado como caminho para alcançar o nó fonte;
- quando o nó *sink* recebe o pacote informando a existência de dados para serem enviados, ele envia um pacote de requisição de dados contendo o *id* do dado sendo requisitado, ou seja, o *id* do nó fonte e o *timestamp* do dado. Dessa vez, o envio é feito diretamente para o nó adotado como caminho para alcançar o nó fonte e, do mesmo modo, cada nó

intermediário deverá localizar as informações relativas a esse dado em sua tabela de rotas e fazer o envio do pacote de requisição de dados diretamente ao nó que o liga ao nó fonte, como ilustrado na figura 2.4(b);

- da mesma forma, o caminho para alcançar o nó *sink* é criado bastando que cada nó atualize a entrada na tabela de rotas inserindo também o nó que o liga ao *sink*. Quando o nó origem recebe o pacote de requisição de dados, ele envia os pacotes de dados através do caminho construído para chegar ao *sink*, como mostrado na figura 2.4(c). O *sink* permanece enviando mensagens de requisição de dados periodicamente para que o nó fonte continue enviando os pacotes referentes àquele dado.

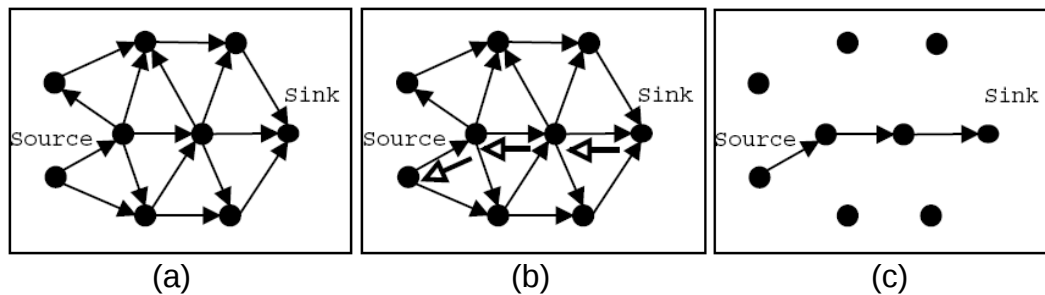


Figura 2.4. Comportamento do protocolo *SID* (FIGUEIREDO; NAKAMURA; LOUREIRO, 2004).

2.4.2 EF-Tree (Earlier-First Tree)

O modo de operação *EF-Tree* (FIGUEIREDO; NAKAMURA; LOUREIRO, 2004) usa uma estrutura de disseminação de dados simples e eficiente que consiste na construção de uma árvore cuja raiz é o nó *sink*. O processo de criação da árvore é realizado proativamente pelo nó *sink* e é descrito a seguir:

- o nó *sink* faz a difusão de um pacote de controle informando a construção da árvore. Cada nó, ao receber o pacote, assume o nó que enviou o pacote como nó pai e encaminha a mensagem através de uma difusão, como ilustrado na figura 2.5(a). Assim como ocorre no protocolo *SID*, os nós só reencaminham a mensagem quando ela é recebida pela primeira vez;

- dessa forma, quando um nó deseja enviar um dado, a estrutura de disseminação dos dados, ilustrada pela figura 2.5(b), já está pronta, bastando que cada nó envie os pacotes de dados para seu nó pai. Assim, não é necessário que cada nó fonte faça uma difusão para construção do caminho até o nó *sink*;
- o protocolo *EF-Tree* adota a reconstrução periódica da árvore, assim, ele pode acomodar possíveis mudanças na topologia da rede. Essa estrutura é criada proativamente e mantida pelo nó *sink* conectando todos os nós alcançáveis da rede.

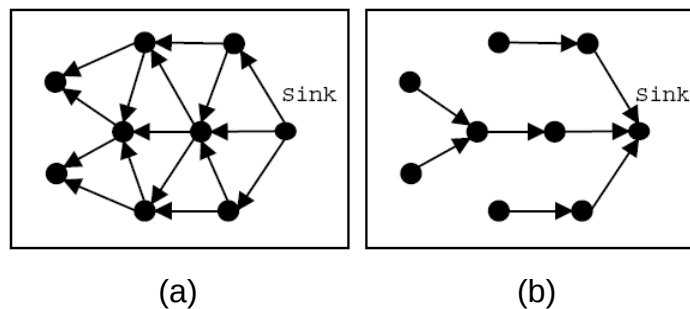


Figura 2.5. Comportamento do protocolo *EF-Tree* (FIGUEIREDO; NAKAMURA; LOUREIRO, 2004).

2.4.3 Funcionamento do Multi

Inicialmente, o protocolo *Multi* funciona como o protocolo *SID* (modo reativo). Quando o número de nós fonte ultrapassa um limite pré-definido, o nó *sink* faz a difusão de uma mensagem de construção da árvore informando que todos os nós devem usar o protocolo *EF-Tree* (modo proativo).

No decorrer do tempo, o número de nós fonte pode diminuir e, quando isso ocorrer, o nó *sink* para de enviar as mensagens de reconstrução da árvore. Assim, após um determinado intervalo de tempo sem receber essa mensagem de controle, os nós da rede passam a funcionar no modo reativo novamente. Caso ainda existam nós fonte, estes devem fazer a difusão de uma mensagem para construir o caminho até o nó *sink*.

Os algoritmos reativos, ao contrário dos algoritmos proativos, não precisam manter continuamente uma estrutura de disseminação. Em aplicações orientadas a eventos, é possível a ocorrência de longos períodos de inatividade da rede, sem a necessidade de disseminação de dados. Nesses casos, o protocolo *SID* tende a ser mais eficiente que o protocolo *EF-Tree*. Em cenários onde uma grande variabilidade da incidência de eventos é notada, o protocolo *Multi* demonstrou ser mais adequado do que algoritmos com uma única estratégia de roteamento (FIGUEIREDO; NAKAMURA; LOUREIRO, 2004).

2.5 Multi-K

O protocolo *Multi-K* (GONÇALVES et al., 2008) foi desenvolvido motivado pelo fato de que em muitas aplicações reais, os nós são distribuídos de modo que o nó *sink* fique localizado na área onde a concentração de eventos é maior. Com o uso dessa estratégia, o número médio de transmissões necessárias para levar uma mensagem até o nó *sink* é menor.

O protocolo *Multi-K* propõe uma melhoria no protocolo *Multi* através da limitação do tamanho da estrutura de roteamento usada no modo proativo. Com essa modificação, o tamanho da árvore construída pelo nó *sink* deve ser suficiente para cobrir todos os nós de origem de dados, mas quando não existem nós distantes do *sink* gerando dados, não é necessário criar e manter árvores grandes.

Para construir a árvore parcial de disseminação, o campo *K* foi inserido na mensagem de construção da árvore. Esse campo é um contador que representa o tamanho da árvore e é decrementado em uma unidade depois de cada salto. Quando o valor de *K* chega a zero, a mensagem de construção da árvore é descartada.

Para determinar o valor de *K*, as mensagens de dados, enviadas pelos nós fonte, também possuem um campo *K*, no entanto, nesse tipo de mensagem, o valor de *K* é incrementado a cada salto, ao invés de

decrementado. Dessa forma, o valor de K é igual ao tamanho do menor caminho que conecta o nó *sink* ao nó fonte de dados mais distante.

Assim, a árvore criada será menor, fazendo com que alguns nós não sejam cobertos pela estrutura de roteamento do *EF-Tree* e continuem usando o protocolo *SID*. A figura 2.6 mostra uma árvore com tamanho $K=3$.

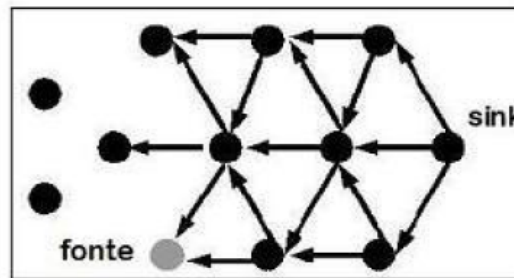


Figura 2.6. Estrutura de disseminação do protocolo *Multi-K* em modo proativo (GONÇALVES et al., 2008).

Essa abordagem mostrou ser mais eficiente em termos de consumo de energia do que o protocolo *Multi*, mantendo a taxa de entrega de pacotes e a robustez, principalmente em cenários onde os eventos ocorrem nas áreas próximas ao nó *sink*. Em cenários onde a ocorrência de eventos é imprevisível, o ganho desse protocolo em relação ao *Multi* é menor, já que a ocorrência de eventos em regiões distantes do nó *sink* pode provocar a criação de uma árvore que conecta todos os nós da rede, assim como ocorre no *Multi*. É importante destacar também que esses protocolos são diferentes apenas quando eles operam no modo de roteamento proativo.

2.6 Multi-Geo

Apesar da melhora no consumo energético alcançada pelo protocolo *Multi-K*, o consumo só é menor nos intervalos onde o protocolo está trabalhando no modo proativo. O *Multi-K* não sugere formas de melhoria no consumo energético durante intervalos onde o protocolo trabalha reativamente.

Além disso, as difusões, realizadas por cada nó da rede, são uma grande fonte de consumo energético. Essas difusões são feitas para

construir a árvore parcial quando o protocolo funciona proativamente e para construir o caminho até o nó *sink* quando o protocolo funciona reativamente.

O protocolo *Multi-Geo* (SILVA; GOULART, 2009) procura amenizar algumas limitações do *Multi-K*, utilizando para isso, informações geográficas como a localização dos nós e a divisão da área em várias regiões virtuais, criando agrupamentos de nós. Com a divisão da rede em agrupamentos pretendeu-se diminuir o número de envios de mensagens pelos nós da rede através de difusões. Para isso, para cada agrupamento é eleito um, e apenas um, nó líder. O nó líder é o nó responsável por fazer a comunicação entre os nós de seu agrupamento e os nós localizados em outros agrupamentos.

Com essa abordagem, quando um nó diferente do nó líder recebe uma mensagem através de uma difusão de um nó localizado em outra região virtual, ele não reenvia essa mensagem, apenas a ignora. Somente os nós líderes aceitam mensagens vindas de outros agrupamentos e, nesse caso, ele pode reenviar a mensagem para outros nós líderes e/ou enviar a mensagem para os nós de sua região. Quando um nó não-líder precisa enviar pacotes para fora de sua região, ele faz o envio diretamente para o nó líder, que repassa o pacote para os nós líderes vizinhos.

Dessa forma, durante a construção de uma árvore nesse protocolo, o nó *sink* faz a difusão de uma mensagem de controle que será aceita apenas pelos nós pertencentes ao seu agrupamento e pelos nós líderes dos agrupamentos vizinhos. Cada nó líder, após receberem a mensagem de construção da árvore, também fará uma difusão dessa mensagem, que também só será aceita pelos nós de seu agrupamento e pelos nós líderes vizinhos. Esse processo se repete sucessivamente até que a árvore seja criada.

A estratégia adotada para dividir os agrupamentos foi a criação de regiões virtuais com o mesmo tamanho, formando uma matriz de regiões virtuais. O número de regiões em que a área será dividida e, conseqüentemente, o tamanho de cada uma dessas regiões é definido a

priori por um novo parâmetro criado para este protocolo. A figura 2.7 ilustra uma matriz de regiões virtuais de tamanho 4x4 em uma área de 100x100 metros.

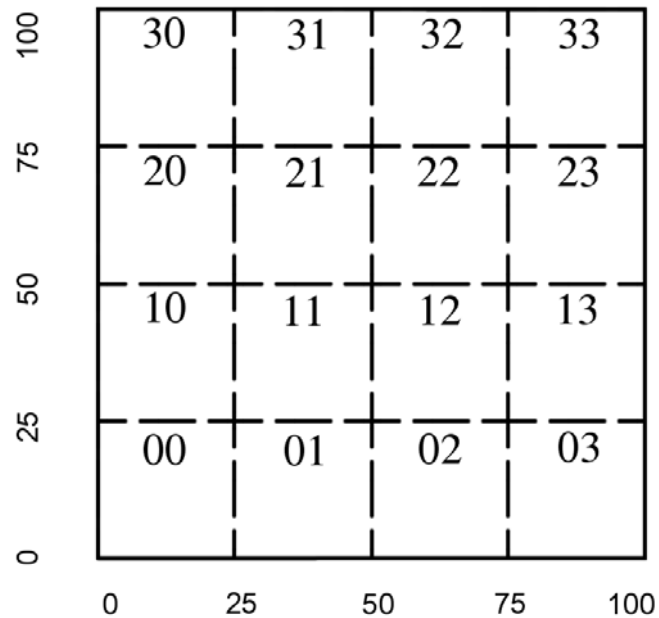


Figura 2.7. Exemplo de distribuição das regiões virtuais em uma matriz 4x4 (SILVA; GOULART, 2009).

A definição dos nós líderes, assim como a divisão dos agrupamentos, seguem a mesma forma centralizada de decisão do protocolo *LEACH-C* (LINDSEY; RAGHAVENDRA; SIVALINGAM, 2002), sendo assim, o nó *sink* deve conhecer a localização de cada nó da rede e é responsável por dividir virtualmente as regiões, criar identificadores para cada uma delas e informar cada nó da rede o identificador do agrupamento onde estão localizados. Além disso, ela é a responsável por eleger um nó líder para cada agrupamento.

Esse protocolo assume que os nós sensores são homogêneos e, sendo assim, a heurística utilizada para a definição do nó líder de uma região virtual considera apenas a posição de cada nó localizado nessa região. O nó líder deve ser o nó com menores coordenadas horizontais e verticais, ou seja, é o nó mais próximo do ponto com as menores coordenadas dentro de uma região virtual e o nó *sink* é o nó líder de seu

agrupamento. Uma outra característica da eleição dos nós líderes é que cada nó líder deve ser capaz de se comunicar com todos os nós líderes vizinhos, dessa forma, o tamanho de cada região virtual deve ter um limite máximo, que pode variar de acordo com o alcance máximo de transmissão dos nós.

A simulações realizadas por (SILVA; GOULART, 2009) mostraram que o protocolo *Multi-Geo* apresentou uma melhoria no consumo energético total da rede, mantendo a taxa de entrega de pacotes do protocolo *Multi-K* quando a probabilidade de falha dos nós não foi considerada. Essa probabilidade de falha não foi considerada porque esse trabalho assume que um nó falha quando a energia de sua bateria se esgota ou quando ocorre algum outro problema que exclui o nó da rede. Assim, como o protocolo *Multi-Geo* não realiza a eleição de um novo nó líder quando o atual falha, a taxa de entrega de pacotes tende a diminuir quando isso acontece já que os pacotes originados naquele agrupamento deixarão de ser entregues.

Capítulo 3 - Multi-Q: A Versão Modificada do Multi-Geo

3.1 Introdução

Como descrito no capítulo anterior, o protocolo de roteamento *Multi* (FIGUEIREDO; NAKAMURA; LOUREIRO, 2004) foi desenvolvido para obter um baixo consumo energético em situações onde o tráfego de dados na rede pode variar, incluindo a ocorrência de longos períodos de inatividade. Para atingir esse objetivo, esse protocolo apresenta duas formas de roteamento que podem alternar entre si de acordo com a variação do tráfego de dados na rede. A primeira abordagem é reativa e é usada em intervalos onde o número de nós fonte é pequeno ou nulo, enquanto que a segunda abordagem é proativa e é usada em intervalos onde o número de fontes aumenta, provocando um maior tráfego na rede.

Como o consumo energético é uma das principais métricas de QoS (*Quality of Service*) e assumindo que a estrutura de roteamento usada pelo *Multi*, quando o mesmo trabalha proativamente, consome mais energia que o necessário, o protocolo *Multi-K* (GONÇALVES et al., 2008), baseado no *Multi*, foi proposto e, através de uma limitação no tamanho da estrutura de roteamento usada, foi possível conseguir melhores resultados no consumo energético. Essa modificação foi motivada pelo fato de que em muitas aplicações reais, o nó *sink* fica localizado próximo à área com maior concentração de eventos e, dessa forma, a estrutura de roteamento usada no modo proativo precisa ser grande o suficiente apenas para abranger os

nós fonte, que estão próximos ao *sink*. Dessa forma, o protocolo consegue evitar um consumo energético desnecessário abrangendo nós que não enviarão dados.

Uma abordagem posterior, chamada de *Multi-Geo* (SILVA; GOULART, 2009), manteve as melhorias propostas pelo protocolo *Multi-K* e propôs a divisão da rede de forma a criar uma matriz de regiões virtuais (ou agrupamentos) iguais em termos de dimensões. Para cada agrupamento, é eleito um nó, denominado nó líder, responsável por fazer a comunicação entre diferentes regiões virtuais. Ao fazer com que os nós não-líderes enviem apenas mensagens unicast, permitindo que apenas os nós líderes enviem mensagens por difusão, foi possível reduzir o consumo total de energia, já que o número total de difusões realizadas pelos nós da rede foi diminuído significativamente.

Com o objetivo de melhorar algumas características do protocolo *Multi-Geo*, uma versão modificada do mesmo, chamada *Multi-Q*, está sendo proposta neste trabalho. As características do *Multi-Geo* que limitam seu desempenho e que foram alteradas na nova proposta são:

- o uso de nós líderes estáticos, que não são modificados mesmo quando um deles falha;
- o uso de agrupamentos com o mesmo tamanho organizados como uma matriz;
- o uso do alcance máximo do rádio dos nós durante a difusão de mensagens destinadas apenas aos nós que estão no mesmo agrupamento.

Após descrever os problemas gerados por essas características da versão original do protocolo *Multi-Geo*, as seções seguintes explicam as modificações realizadas na nova versão do protocolo e como elas podem resolver, ou pelo menos amenizar, cada um desses problemas. Ainda neste capítulo, na seção 3.5, é exposto o problema da localização dos nós em redes de sensores sem fio, além de algumas alternativas para sua resolução

sugeridas em outras pesquisas. A última seção deste capítulo apresenta o funcionamento completo do protocolo proposto neste trabalho.

3.2 Eleição Dinâmica do Nó Líder

No *Multi-Geo*, o nó líder de cada região virtual é escolhido logo no início da vida da rede, imediatamente após a definição da matriz de regiões virtuais, e é mantido até o final. Em cenários onde existe a possibilidade de falha dos nós, ou seja, existe uma probabilidade maior que 0 (zero) de um ou mais nós deixarem de fazer parte da rede devido ao término da carga de sua bateria ou devido a outro problema qualquer, pode ocorrer a falha de um nó líder e, nesse caso, nenhuma mensagem de dados gerada dentro desse agrupamento conseguirá sair de dentro do mesmo e não chegará, dessa forma, ao nó *sink*.

Sendo assim, quando um nó líder falha, é indispensável que um outro nó da mesma região virtual seja eleito o novo líder para substituir o antigo e permitir que os dados gerados nesse agrupamento continuem sendo entregues ao nó *sink*.

Para a eleição dos nós líderes no protocolo *Multi-Geo*, foi usada uma tática que consiste em calcular a distância de cada nó do agrupamento até o ponto de menor coordenada dessa região virtual (i.e.: o ponto do agrupamento com a coordenada relativa 0x0), que será chamado por este trabalho de origem do agrupamento e é destacado na figura 3.1(a) por um asterisco. O nó líder, destacado na mesma figura por um círculo mais escuro, é sempre o nó que está mais próximo desse ponto.

No *Multi-Q*, a tática usada pelo *Multi-Geo* foi ligeiramente alterada para permitir que cada nó líder possa se comunicar com os nós de sua região virtual com um menor gasto energético. Sendo assim, ao invés do nó líder eleito ser o nó mais próximo da origem do agrupamento, ele será o nó mais próximo do ponto de intersecção das duas diagonais da região virtual, que será chamado de centro do agrupamento. A figura 3.1(b) ilustra esse ponto, assim como o novo nó líder.

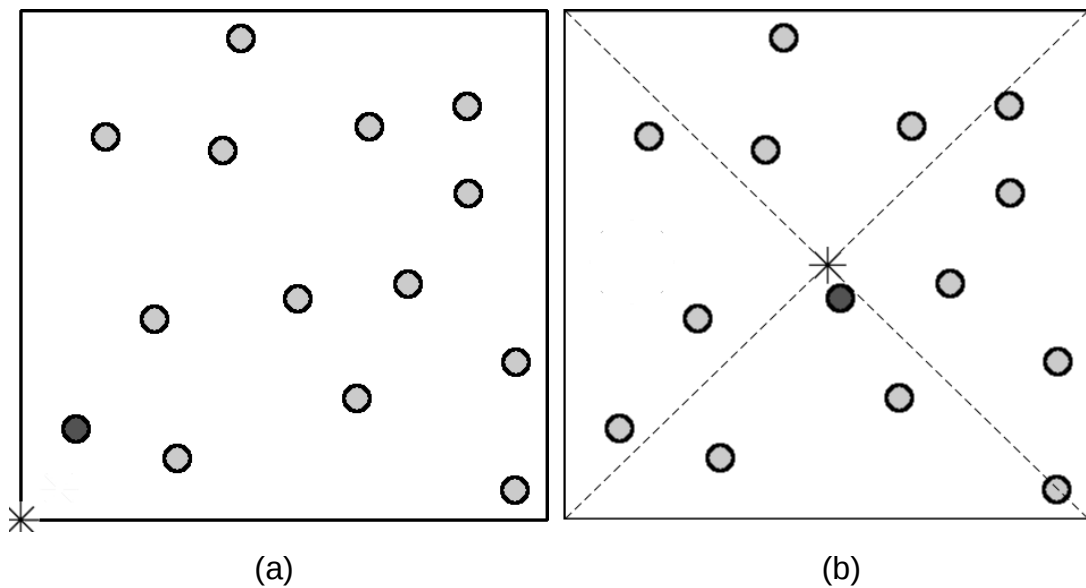


Figura 3.1. Seleção do nó líder no Multi-Geo e no Multi-Q.

Além disso, uma outra diferença entre o *Multi-Geo* e o *Multi-Q* é que a nova versão implementa uma eleição dinâmica do nó líder, o que significa que os nós de cada agrupamento comunicam entre si para decidir qual deles deve ser o nó líder. A comunicação é feita usando uma mensagem de controle contendo informações sobre a distância do nó que envia a mensagem até o centro do agrupamento. A primeira eleição de cada nó líder ocorre imediatamente após a definição de cada região virtual, como descrito a seguir:

- cada nó agenda o envio de uma mensagem de controle informando a sua distância até o centro do agrupamento em que está contido. O primeiro nó que enviar essa mensagem assume que é o nó líder. A distância do nó até o centro do agrupamento é calculada por cada nó logo após receber as informações sobre as coordenadas de seu respectivo agrupamento. O modo como as informações dos agrupamentos são geradas é discutido na seção 3.3;
- cada nó que recebe essa mensagem, compara sua distância até o centro do agrupamento com a distância do nó que enviou a mensagem. Essa informação está presente no próprio pacote recebido e, se a distância do nó que fez o envio é menor do que a distância do nó que recebeu, o nó

recedor cancela o envio de sua mensagem de controle e assume que o nó que enviou a mensagem é o nó líder. Caso contrário, ele não cancela o envio e espera até que receba uma mensagem de controle de outro nó ou até que o momento de realizar o seu envio chegue;

- quando esse momento chega, o nó assume que é o nó líder e faz a difusão da mensagem contendo sua distância até o centro da região virtual. O nó que fez a primeira difusão, e que também assumiu ser o nó líder, recebe essa nova mensagem de controle e, depois de comparar as distâncias até o centro do agrupamento, assume que o nó líder é o nó que enviou a mensagem. Os outros recebem a segunda mensagem enviada por difusão e atualizam suas referências para o nó líder.

Durante a vida da rede, quando um nó líder falha, é importante que o processo de eleição se repita para evitar que o agrupamento fique sem um nó líder. Esse processo pode começar em diferentes momentos dependendo da estrutura de roteamento sendo utilizada.

Quando a rede está funcionando no modo reativo, os nós de uma região virtual só perceberão que o nó líder falhou quando eles precisarem enviar algum dado ao mesmo. Quando isso ocorre, o nó que deseja fazer o envio assume o papel de nó líder e inicia o processo de eleição enviando uma mensagem de controle informando sua distância até o centro do agrupamento. Depois de receber a mensagem, cada nó assume o nó que enviou a mensagem como nó líder ou agenda a difusão de uma nova mensagem de controle, dependendo de sua distância até o centro do agrupamento.

Quando a rede está funcionando proativamente e o nó líder falha, os nós dessa região virtual deixarão de receber a mensagem de construção da árvore de roteamento e irão adotar o modo reativo de roteamento. Com isso, todos os nós do agrupamento agendarão a difusão da mensagem de controle, iniciando o processo de eleição do novo nó líder. Assim, um novo nó líder será escolhido e as mensagens geradas nessa região virtual continuarão sendo entregues ao nó *sink* normalmente.

Quando os nós agendam a difusão da mensagem de controle, o uso de um atraso aleatório para envio de cada mensagem pode fazer com que os nós mais próximos do centro do agrupamento tenham tempos de atraso altos, fazendo com que nós mais distantes e com tempos de atraso menores façam a difusão de suas mensagens desnecessariamente, já que estão distantes e, sendo assim, dificilmente serão nós líderes. Para evitar esse aumento no número de difusões durante a definição do nó líder, foi adotada uma estratégia para permitir que os nós mais próximos do centro do agrupamento realizem suas difusões antes que os nós mais distantes. Para isso, quando o envio da mensagem é agendado, o atraso no envio é calculado em função da distância de cada nó até o centro do agrupamento de acordo com a equação

$$\Delta d = \frac{d}{D/2} \cdot C = \frac{2d}{D} \cdot C, \quad (1)$$

onde d é a distância do nó até o centro do agrupamento, D é o tamanho do diâmetro do agrupamento, C é um valor constante e Δ é o atraso no envio. O centro do agrupamento e seu diâmetro são calculados por cada nó ao receber as informações de seu agrupamento no início da vida da rede e o valor da constante C é definido através dos parâmetros do protocolo, representando o maior atraso tolerado para o envio da mensagem de controle.

3.3 Agrupamentos com Tamanho Desigual

O *Multi-Geo* usa um número fixo de agrupamentos com as mesmas dimensões e organizados de modo a formar uma matriz. Em muitas aplicações do mundo real, o uso de nós distribuídos de forma não uniforme é comum causando a ocorrência de densidades de nós diferentes em diferentes regiões da rede. Nesses casos, o uso de agrupamentos com o mesmo tamanho pode não ser uma ideia muito boa, já que, em áreas onde a densidade de nós é alta, os agrupamentos serão compostos por uma grande quantidade de nós e apenas um nó líder, que se torna um gargalo para a saída de todas as mensagens geradas dentro de sua região virtual,

uma vez que ele será responsável por encaminhar as mensagens geradas por um grande número de nós, ficando sobrecarregado.

Por outro lado, se a densidade de nós em uma área é muito baixa, haverá a ocorrência de agrupamentos vizinhos com poucos nós em cada um deles, podendo inclusive existir agrupamentos com apenas um nó (i.e.: o nó líder). A ocorrência desses tipos de agrupamentos acabam indo contra o objetivo da formação das regiões virtuais, que é de diminuir o número de nós que realizam difusão de mensagens, uma vez que a porcentagem de nós líderes nessa área será alta. Assim, seria melhor se houvesse uma região virtual maior abrangendo mais nós e com menos nós líderes na área. Desse modo, o número de difusões realizadas seria reduzido.

Muitos protocolos baseados na formação de agrupamentos, tais como (CHIANG et al., 1997) e (YE; HEIDEMANN; ESTRIN, 2004), fazem com que cada nó líder abranja em sua região virtual todos os nós que estão ao seu alcance, dessa forma, esses protocolos também consideram somente o tamanho físico das regiões virtuais e não o número de nós contidos nas mesmas, ou seja, todos os agrupamentos têm aproximadamente o mesmo tamanho físico, que corresponde a um círculo onde o raio é igual ao alcance do nó líder. Esses protocolos conseguem atingir bons resultados quando usados em cenários com distribuição dos nós uniforme e com baixas densidades, no entanto, isso não acontece sempre nas aplicações reais. A figura 3.2 ilustra a formação de agrupamentos nesses tipos de protocolos em um cenário com distribuição dos nós não uniforme.

Note que os tamanhos físicos dos agrupamentos são iguais, fazendo com que haja sobrecarga do nó líder localizado na área onde a densidade de nós é maior. Através dessa ilustração, pode-se destacar também que, nesses tipos de protocolos, é comum o uso de nós roteadores, que geralmente, fazem parte de dois agrupamentos simultaneamente e são responsáveis por fazer a comunicação entre nós líderes já que, nesses casos, a distância não permite que estes se comuniquem diretamente (CHIANG et al., 1997). No protocolo *Multi-Geo*, uma vez que cada nó líder

pode se comunicar com os nós líderes vizinhos, o uso de nós exclusivamente roteadores não é necessário. Em outras palavras, além de ser a porta de entrada e saída do agrupamento, o próprio nó líder faz também o papel de roteador.

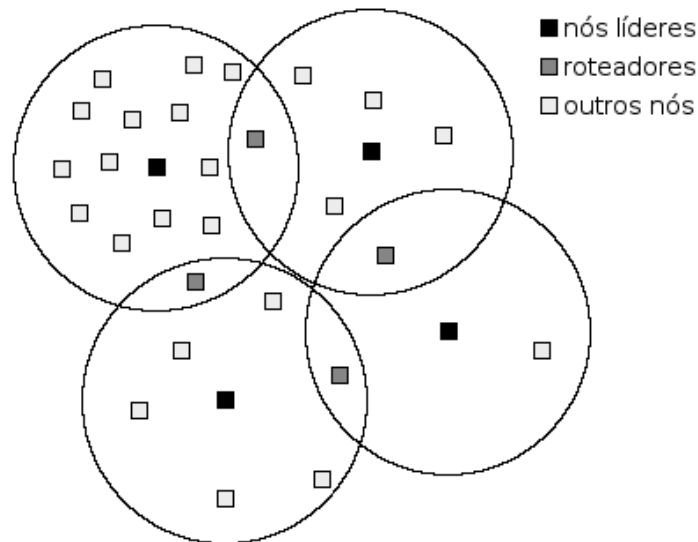


Figura 3.2. Divisão de agrupamentos baseada no alcance dos nós.

Para resolver o problema gerado pelo uso de agrupamentos de mesmo tamanho, neste trabalho se propõe que cada agrupamento tenha seu tamanho determinado pelo número de nós e não pela sua área. Assim, é possível a criação de regiões virtuais de diferentes dimensões, cada uma com um número de nós que deve ser aproximadamente igual a um valor especificado a priori.

Para tornar isso possível, esses agrupamentos são criados através do uso de uma estrutura *Quadtree* (SAMET, 1984), ou seja, para cada região virtual, se o número de nós é maior que o número máximo de nós permitidos por agrupamento, ela deve ser dividida em 4 novas regiões virtuais de tamanhos iguais. Esse processo é repetido sucessivamente para cada nova região virtual formada até que nenhuma delas contenha mais do que o número máximo de nós por agrupamento. Esse valor máximo é um novo parâmetro de simulação, criado para uso deste protocolo de roteamento. Deve-se ressaltar também que é necessário levar em consideração o raio

máximo de comunicação do nó sensor, uma vez que, uma região virtual muito grande pode impedir que o nó líder seja capaz de dar vazão aos pacotes gerados dentro do agrupamento. A figura 3.3 mostra uma comparação de duas divisões de agrupamentos em um mesmo cenário com uma distribuição não uniforme dos nós. A primeira com o método utilizado pelo *Multi-Geo* e a segunda com o uso da divisão através de uma *Quadtree*, proposta pelo *Multi-Q*.

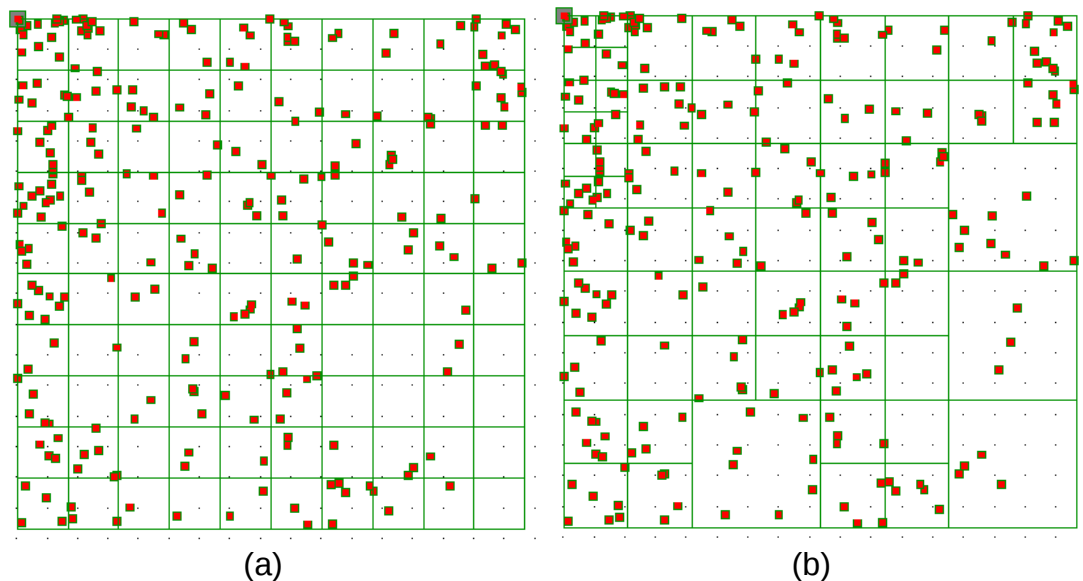


Figura 3.3. Divisão de agrupamentos em um mesmo cenário com utilização de agrupamentos de tamanho fixo e variável com o uso de *Quadtree*.

Através da análise da figura 3.3(b), pode-se observar que mesmo com a nova técnica de divisão dos agrupamentos, há a ocorrência de agrupamentos com poucos, ou até nenhum nó. Uma dessas ocorrências é destacada na figura 3.4(a). Isso acontece porque a estrutura *Quadtree* usada define apenas um número máximo de nós por agrupamento e não um número mínimo. Sendo assim, a ocorrência de um número elevado de nós em um agrupamento de forma não uniforme, pode acarretar na divisão do mesmo de forma que uma ou mais regiões virtuais resultantes contenham um número reduzido de nós.

Para reduzir esse problema, pode-se adequar o tamanho dos novos agrupamentos de acordo com o número de nós existentes em cada região

do agrupamento-pai ao invés de fazer a divisão em regiões com o mesmo tamanho. Essa técnica, ilustrada na figura 3.4(b), não foi implementada neste trabalho e é citada como sugestão para futuros trabalhos.

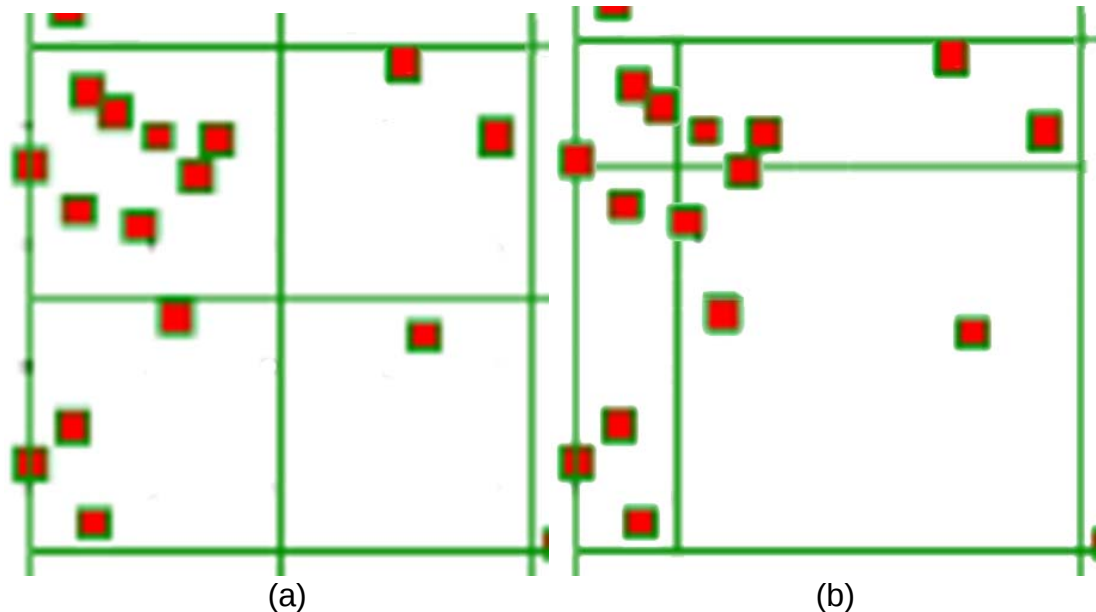


Figura 3.4. Divisão de um agrupamento com distribuição de nós não uniforme.

Todo o processo de definição dos agrupamentos é centralizado pelo nó *sink*, assim como é realizado no *Multi-Geo*, que, por sua vez, herdou essa característica do protocolo *LEACH-C* (LINDSEY; RAGHAVENDRA; SIVALINGAM, 2002). Assim, no início da vida da rede, os nós enviam as informações sobre sua localização, de modo que o nó *sink* possa definir cada agrupamento e enviar para cada nó as informações relativas à sua região virtual, como um identificador do agrupamento e os pares coordenados de cada vértice do quadrilátero que o delimita. Uma vez que os agrupamentos são definidos apenas no início da vida da rede, esses protocolos devem ser usados apenas para redes que não possuem mobilidade dos nós, já que as informações recebidas ficariam desatualizadas logo que o nó saísse da área de abrangência de seu agrupamento.

3.4 Alcance Limitado das Difusões

Uma vez que a potência empregada para realização da transmissão de um sinal é diretamente proporcional ao quadrado do raio de alcance do mesmo, ao enviar um pacote, o nó deve definir a potência que será utilizada de modo que o pacote possa alcançar o nó receptor, como descrito por (CORREIA et al., 2005), que apresenta dois métodos para ajuste de potência de transmissão para RSSFs. No caso da difusão de uma mensagem, não existe um único receptor, já que, o objetivo dessa mensagem é de atingir todos os nós que estão próximos ao nó que está fazendo o envio. Dessa forma, é comum que os nós realizem difusões de mensagens utilizando o alcance máximo de seus transmissores, empregando uma potência elevada e gerando maior consumo energético durante as difusões.

Essa prática é utilizada pelos protocolos *Multi* e *Multi-K*, que usam o alcance máximo do rádio dos nós para realizar difusão de mensagens com o objetivo de reduzir o número de saltos no caminho que liga o nó de origem dos dados até o nó *sink*. Embora essa prática aumente o consumo total de energia, uma vez que o gasto energético é proporcional ao alcance do sinal durante a transmissão de pacotes, isso pode ser uma boa forma de minimizar a latência da rede.

O protocolo *Multi-Geo* herda esse comportamento também com o objetivo de diminuir o número de saltos. No entanto, essa prática é repetida mesmo quando um nó está fazendo difusão de um pacote apenas para os nós do mesmo agrupamento. Note que nesse tipo de difusão, quanto menor for o tamanho do agrupamento, menor será a potência necessária para que o pacote alcance todos os nós dentro do mesmo.

Assim, esta modificação tem o objetivo de reduzir o consumo energético das difusões realizadas dentro de um agrupamento limitando o alcance do mesmo de modo que seja usada uma potência suficiente para alcançar apenas os nós que estão dentro desse agrupamento. Essa modificação pode resultar em uma outra forma de melhoria no consumo energético, já que a redução do alcance da transmissão reduz o ruído nos

outros agrupamentos, reduzindo colisões e retransmissões de pacotes. Além disso, a redução do ruído e das retransmissões de pacotes é também uma forma de melhoria na latência da rede, que pode amenizar um provável aumento na mesma causado pelo maior número de saltos.

Para que esta modificação no protocolo original seja possível, é preciso calcular a potência que deve ser empregada para a realização da transmissão e, sendo assim, foram definidas duas formas diferentes de se realizar esse cálculo, variando de acordo com tipo de difusão a ser realizado. Inicialmente, durante a definição dos nós líderes, os nós de um mesmo agrupamento ainda não se conhecem e, dessa forma, não sabem a potência necessária para atingir os nós da região. Assim, o cálculo da potência é baseado na maior distância possível entre dois nós de um mesmo agrupamento, ou seja, para se assegurar que todos os nós do agrupamento receberão a mensagem sendo enviada por difusão, o nó transmissor calcula o tamanho do diâmetro da região virtual, assim como feito na seção 3.2, e usa esse valor como o raio de alcance da transmissão.

A potência necessária para atingir a distância desejada é calculada de acordo com o *Wall Attenuation Factor* (WAF), que é um fator que determina o nível de atenuação do sinal à medida que sua distância até o rádio emissor aumenta (BULUSU; HEIDEMANN; ESTRIN, 2000). O WAF é baseado no modelo de propagação do sinal usado na simulação.

Deve-se destacar que durante a eleição do nó líder, todos os nós do agrupamento ficam conhecendo o nó eleito. Entretanto, o nó eleito só tomará conhecimento dos nós que também fizeram envio da mensagem de controle usada durante a eleição e, como apresentado na seção 3.2, geralmente esses nós serão os nós mais próximos do nó líder. Dessa forma, as próximas difusões realizadas pelo nó líder não terão potência suficiente para atingir todos os nós do agrupamento, mas apenas os nós mais próximos, isso se algum outro nó, além do nó líder, tiver feito o envio da mensagem de controle durante a eleição.

Entretanto, esse fato não gera um problema, já que, a partir desse momento, as difusões realizadas no interior da região virtual pelo nó líder serão com o objetivo de construção da árvore de roteamento quando o protocolo estiver atuando proativamente. Assim, os nós que não receberem o pacote de construção da árvore continuarão trabalhando reativamente e, quando houver a necessidade de enviar um pacote de dados, o nó fonte enviará diretamente ao nó líder, que passará a conhecê-lo e atualizará a potência necessária para atingir o nó mais distante, fazendo com que o nó fonte receba as próximas mensagens de construção da árvore.

Caso o nó líder falhe, a eleição de um novo líder será feita novamente usando a difusão de mensagens de controle com um alcance baseado no diâmetro do agrupamento. Esse diâmetro é conhecido desde o início uma vez que o tamanho das regiões virtuais não se alteram durante o tempo de vida da rede.

Sendo assim, esta modificação se resume em usar uma potência necessária para ter um raio de alcance igual ao diâmetro do agrupamento durante as eleições dos nós líderes e, nas difusões realizadas dentro de um agrupamento para construção da árvore de roteamento, usar uma potência suficiente para atingir o nó conhecido mais distante. Vale salientar ainda que essa estratégia de limitação no alcance das difusões só deve ser utilizada para difusões que objetivam alcançar unicamente os nós localizados na mesma região virtual do nó líder.

3.5 O Problema da Localização

Como mostrado anteriormente, o *Multi-Geo* é um protocolo que precisa determinar a localização de cada nó para realizar a divisão dos agrupamentos e a escolha dos nós líderes e, sendo assim, é indispensável o uso de técnicas para se obter essas informações geográficas.

Uma das possibilidades propostas pelo protocolo *Multi-Geo* é a de usar nós equipados com GPS (*Global Positioning System*) para permitir a descoberta da localização de cada nó, apesar do uso de GPS aumentar o

tamanho dos nós e deixá-los mais caros. O consumo energético do GPS também é um problema a ser levado em consideração.

Outras alternativas são apresentadas em (BULUSU; HEIDEMANN; ESTRIN, 2000) e (WANG; LEDERER; GAO, 2009). Nesses trabalhos, são propostos métodos de localização para redes de sensores sem fio sem o uso de GPS para nós pequenos localizados em áreas abertas. É ainda ressaltado em (WANG; LEDERER; GAO, 2009) que o problema da localização dos nós pode ser um problema NP-completo em alguns contextos, como por exemplo, em aplicações de prospecção de petróleo.

No entanto, como o problema de localização dos nós não é o foco deste trabalho, nos testes realizados, foi suposto que cada nó conhece sua localização e que não há mobilidade. Além disso, os avanços alcançados na produção de *chips* GPS podem fazer com que problemas referentes a custo, tamanho e consumo desses dispositivos se tornem cada vez mais irrelevantes, como pode ser observado no *chip Infineon XPOSYS* da Epson (RICKER, 2009).

3.6 Funcionamento do Protocolo Multi-Q

Agora que as modificações implementadas no protocolo *Multi-Geo* foram apresentadas, essa seção apresenta o funcionamento da versão final do protocolo de roteamento *Multi-Q*, proposto por este trabalho.

- Inicialmente, é realizada a divisão das regiões virtuais. Para isso, os nós informam sua localização ao nó *sink*, que faz a divisão dos agrupamentos baseada na densidade de nós em cada região da rede. Assim, o tamanho de cada agrupamento pode ser diferente dos demais com o objetivo de possibilitar que cada um deles tenha um número aproximadamente igual de nós, sempre considerando o raio máximo de comunicação dos nós. Essa divisão é realizada com o uso de uma estrutura *Quadtree* e, logo em seguida, o *sink* envia uma mensagem a cada nó contendo as informações geográficas de seu respectivo agrupamento.

- Após receber as informações de seus agrupamentos, os nós iniciam a eleição dos nós líderes para cada região virtual. Para isso, todos os nós do agrupamento agendam a difusão de uma mensagem de controle que informa sua distância até o centro da região virtual e assim, o nó eleito como nó líder será o nó mais próximo desse ponto. Para evitar o excesso de difusões, cada nó que recebe a mensagem de controle compara a sua distância à distância informada na mensagem de controle e, se sua distância for maior, ele cancela a sua difusão. Durante o tempo de vida da rede, esse processo de eleição do nó líder se repete em cada agrupamento onde o nó líder falha.
- A princípio, todos os nós adotam uma técnica reativa de roteamento, ou seja, cada nó fonte inicia a construção de um caminho para a entrega dos dados. Para construir esse caminho, o nó fonte deve enviar um pacote de controle destinado ao nó *sink* informando a existência de um dado para ser enviado e, para atingir esse objetivo, o nó fonte envia esse pacote diretamente para o nó líder de sua região virtual, que é responsável por fazer uma difusão para os outros agrupamentos para que a mensagem possa viajar até o *sink*. Em seguida, o nó *sink* envia uma mensagem de requisição de dados que volta ao nó fonte através de transmissões *unicast* pelo caminho inverso percorrido pela primeira mensagem, como apresentado anteriormente na seção 2.4.1. Assim, o caminho de disseminação dos dados é criado e o nó fonte consegue enviar pacotes de dados para o nó *sink* através dos nós líderes que fazem parte desse caminho.
- Quando o nó *sink* percebe que o número de nós gerando dados é alto, ele dá início a uma técnica proativa de roteamento que consiste na construção de uma árvore com raiz no próprio *sink*. Sendo assim, esse nó faz a difusão de um pacote de construção da árvore para os nós de seu agrupamento e para os nós líderes vizinhos. Cada nó líder repete esse procedimento e realiza a difusão da mensagem para os nós de seu agrupamento e para os nós líderes vizinhos, até que todos os nós tenham recebido a mensagem. Desse modo, os nós podem enviar seus dados

através da estrutura gerada, de modo que, a cada salto das mensagens de dados, um contador K , contido no cabeçalho da mensagem, é incrementado em uma unidade, e assim, o nó *sink* pode calcular a distância, em saltos, do nó fonte mais distante. Isso é feito para que nas próximas construções da árvore de roteamento, o *sink* insira esse valor K na mensagem enviada, então, cada nó líder decrementa esse valor antes de realizar a difusão da mensagem. Quando o contador K atinge o valor 0 (zero), a mensagem é descartada.

- A construção da árvore é repetida periodicamente enquanto o nó *sink* considerar que o número de nós fonte é alto. Quando esse número voltar a ser baixo, a árvore não é mais reconstruída e os nós da rede voltam a atuar reativamente.

Capítulo 4 - Avaliação do Multi-Q

Este capítulo apresenta uma avaliação de desempenho do protocolo *Multi-Q*, apresentado neste trabalho. Nas seções a seguir, serão apresentadas a metodologia utilizada para avaliação e os resultados obtidos.

4.1 Metodologia

4.1.1 Introdução

A avaliação do protocolo *Multi-Q* consiste na execução de diversas simulações, utilizando as diferentes versões desse protocolo, objetivando a comparação dos resultados obtidos por cada uma dessas versões. Dessa forma, esta seção apresenta os diversos tipos de cenários utilizados nas simulações, bem como os parâmetros empregados.

As simulações apresentadas em (SILVA; GOULART, 2009), realizadas com o uso do protocolo *Multi-Geo*, foram feitas em cenários cujo número de nós variam de 50 a 300 nós e os agrupamentos são organizados como uma matriz quadrada contendo 100 agrupamentos, assim, o número médio de nós em cada agrupamento é igual a 3 no melhor caso, ou seja, quando o número de nós é 300.

Embora essa média represente um valor muito baixo, não era praticável a execução de simulações com um maior número de nós porque o simulador utilizado (i.e.: NS-2) tem um problema conhecido de escalabilidade fazendo com que as simulações se tornem extremamente lentas, tendo seu tempo aumentado exponencialmente à medida que o número de nós é aumentado (XUE et al., 2007). Para contornar o problema

da escalabilidade, um novo simulador foi desenvolvido para este trabalho. A seguir, esse simulador será descrito com mais detalhes e, posteriormente, alguns importantes parâmetros de simulação serão discutidos.

4.1.2 O Simulador

O simulador utilizado, apresentado em (SOUZA; ROCHA; FERREIRA, 2009), não é um simulador tão genérico quanto o NS-2, que foi desenvolvido para suportar simulações de vários tipos diferentes de redes. Ele é uma extensão do simulador Hades (HENDRICH, 2000) que foi desenvolvida especificamente para simulação de redes onde a comunicação é feita sem fio, em um esquema *ad-hoc*, como é o caso das RSSFs. Além disso, ele é um simulador simples, desenvolvido para ser usado em simulações simples, ou seja, com uma taxa de ocorrência de eventos não muito elevada, uso de ambientes bidimensionais, abertos e sem obstáculos e ainda foi otimizado para simulações de ambientes onde não há mobilidade dos nós.

Uma outra simplificação nesse simulador é a minimização da quantidade de informações geradas pelo mesmo. Uma análise do arquivo *trace*, gerado pelas simulações executadas no NS-2, demonstra um grande detalhamento nos dados de saída do simulador, sendo que estes nem sempre são necessários para a análise dos resultados.

Além disso, uma integração entre a camada MAC e a camada física dos nós possibilitou um menor processamento com troca de mensagens entre essas duas camadas. Assim, com o uso dessas e outras simplificações, o simulador pôde conseguir maior escalabilidade, permitindo simulações mais rápidas, com um maior número de nós e obtendo resultados com uma precisão muito próxima aos resultados obtidos pelas simulações realizadas no NS-2.

Como o simulador Hades é originalmente um simulador de circuitos digitais, ou seja, foi desenvolvido para ambientes que utilizam comunicação através de fios, para possibilitar a comunicação sem fio dos sensores, um componente *Environment* foi criado para representar o cenário onde os nós estão situados. Assim, ao invés dos componentes *Node*, que representam

os nós, se conectarem diretamente uns aos outros, eles possuem uma ligação com o componente *Environment* e, dessa forma, quando um componente *Node* envia um pacote, esse pacote é recebido pelo *Environment*, que, baseado na distância entre os nós e a potência empregada no sinal enviado, calcula quais nós serão capazes de receber o sinal, e assim, o próprio *Environment* faz o envio do pacote para esses componentes *Node*, informando também a potência do sinal recebido.

O tratamento das colisões é feito pelo próprio componente *Node* à medida que o mesmo recebe os pacotes informando a existência de uma transmissão sendo realizada por um vizinho. A implementação do componente *Node* foi feita através da implementação dos módulos que o compõe, como o módulo de consumo energético e os protocolos de roteamento e de controle de acesso ao meio.

Existem também diferentes classes para representar diferentes tipos de pacotes. Um desses é o pacote de dados que é gerado pelos protocolos implementados no nó e enviado por um nó até outro através do componente *Environment*. Outro tipo de pacote é o de controle usado para troca de informações entre nó e ambiente. Estes são usados, por exemplo, para o nó receber informações sobre eventos detectados ou informações sobre a disponibilidade do meio quando solicitado pelo protocolo da camada MAC (SOUZA; ROCHA; FERREIRA, 2009).

A figura 4.1 ilustra os componentes implementados na extensão desenvolvida e sua interação com os componentes nativos do simulador *Hades*. Cada objeto *Node* possui uma instância do componente *Port* para se comunicar com objeto *Environment*, enquanto que o número de objetos *Port* existentes no componente *Environment* é determinado dinamicamente de acordo com o número de nós utilizados na rede, já que o objeto *Environment* utiliza uma porta distinta para se conectar a cada objeto *Node* existente.

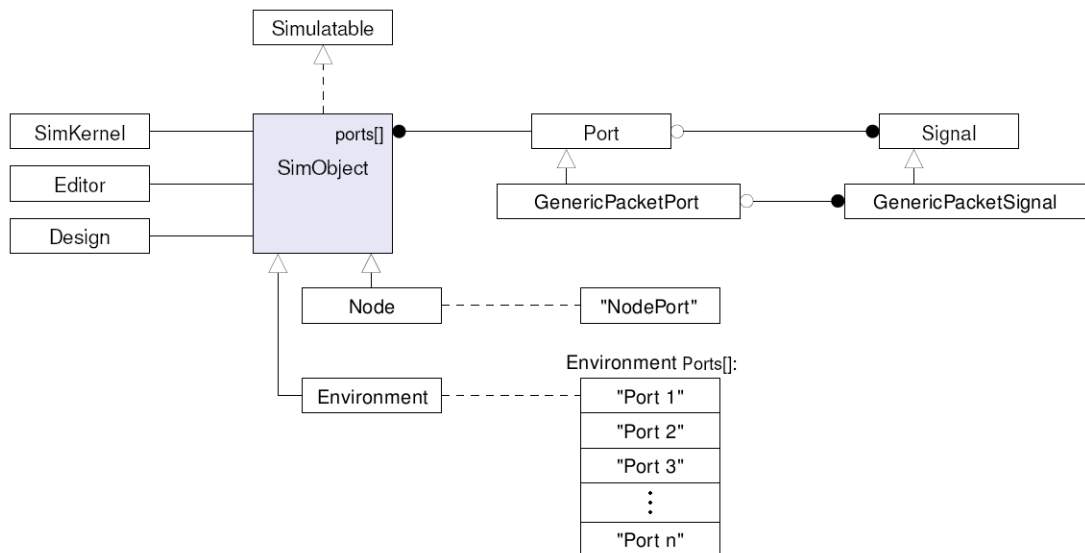


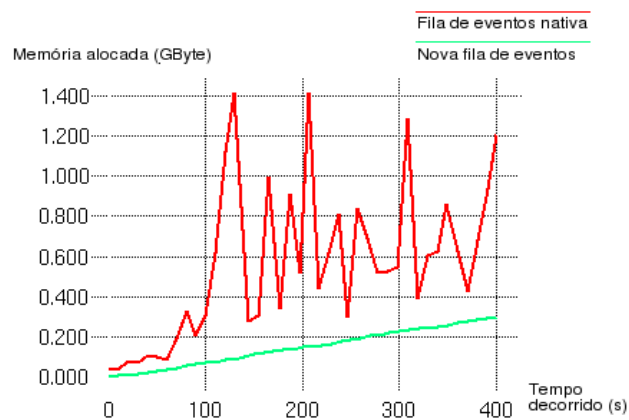
Figure 4.1. Visão esquemática dos componentes implementados.

Para permitir a simulação, os eventos (e.g.: envio e recebimento de pacotes, escuta do canal, etc.) devem ser inseridos em uma lista de eventos disponível no *Hades*. Esses eventos são ordenados pelo tempo de início de sua execução e, em cada iteração, o simulador remove o primeiro evento da fila e o executa. Após a simulação de cada evento, o tempo de simulação é atualizado com o valor do tempo de ocorrência do evento. Isso significa que o tempo de simulação não é contínuo, podendo haver saltos no tempo evitando, assim, períodos de inatividade do simulador.

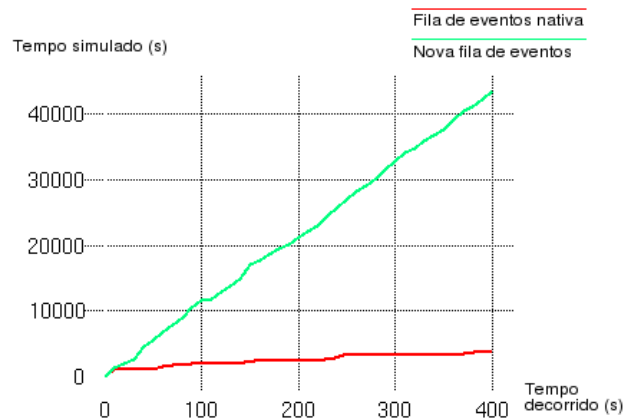
Para adicionar eventos na fila de eventos do *Hades*, a classe desses eventos deve estender a superclasse *SimEvent*, disponibilizada na biblioteca do *Hades*. No entanto, objetos dessa classe ocupam uma memória relativamente alta e os testes com cenários grandes (e.g.: 1000 nós) resultaram em falta de memória ao usar uma máquina virtual Java com 1,5Gb de memória. Para resolver esse problema, a fila de eventos do *Hades* foi substituída por uma nova estrutura e, assim, foi possível utilizar novas classes de eventos sem a necessidade de estender a classe *SimEvent*. Essas novas classes de eventos são mais simples e seus objetos precisam implementar apenas o método *execute()*.

Usando essa classe de eventos mais simples, foi possível reduzir a memória necessária para executar simulações. Assim, há um aumento da

memória disponível e menos chamadas ao coletor de lixo do Java, resultando em simulações mais rápidas e sem erros de execução por falta de memória. A figura 4.2 mostra uma comparação entre a fila de eventos nativa do *Hades* e a fila implementada. Na figura 4.2(a), é comparada a memória utilizada em cada abordagem, sendo que a grande oscilação apresentada na primeira alternativa é resultado das inúmeras chamadas ao coletor de lixo do Java. Na figura 4.2(b), é comparado o tempo simulado pelas duas abordagens em função do intervalo de tempo decorrido desde o início da simulação.



(a)



(b)

Figure 4.2. Comparação entre as duas implementações da fila de eventos do simulador.

Através dessas figuras pode-se perceber que a modificação realizada na fila de eventos do *Hades* é capaz de permitir que uma simulação seja

realizada com um menor custo computacional. Além disso, pode-se realizar simulações mais longas em um menor tempo.

4.1.3 Parâmetros da Simulação

Os parâmetros do cenário foram baseados nos parâmetros utilizados no desenvolvimento e teste das versões anteriores deste protocolo (i.e.: *Multi*, *Multi-K* e *Multi-Geo*). Assim, o cenário tem o tamanho igual a 100x100 metros, o nó *sink* está posicionado na coordenada 0x0 e nem o *sink*, nem os outros nós, possuem mobilidade.

Nas pesquisas anteriores a esta, o número total de nós na rede variavam de 50 até 300 devido ao problema de escalabilidade do NS-2, apresentado anteriormente. Entretanto, como a principal característica do *Multi-Geo* e do *Multi-Q* é a de agrupamento dos nós, o número total de nós na rede foi aumentado para permitir a existência de um maior número de nós em cada agrupamento, permitindo uma melhor percepção dos benefícios na utilização de agrupamentos. Dessa forma, foram criados cenários diferentes, contendo 250, 500, 750 e 1000 nós, para as simulações realizadas neste trabalho.

Os tipos de distribuição dos nós pelo cenário foram mantidos semelhantes aos usados pelos trabalhos anteriores. Assim, foram usados dois tipos de distribuição: a distribuição uniforme, onde o número de nós em cada parte do cenário é aproximadamente igual, e a distribuição exponencial, onde a maior concentração de nós ocorre nas proximidades do nó *sink*. A distribuição exponencial é frequentemente utilizada no mundo real quando os nós sensores, incluindo o nó *sink*, são posicionados onde a concentração de eventos é maior, permitindo que a entrega de mensagens de dados ao *sink* seja feita com menor número de saltos. A figura 4.3 ilustra esses dois tipos de distribuição de nós em um cenário contendo 500 nós.

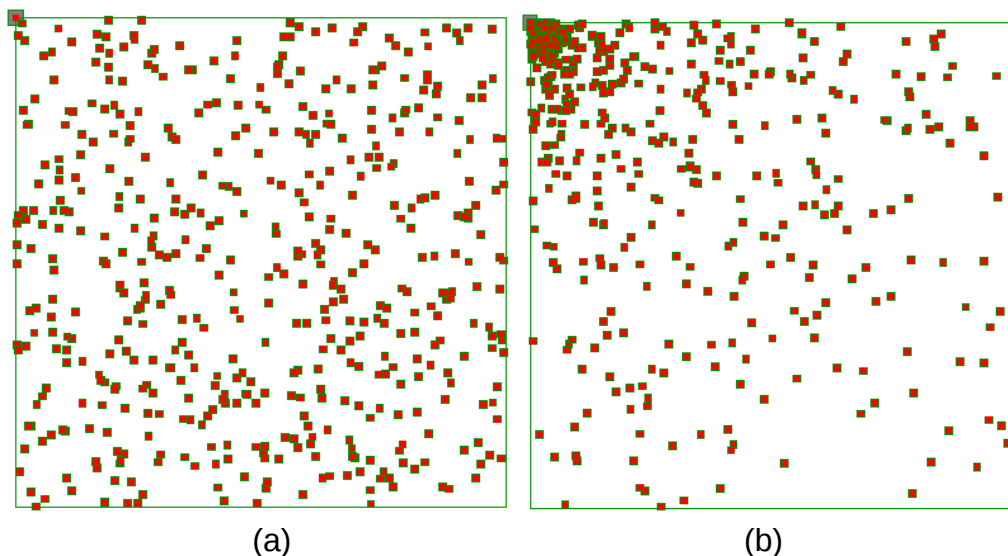


Figura 4.3. Tipos de distribuição dos nós: (a) Distribuição uniforme;
(b) Distribuição exponencial;

Também baseado nas pesquisas anteriores, o tempo de cada simulação foi definido para 4000 segundos, sendo que, as detecções de eventos e os consequentes envios de pacotes de dados começam a ser realizados a partir dos 1000 segundos de simulação. Além disso, o número de nós fonte de dados também foi fixado em 20. Esses 20 nós são escolhidos aleatoriamente e o início do envio dos dados gerados por cada nó também acontece em instantes definidos de forma aleatória. O intervalo entre o envio dos pacotes gerados por cada nó fonte foi igual a 10 segundos. Esses parâmetros, entre outros, estão listados na tabela 4.1.

Na camada MAC, os trabalhos anteriores, usaram o protocolo IEEE 802.11 para a realização da comunicação sem fio. No entanto, esse protocolo é mais apropriado para dispositivos maiores, com capacidade energética relativamente alta e com possibilidade de recarga das baterias. Esse não é o caso das RSSF convencionais, onde frequentemente são usados nós com baixas capacidades energéticas e sem a possibilidade de recarga de suas baterias, fazendo com que o uso do protocolo IEEE 802.11 deixe de ser apropriado para esse tipo de rede. Desse modo, nos testes realizados neste trabalho foi implementado e utilizado o protocolo S-MAC (YE; HEIDEMANN; ESTRIN, 2004), que foi desenvolvido

especificamente para ao uso em RSSF e inclui técnicas para reduzir o consumo de várias fontes de desperdício de energia. No entanto, esse protocolo foi implementado sem ciclos periódicos de inatividade dos nós que são agendados e executados conjuntamente por todos os nós de um agrupamento. O único período de inatividade utilizado nessa implementação é o que ocorre quando um nó recebe um pacote referente a uma mensagem que não é destinada a ele.

Tabela 4.1. Alguns parâmetros da simulação

Parâmetro	Valor
Número de nós geradores de dados	20
Intervalo entre envio de dados	10 segundos
Tamanho dos pacotes de controle	16 bytes
Tamanho dos pacotes de dados	20 bytes
Início da geração de dados	1000 segundos
Tempo de simulação	4000 segundos
Potência de envio	33 mW
Potência de recepção	30 mW
Largura de banda	76800 bps
Alcance do rádio	40 metros

Como descrito anteriormente, o *Multi-Q* usa um algoritmo baseado na estrutura *Quadtree* para definir o tamanho de cada agrupamento. Assim, os agrupamentos são redivididos até que o número de nós em cada agrupamento seja menor ou igual ao valor máximo de nós permitido. Esse valor é definido por um parâmetro do protocolo, sendo que, várias simulações foram executadas com diferentes valores para esse parâmetro objetivando determinar o valor que obtém os melhores resultados.

Além disso, outras simulações foram realizadas para determinar se a mudança no posicionamento do nó líder, como descrito em capítulos anteriores, proporcionou melhores resultados. Foram realizadas diversas simulações utilizando o nó líder posicionado nas proximidades da origem do agrupamento, como sugerido pelo protocolo *Multi-Geo*, e os resultados obtidos foram comparados com resultados oriundos de outras simulações,

utilizando o nó líder posicionado nas proximidades do centro do agrupamento.

Um outro parâmetro variável das simulações foi a probabilidade de falha de cada nó durante a simulação. A falha de um nó é permanente, assim, quando um nó falha, ele é excluído da rede, deixando de receber e enviar pacotes. Nas simulações, esse parâmetro variou de 0% a 50% para permitir a comparação da tolerância a falhas entre os protocolos de roteamento implementados. As falhas são geradas aleatoriamente, de modo que, no início da simulação é determinado se e quando cada nó irá falhar baseado na probabilidade de falha. Como a seleção desses nós é aleatória, qualquer nó pode falhar durante a simulação, inclusive os nós líderes.

Para verificar as melhorias obtidas pelo protocolo *Multi-Q*, os protocolos *Multi*, *Multi-K*, duas versões do protocolo *Multi-Geo* e o protocolo *Multi-Q* foram implementados e comparados. As duas versões implementadas do *Multi-Geo* foram a versão original do protocolo e uma versão parcialmente modificada, que implementa apenas duas modificações propostas por este trabalho: o alcance limitado das difusões e a eleição dinâmica dos nós líderes. O *Multi-Q*, que é a versão final proposta por este trabalho, implementa também o uso de agrupamentos de tamanhos diferentes.

4.2 Resultados

4.2.1 Definição dos Parâmetros do Protocolo

Esta seção apresenta os resultados obtidos da simulação dos cenários descritos na seção anterior. Os resultados, também apresentados em (SOUZA et al., 2009), foram obtidos através da média dos resultados de 30 simulações, incluindo cenários com distribuições uniformes e exponenciais dos nós, uma vez que o uso de *Quadtrees* minimiza as diferenças entre esses diferentes tipos de distribuições.

Inicialmente, as simulações tiveram o objetivo de determinar o melhor valor para o novo parâmetro inserido no *Multi-Q* que determina o número

máximo de nós dentro de um mesmo agrupamento, além de verificar se a mudança no posicionamento do nó líder da origem para o centro do agrupamento proporciona melhores resultados.

A figura 4.4 e a figura 4.5 exibem gráficos comparando os resultados das simulações do *Multi-Q* quando o número máximo de nós por agrupamento é variado. Esses gráficos mostram a variação do consumo médio por pacote entregue usando redes de sensores com 500 e 1000 nós respectivamente.

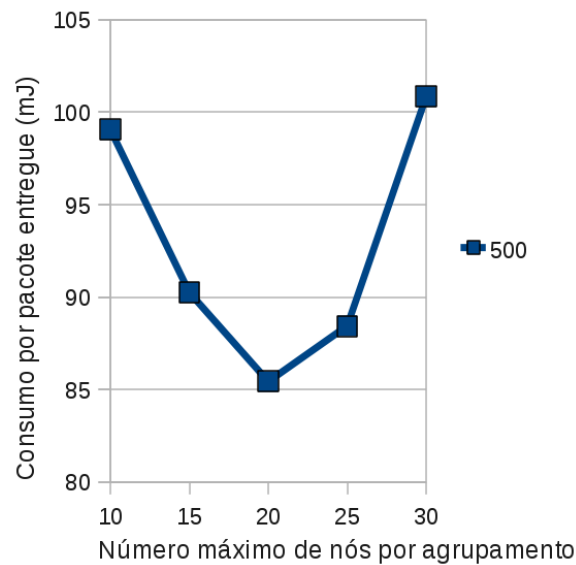


Figura 4.4. Variação do consumo de acordo com o número máximo de nós por agrupamento utilizando redes com 500 nós.

Baseado nos resultados apresentados por essas figuras, o valor adotado para o parâmetro que define o número máximo de nós por agrupamento foi igual a 20.

A figura 4.6 e a figura 4.7 exibem gráficos comparando os resultados obtidos por simulações onde foram usadas duas estratégias diferentes para a eleição dos nós líderes. A linha tracejada indica os resultados obtidos quando foram eleitos os nós líderes posicionados mais próximos à origem do agrupamento e a linha contínua indica os resultados obtidos quando os nós líderes eram os mais próximos ao centro do agrupamento. Ambos os gráficos exibem resultados obtidos em função da quantidade de nós na rede,

variando esse valor de 250 até 1000 nós, no entanto, na figura 4.6 é utilizado o consumo energético total da rede enquanto que na figura 4.7 é utilizada a taxa de pacotes entregues como métrica para comparação das duas estratégias.

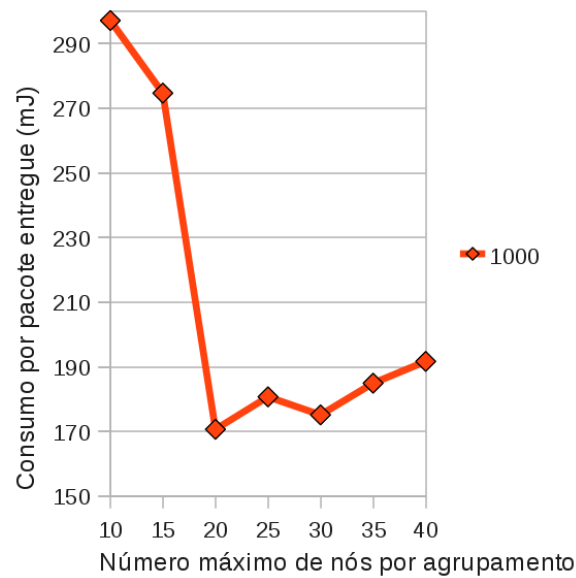


Figura 4.5. Variação do consumo de acordo com o número máximo de nós por agrupamento utilizando redes com 1000 nós.

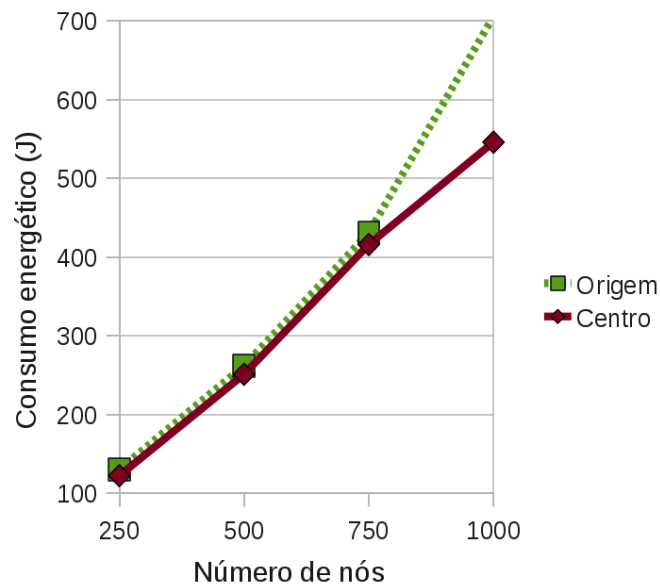


Figura 4.6. Variação do consumo energético total em função do número de nós na rede utilizando diferentes posicionamentos para os nós líderes.

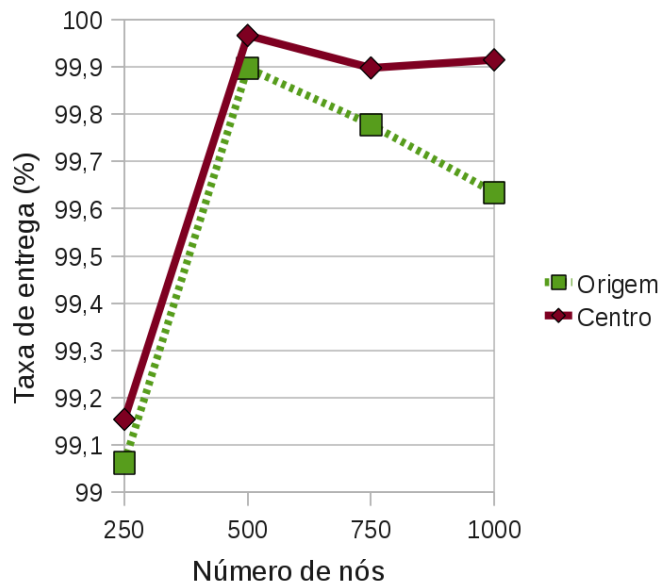


Figura 4.7. Variação da taxa de entrega de pacotes em função do número total de nós na rede utilizando diferentes posicionamentos para os líderes .

Dessa forma, os gráficos apresentados mostram que, como era esperado, o uso de nós líderes posicionados na região central do agrupamento obtém melhores resultados. Essa melhoria nos resultados se deve ao fato de que a distância média dos nós líderes aos demais nós do agrupamento é menor quando essa estratégia é utilizada, fazendo com que o gasto energético oriundo de transmissões realizadas no interior do agrupamento seja menor. Além disso, o uso dessa estratégia juntamente com a limitação do alcance das difusões diminui o ruído nos agrupamentos vizinhos, possibilitando um menor número de colisões e uma maior taxa de entrega dos pacotes.

Nas seções seguintes, são feitas comparações dos resultados obtidos pelos protocolos *Multi*, *Multi-K*, *Multi-Geo*, uma versão intermediária entre o *Multi-Geo* e o *Multi-Q* e a versão final do protocolo *Multi-Q*. Essa versão parcialmente modificada implementa o alcance limitado das difusões e a eleição dinâmica dos nós líderes, enquanto que a versão final do *Multi-Q* implementa o alcance limitado das difusões, a eleição dinâmica dos nós líderes e a variação do tamanho dos agrupamentos de acordo com o número de nós contidos em cada região do cenário. Nas figuras apresentadas, os

resultados obtidos pela versão parcialmente modificada são apresentados pela linha chamada *Multi-Geo 2*.

4.2.2 Avaliação do Consumo Energético

Esta seção apresenta uma comparação entre o consumo energético dos protocolos apresentados. A figura 4.8 mostra, para cada versão, a evolução do consumo energético quando o número de nós da rede é aumentado. Nestas simulações a probabilidade de falha dos nós é 0%.

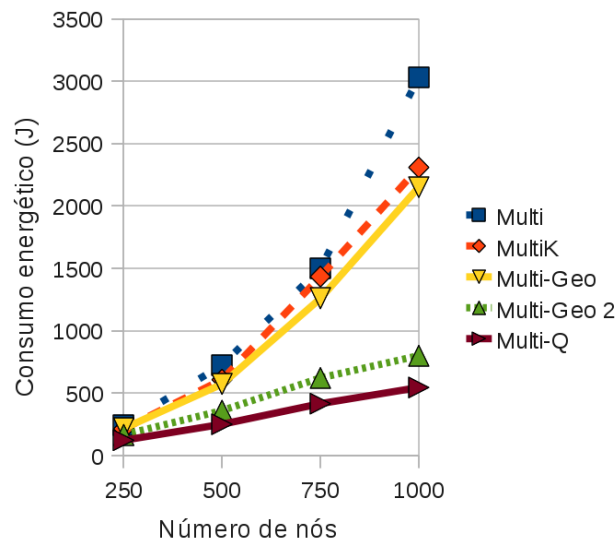


Figura 4.8. Variação do consumo total de energia de acordo com o número total de nós na rede.

Uma outra avaliação do consumo energético é apresentada na figura 4.9. Esta comparação varia a probabilidade de falha dos nós enquanto mantém o número total de nós fixo em 1000 nós.

A princípio, no *Multi-Geo*, o aumento da probabilidade de falha fazia com que o consumo diminuísse consideravelmente, no entanto, essa diminuição acontecia unicamente porque quando um nó líder falhava, as mensagens daquele agrupamento deixavam de ser encaminhadas, diminuindo o número de transmissões e, conseqüentemente, o consumo total da rede. Por isso, para possibilitar uma avaliação mais confiável quando a probabilidade de falhas é maior do que zero, a eleição dinâmica do nó líder foi implementada inclusive para a versão original do *Multi-Geo*, assim,

quando um nó líder falha, os pacotes originados no respectivo agrupamento não são perdidos. Um novo nó líder é eleito e os pacotes continuam sendo entregues.

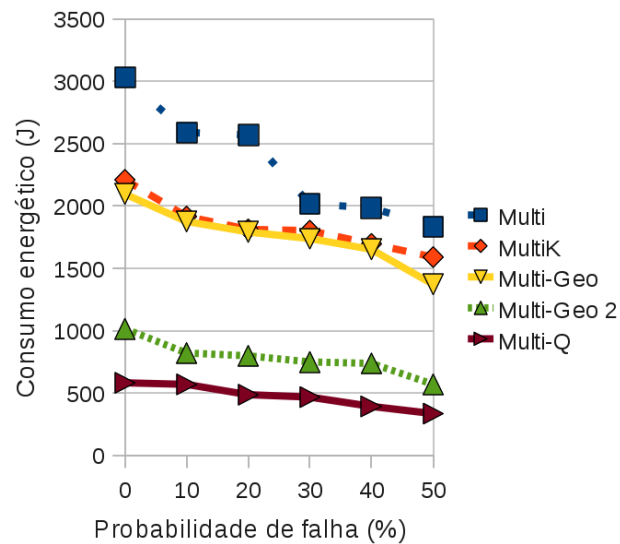


Figura 4.9. Variação do consumo energético total em função da probabilidade de falha dos nós.

Uma outra comparação mais detalhada entre o consumo energético dos protocolos *Multi-Geo*, *Multi-Geo 2* e *Multi-Q* é ilustrada na figura 4.10. Os valores obtidos se encontram dentro de um intervalo de confiança igual a 95%.

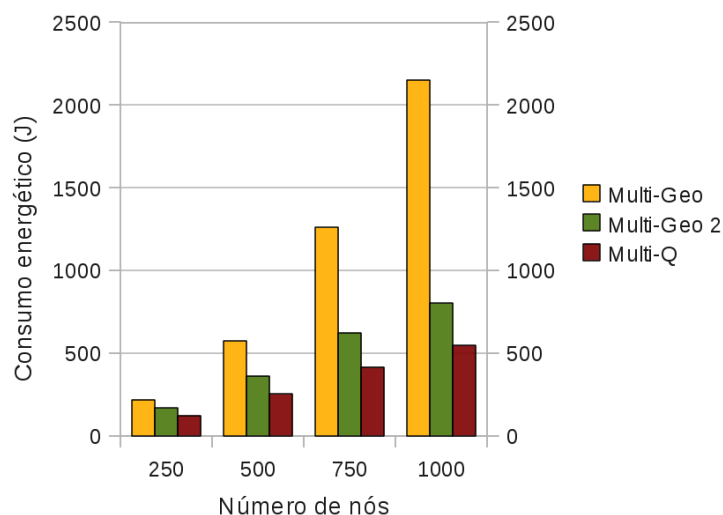


Figura 4.10. Comparação detalhada entre o consumo energético dos protocolos *Multi-Geo*, *Multi-Geo 2* e *Multi-Q*.

Uma análise da figura 4.8, da figura 4.9 e da figura 4.10 demonstra que a limitação do alcance das difusões possibilita um menor gasto energético. Além disso, o uso de agrupamentos menores nas regiões onde a densidade é maior também provou que pode diminuir o consumo energético, especialmente para redes maiores e mais densas, onde o tamanho médio dos agrupamentos é menor. Assim, o transporte de dados do nó fonte até o nó *sink* é feito com um maior número de pequenos saltos, usando menores potências de transmissão.

4.2.3 Avaliação da Taxa de Entrega

A taxa de entrega de pacotes é uma importante métrica de avaliação já que, através dela, pode-se determinar o grau de confiança da avaliação do consumo energético. Isso acontece porque é preciso determinar se a diminuição no consumo energético é consequência da maior eficiência dos protocolos sendo utilizados ou se é apenas uma consequência da diminuição do tráfego de pacotes de dados na rede, minimizando o gasto energético com transmissões. Sendo assim, se a taxa de entrega diminuir consideravelmente, a economia no consumo pode ser apenas uma consequência da baixa taxa de entrega.

A figura 4.11 mostra que além das taxas de entrega média das duas novas versões (i.e.: *Multi-Geo 2* e *Multi-Q*) não diminuírem em relação ao protocolo *Multi-Geo*, elas são ainda maiores que as taxas de entrega das versões anteriores do protocolo quando o tamanho da rede é aumentado consideravelmente já que essas versões foram desenvolvidas para uso em redes com maior número de nós.

Na figura 4.12, o gráfico apresentado mostra que a taxa de entrega de pacotes continuou mais alta do que a versão original do protocolo *Multi-Geo* quando a probabilidade de falha dos nós foi variada em uma rede com 1000 nós. No entanto, em alguns casos, a taxa de entrega do *Multi-Q* não foi maior do que as taxas de entrega dos protocolos *Multi* e *Multi-K*, conseguindo apenas resultados aproximadamente iguais. Isso foi observado em redes com menor número de nós e se deve ao fato de que o protocolo

Multi-Q foi desenvolvido para trabalhar principalmente com redes contendo muitos nós.

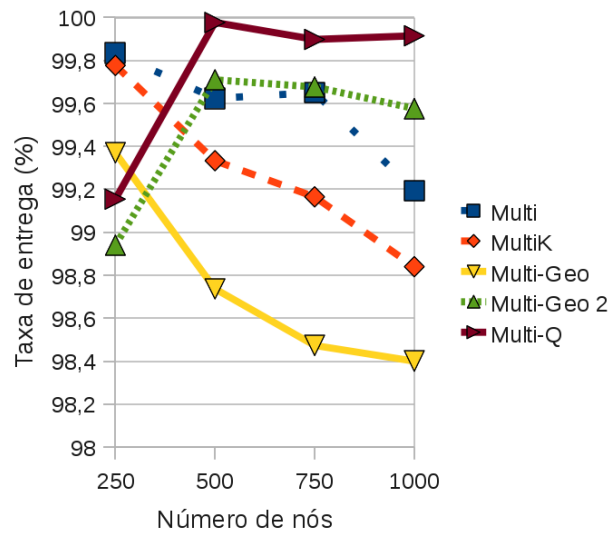


Figura 4.11. Variação da taxa de entrega de pacotes em função do número total de nós na rede.

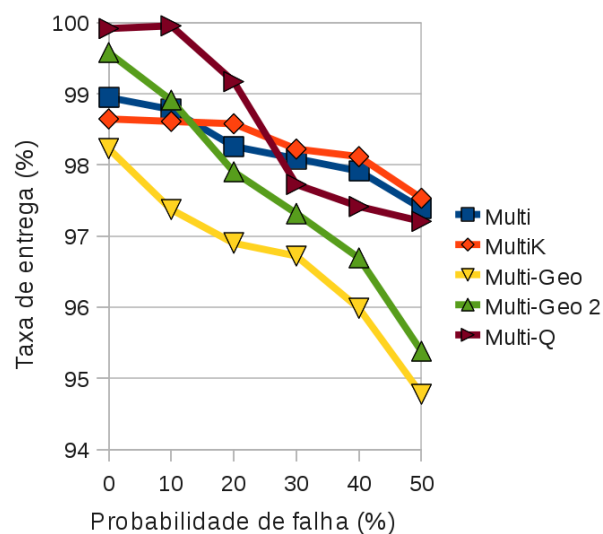


Figura 4.12. Variação da taxa de entrega de pacotes em função da probabilidade de falha de cada nó.

Assim, as figuras apresentadas mostraram que o *Multi-Q* alcançou uma taxa de entrega de pacotes satisfatória e demonstraram ainda que a melhoria no consumo energético não acontece devido a uma diminuição no número de pacotes de dados trafegando pela rede.

4.2.4 Avaliação da Latência

Uma das modificações feitas no protocolo *Multi-Geo* foi a criação de agrupamentos com tamanhos diferentes de acordo com o número de nós em cada área da rede. O aumento do número de agrupamentos em uma área tende a aumentar o número de saltos necessários para que um pacote passe por esses agrupamentos. Embora o uso de múltiplos pequenos saltos geralmente consuma menos energia que o uso de menos saltos maiores, essa prática pode ter um impacto negativo na latência da rede.

Nesta seção, uma comparação da latência é apresentada pelos gráficos na figura 4.13 e figura 4.14. O primeiro gráfico compara a latência média dos protocolos variando o número de nós na rede. O segundo gráfico varia a probabilidade de falha em uma rede de 1000 nós para comparar a latência média da rede.

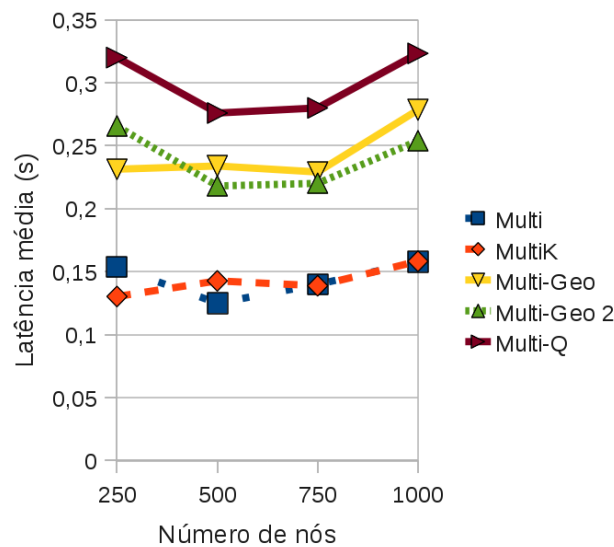


Figura 4.13. Variação da latência média dos protocolos de acordo com o número total de nós na rede.

As figuras apresentadas mostram que, embora o novo protocolo tenha alcançado, em alguns casos, quase o dobro da latência do protocolo *Multi* e uma latência média maior do que a do protocolo *Multi-Geo*, isso pode ser aceitável, já que, o aumento da latência média da rede com o objetivo de reduzir o consumo energético total é um *tradeoff* comum em muitas aplicações reais. Em aplicações de vigilância e monitoramento, por exemplo,

pode haver uma tolerância a uma latência adicional na entrega de mensagens porque a velocidade da rede é ordens de magnitude maior que a velocidade do objeto físico sendo detectado. Latências de ordem menores do que um segundo não são tão importantes e podem compensar uma economia energética.

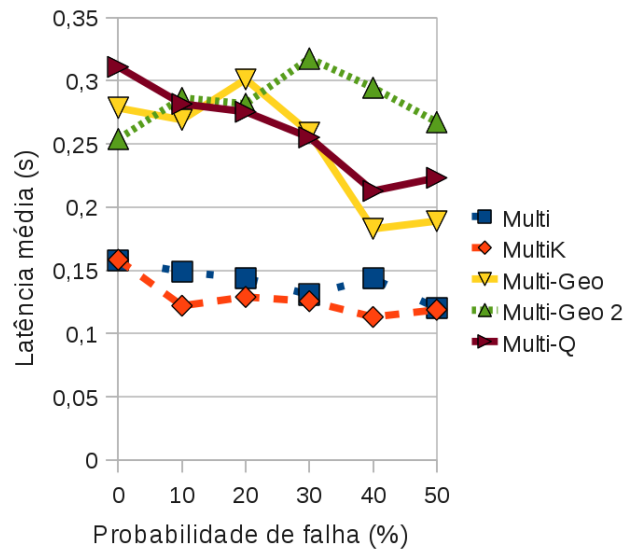


Figura 4.14. Variação da latência média dos protocolos em função da probabilidade de falha dos nós.

Capítulo 5 - Conclusões e Trabalhos Futuros

Neste trabalho foi apresentado o *Multi-Q*, uma versão modificada do protocolo *Multi-Geo*. As modificações propostas incluíram a divisão de regiões virtuais baseadas no número de nós em cada região da área, a limitação na potência de transmissão para difusões realizadas exclusivamente para nós no interior de um agrupamento e a eleição dinâmica de novos nós líderes quando o atual falha.

Depois da implementação das modificações sugeridas neste trabalho, várias simulações foram executadas com o objetivo de comparar e avaliar o novo protocolo em diferentes tipos de cenários. As diferenças nos cenários simulados incluíram variações na distribuição dos nós, no número de nós na rede, no número de nós em cada agrupamento, no posicionamento dos nós líderes dentro de cada agrupamento e na probabilidade de falha dos nós.

Várias comparações entre as diferentes versões dos protocolos apresentados foram feitas e demonstraram uma melhoria no consumo energético da nova versão, no entanto, evidenciou um aumento na latência média da rede. Essa solução de compromisso é aceitável em muitas aplicações reais de RSSF uma vez que a energia é um recurso escasso nesse tipo de rede.

Assim, o protocolo de roteamento apresentado neste trabalho é uma boa alternativa para redes que apresentam uma alta densidade de nós em sua totalidade ou em partes. Além disso, como herança do protocolo *Multi*,

que é um antecessor do protocolo Multi-Geo, esta nova proposta é um protocolo com duas estratégias de roteamento, que é recomendado para redes onde a taxa de incidência de eventos pode variar. Assim, ele adota uma estratégia reativa quando o número de eventos é baixo e uma estratégia proativa quando esse número aumenta.

Como limitação do protocolo *Multi-Q*, pode-se destacar o uso de agrupamentos estáticos, não sendo possível a reorganização dos mesmos quando a topologia da rede é modificada devido a falhas, inserções de novos nós ou até a movimentação dos mesmos, fazendo com que passem de uma região virtual para outra. Assim, propostas de trabalhos futuros podem incluir a melhoria do protocolo nesse aspecto, possibilitando que os próprios nós realizem a divisão dos agrupamentos dinamicamente e implementando algoritmos de localização dos nós, como os já referenciados neste trabalho.

Pode-se ainda alterar o método atual de divisão dos agrupamentos, que divide um agrupamento em quatro novos com tamanhos iguais, por um método que considera a densidade de nós em cada área do agrupamento, criando quatro novas regiões virtuais de tamanhos diferentes, cada uma com um número de nós aproximadamente igual. Isso faria com que os agrupamentos da rede fossem mais homogêneos em termos de número de nós, possibilitando uma definição mais exata do número de nós por agrupamento, já que o método atual define apenas o número máximo de nós, permitindo assim, uma variação no número de nós em cada agrupamento de zero até o valor máximo definido.

Uma outra limitação que pode ser destacada é que a estratégia usada para a eleição de novos nós líderes poderia gerar “buracos” no centro dos agrupamentos. Uma técnica que pode ser analisada para amenizar esse problema é o uso de nós líderes localizados no centro de massa do agrupamento, também chamado de centro absoluto, ao invés de localizá-los no centro geográfico do mesmo. Essa técnica poderia beneficiar

distribuições não uniformes dos nós. Também pode-se estudar técnicas de alternância do líder para evitar o consumo desigual de energia.

Capítulo 6 - Referências

- AL-KARAKI, J. N.; KAMAL, A. E. Routing Techniques in Wireless Sensor Networks: A Survey. In *IEEE Trans. on Wireless Communications*, 2004, v. 11(6), p. 6-28.
- BECKER, M. 2007. *WSN Simulation Tool Knowledgebase*. Available at: <http://mobilesummit2006.telecom.ece.ntua.gr/cruise/Public%20documents/wp123-wsn-simulation-tool-knowledgebase/simulation-tool-comparison-matrix/> [Accessed 9 September 2008].
- BULUSU, N.; HEIDEMANN, J.; ESTRIN, D. *GPS-less Low-Cost Outdoor Localization for very Small Devices*. Tech. Rep. 00-729, Computer Science Department, University of Southern California, Apr. 2000.
- CHIANG, C.C.; WU, H.K.; LIU, W.; GERLA, M. Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel. In *IEEE Singapore International Conference on Networks SICON'97*, pages 197–211, Singapore, Apr. 14-17 1997.
- CORREIA, L. H. A.; MACEDO, D. F.; SILVA, D. A. C.; SANTOS, A. L.; LOUREIRO, A. A. F.; NOGUEIRA, J. M. S. Controle de Potência de Transmissão em Protocolos MAC para Redes de Sensores Sem Fio. In *XXII Simpósio Brasileiro de Telecomunicações – SbrT'05*, Campinas – SP – Brasil, 04-08 Set. 2005.
- FERREIRA, R.; CARDOSO, J. M. P.; NETO, H. C. An Environment for Exploring Data-Driven Architectures, in *14th Int'l Conference on Field Programmable Logic and Applications (FPL'04)*, LNCS 3203, Springer-Verlag, 2004, pp. 1022-1026.

- FERREIRA, R.; BECK, A. C. S.; CARRO, L.; TOLEDO, A.; SILVA, A. A Java Framework to Teach Computer Architecture, in *New Trends and Technologies in Computer-Aided Learning for Computer-Aided Design*, vol.2 Springer Boston, 2005, pp. 25-35.
- FIGUEIREDO, C. M. S.; NAKAMURA, E. F.; LOUREIRO, A. A. F. Multi: A Hybrid Adaptive Dissemination Protocol for Wireless Sensor Networks. In *International Workshop on Algorithmic Aspects of Wireless Sensor Networks*, 2004, Turku. Lecture Notes in Computer Science, 2004. v. 3121. p. 171-186.
- GONÇALVES, R. T.; GOULART, C. C.; FIGUEIREDO, C. M.; LOUREIRO, A. A. F. Multi-K: Um Protocolo de Roteamento para Redes de Sensores sem Fio Usando Árvores de Espalhamento Parciais. In: *XXVI Simpósio Brasileiro de Redes de Computadores*, 2008, Rio de Janeiro – RJ – Brasil. XXVI Simpósio Brasileiro de Redes de Computadores, 2008. p. 1-14.
- HEINZELMAN, W. R.; KULIK, J.; BALAKRISHNAN, H. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks, in *Proceedings of the ACM MobiCom'99*, Seattle, Washington, 1999, pp. 174–185.
- HENDRICH, N. A Java-based Framework for Simulation and Teaching, in *3rd European Workshop on Microelectronics Education*, France, 18-19 May 2000, Kluwer Academic Publishers, pp. 285-288.
- HENDRICH, N. *HADES Tutorial*. Version 0.92. Hamburg, 124 p., 2006.
- KICZALES, G.; LAMPING, J.; MENDHEKAR, A.; MAEDA, C.; LOPES, C.; LOINGTIER, J.; IRWIN, J. Aspect-Oriented Programming. In: *Proceedings of the European Conference on Object-Oriented Programming (ECOOP)*. Springer-Verlag, Finland, 1997.
- LINDSEY, S.; RAGHAVENDRA, C.; SIVALINGAM, K. Data Gathering Algorithms in Sensor Networks Using Energy Metrics, in *IEEE Transactions on Parallel and Distributed Systems*, v. 13, p. 924-935, 2002.

- PARK, S.; SAVVIDES, A.; SRIVASTAVA, M. B. Simulating Networks of Wireless Sensors, in *Proceedings of the 33rd conference on Winter simulation*, Arlington, Virginia, 2001, p. 1330-1338.
- RICKER, T. *Epson's tiny GPS receiver will make everything location aware*. Disponível em: <<http://www.engadget.com/2009/02/12/epsons-tiny-gps-receiver-will-make-everything-location-aware/>>. Acesso em: 02 set. 2009.
- SAMET, H. The Quadtree and Related Hierarchical Data Structures. In *ACM Computing Surveys* 16(2) (June 1984) 187–260.
- SILVA, A. P. ; GOULART, C. C. Multi-Geo: Um Protocolo de Roteamento Hierárquico para Redes de Sensores Sem Fio. In *XXXV Conferência Latino-americana de Informática*, 2009, Pelotas-RS – Brasil. XXXV Latin American Informatics Conference. Pelotas-RS: UFPel / UCPel, 2009. v. CD. p. 1-10.
- SINGH, S.; RAGHAVENDRA, C. S. PAMAS: Power Aware Multi-Access Protocol with Signalling for Ad Hoc Networks, in *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 3, pp. 5–26, July 1998.
- SOUZA, V. B. C.; ROCHA, M. N.; FERREIRA, R. S. Simulating Wireless Sensor Network Protocols and Energy Consumption on a Hades Tool Extension, in *28th Annual International Conference on Computer Communications (IEEE INFOCOM 2009) - Student Workshop*, 2009, Rio de Janeiro - RJ - Brazil. 2009 Proceedings IEEE INFOCOM - Student Workshop, 2009.
- SOUZA, V. B. C.; ROCHA, M. N.; FERREIRA, R. S.; GOULART, C. C. An Implementation of the Multi-Geo Routing Protocol for Wireless Sensor Networks Using Quadrees, in *The Sixth ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems (6th ACM PE-WASUN 2009)*, 2009, Tenerife, Canary Islands. Proceedings of the 6th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks. New York, NY, USA : ACM, 2009. p. 23-26.

- WANG, Y.; LEDERER, S.; GAO, J. Connectivity-based Sensor Network Localization with Incremental Delaunay Refined Method, in *28th Annual International Conference on Computer Communications (IEEE INFOCOM 2009)*, 2009, Rio de Janeiro - RJ - Brazil. 2009 Proceedings IEEE INFOCOM, 2009.
- WOO, A.; CULLER, D. A Transmission Control Scheme for Media Access in Sensor Networks, in *Proceedings of ACM MobiCom'01*, Rome, Italy, July 2001, pp. 221–235.
- XUE, Y.; LEE, H. S.; YANG, M.; KUMARAWADU, P.; GHENNIWA, H.; SHEN, W. Performance Evaluation of NS-2 Simulator for Wireless Sensor Networks, in *Canadian Conference on Electrical and Computer Engineering*, 2007, April 2007, pp. 1372–1375.
- YE, W.; HEIDEMANN, J.; ESTRIN, D. Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks, in *IEEE/ACM Transactions on Networking*, vol.12, 2004, p. 493- 506.