UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

MÉTODO HÍBRIDO DE DETECÇÃO DE DEFEITOS EM REDES DE SENSORES MÓVEIS UTILIZANDO ALGORITMO DE LOCALIZAÇÃO

Dissertação de Mestrado

Leonardo Guedes da Luz Martins

Santa Maria, RS, Brasil
2012

MÉTODO HÍBRIDO DE DETECÇÃO DE DEFEITOS EM REDES DE SENSORES MÓVEIS UTILIZANDO ALGORITMO DE LOCALIZAÇÃO

Por

Leonardo Guedes da Luz Martins

Dissertação apresentada ao Curso de Mestrado do Programa de Pós-Graduação em Informática, Linha de Pesquisa em Microeletrônica e Processamento de Sinais, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Computação**

Orientador: Prof. Dr. Raul Ceretta Nunes

Co-orientador: Prof. Dr. João Baptista dos Santos Martins

Santa Maria, RS, Brasil
2012

Universidade Federal de Santa Maria
Centro de Tecnologia
Programa de Pós-Graduação em Informática

A Comissão Examinadora, abaixo assinada,
aprova a Dissertação de Mestrado

MÉTODO HÍBRIDO DE DETECÇÃO DE DEFEITOS EM REDES DE SENSORES MÓVEIS UTILIZANDO ALGORITMO DE LOCALIZAÇÃO

elaborada por
Leonardo Guedes da Luz Martins

como requisito parcial para obtenção do grau de
Mestre em Computação

COMISSÃO EXAMINADORA:

Raul Ceretta Nunes, Dr.
(Presidente/Orientador)

Leonardo Londero de Oliveira, Dr. (UFSM)

Sérgio José Melo de Almeira, Dr. (UCPel)

Santa Maria, 01 de Novembro de 2012

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Guedes da Luz Martins, Leonardo
MÉTODO HÍBRIDO DE DETECÇÃO DE DEFEITOS EM REDES DE SENSORES MÓVEIS UTILIZANDO ALGORITMO DE LOCALIZAÇÃO / Leonardo Guedes da Luz Martins.-2012.
90 f.; 30cm

Orientador: Raul Ceretta Nunes
Coorientador: João Baptista Santos Martins
Dissertação (mestrado) - Universidade Federal de Santa Maria, Centro de Tecnologia, Programa de Pós-Graduação em Informática, RS, 2012

1. Redes de Sensores 2. Detectores de Defeitos 3. Algoritmos de Localização I. Ceretta Nunes, Raul II. Santos Martins, João Baptista III. Título.

“Aos meus amados avós”

AGRADECIMENTOS

Inicialmente gostaria de agradecer ao meu amigo e orientador Raul Ceretta Nunes pela paciência e confiança com que me orientou ao longo destes 2 anos de mestrado. Suas ideias foram esclarecedoras e de fundamental importância no decorrer deste trabalho.

Também gostaria de agradecer em igual porcentagem ao meu co-orientador João Baptista dos Santos Martins, pelas longas conversas e grande atenção com que teve para comigo. Assim como ao professor Leonardo Londero de Oliveira pelas ideias e esclarecimentos sobre o tema abordado.

Gostaria de agradecer a toda minha família, inicialmente a Elenize pelas noites de correção de português e leituras sobre o trabalho mesmo que fosse de seu pouco entendimento, ao carinho e paciência com que teve comigo ao longo dos últimos anos. A meus pais João e Luccianne pelo apoio e suporte que sempre me deram desde o início de meus estudos. Ao amor com que me criaram e educaram com certeza sem tamanha dedicação este sonho não seria possível.

Não poderia esquecer de agradecer também aos meus irmãos, amigos de verdade que terei para a vida toda e em quem confio a minha própria vida. Aos meus amados avós a quem dedico este trabalho Luiz, Cecy, Dalva e Simão, pelo exemplo de amor e companheirismo que defenderam por toda a vida, de perto ou de longe, daqui ou de outro mundo me ensinaram valores que levarei pelo o resto de minha vida. Agradeço a todos meus tios e demais familiares, assim como a toda equipe da Chip Inside Engenharia e Tecnologia, GMicro e amigos próximos, sem os quais toda a caminhada seria bastante mais penosa.

Agradeço enfim, por ter tido a honra de ter pessoas tão iluminadas ao meu lado. Muito Obrigado

RESUMO

Os avanços tecnológicos decorrentes da miniaturização de sistemas eletrônicos nas últimas décadas foram determinantes para o alcanço da tecnologia atual. A microeletrônica, em conjunto com estudos de redução de consumo de energia em dispositivos eletrônicos, propiciou que as redes de sensores móveis se difundissem rapidamente ao redor do mundo. Uma rede de sensores móveis é composta por diversos nodos sensores, que trabalham colaborativamente para alcançar um determinado objetivo. Cada nodo da rede possui camadas e aplicações com fins específicos, dois dos mais importantes algoritmos presentes na maioria das aplicações em redes de sensores são os detectores de defeitos e os algoritmos de localização.

Os detectores de defeitos são algoritmos que normalmente são executados na camada de aplicação e seu objetivo é detectar através da troca de mensagem entre eles possíveis nodos falhos na rede. O algoritmo detector de defeitos é bloco fundamental em aplicações distribuídas e tolerantes a falhas. Por outro lado, o algoritmo de localização é uma proposta alternativa a utilização do GPS. Por meio da troca de mensagem entre os nodos, a partir de um ponto de referência o algoritmo de localização calcula as coordenadas do nodo no espaço.

Diversas metodologias já foram propostas na bibliografia para ambos os algoritmos. Porém, mesmo com similaridades funcionais, não são encontradas pesquisas que integrem estes algoritmos, a fim de economizar recursos e melhorar a eficiência das redes de sensores.

Este trabalho propõe um método híbrido em que o detector de defeitos aproveita a troca de mensagens realizada pelo algoritmo de localização para realizar sua tarefa. A solução aperfeiçoa a troca de mensagens enviadas pelos nodos, evitando o envio de pacotes de dados. Como resultado, a solução diminui o consumo de energia dos nodos e melhora o desempenho do sistema.

Palavras Chave: Redes de Sensores, Algoritmos de Localização, Detectores de Defeitos.

ABSTRACT

The technological advances resulting from electronic systems miniaturization were crucial to the reach of current technology. The microelectronics and studies to reduce power consumption in electronic devices provided that mobile sensor networks become increasingly used. A mobile sensor network is composed of several sensors nodes, which work collaboratively to achieve a certain goal. Each network node has layers and applications for specific purposes, two of the most important algorithms present in most applications in sensor networks are failure detectors and localization algorithms.

Failure detectors are algorithms that check faulty nodes in the network by exchanging messages. This algorithm is a fundamental block in distributed applications and fault tolerant. Moreover, the localization algorithm is an alternative to GPS use. Using the content of messages transmitted by nodes, the algorithm computes a location coordinates of the node from a reference point.

Several methodologies have been proposed in literature for both algorithms. However, even with functional similarities are not found searches uniting these algorithms in one in order to save resources and improve efficiency in wireless sensor networks.

This study proposes a hybrid method in which the failure detector takes advantage of messages sent by localization algorithm to perform its task. This solution optimizes the messages exchange sent by nodes, reducing the transmission of data packets. As result, the solution reduces the power consumption of nodes and improves system performance.

Keywords: Sensor Networks, Localization Algorithms, Failure Detectors.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

AOA	*Angle of Arrival*
CAB	*Concentric Anchors-Beacons*
DARPA	*United States Defense Advanced Research Projects Agency*
FFD	*Friedman Failure Detector*
GPS	*Global Positioning System*
MANETS	*Mobile Ad Hoc Networks*
MCL	*Monte Carlo Localization*
PRNET	*Packet Radio Network*
RF	*Radio Frequency*
RSSF	*Redes de Sensores Sem Fio*
RSSI	*Received Signal Strength Indicator*
TOA	*Time of Arrival*
TDOA	*Time Diference of Arrival*

SUMÁRIO

1. INTRODUÇÃO

Desde os primórdios de sua existência o ser humano tem se acostumado a monitorar eventos, seja na simples contagem de animais de um rebanho, no comportamento de uma determinada caça, até o monitoramento das estações do ano e épocas propícias a pesca. Mesmo sem os conhecimentos detalhados e específicos que temos disponíveis atualmente, o homem de antigamente já sabia que ao monitorar certos processos teria informações relevantes que o ajudariam a tomar decisões corretas.

Obviamente, com o passar dos anos, o homem foi monitorando eventos de maior complexidade, que trouxessem informações ainda mais relevantes para a tomada de decisão, e para isto a evolução da tecnologia se tornou parte indispensável.

Como exemplos desta evolução tecnológica pode-se citar o computador e a internet como dois dos principais responsáveis pelo desenvolvimento de outras tecnologias de monitoramento. Os avanços na fabricação de transistores é outro exemplo que contribuiu muito para que houvesse uma redução do consumo de potência em dispositivos eletrônicos, o que possibilitou o crescimento da área de redes de sensores, muito utilizada hoje para o monitoramento de eventos e processos. Atualmente, utilizam-se redes de sensores para fazer o monitoramento e controle de condições ambientais, animais, processos industriais, dentre inúmeras outras aplicações.

Em cada nodo sensor espalhado pela rede de sensores móvel existem diversas camadas e aplicações com objetivos específicos, dentre as quais responsáveis por estimar a posição geográfica dos nodos e por monitorar o seu estado de funcionamento. Algoritmos de localização de nodos realizam o cálculo das coordenadas de uma posição em um dado sistema de coordenadas. Este processo demanda que existam pontos de referência para os quais seja possível receber ou trocar informações. Uma vez que existe esta disponibilidade, duas tarefas precisam ser completadas: relacionar o ponto desconhecido com os pontos de referência e utilizar esta informação para calcular estimativas de posição através de algoritmos (OLIVEIRA, 2009).

Os algoritmos detectores de defeitos monitoram o correto funcionamento de cada um dos nodos vizinhos através da troca de mensagens, neste processo cada nodo pode ser monitor, monitorado ou monitor/monitorado.

Tanto algoritmos de localização como detectores de defeitos são essenciais para o funcionamento correto dos nodos na rede de sensores, porém o seu funcionamento atual

requer que os nodos realizem dois processamentos paralelos, além de necessitarem enviar mensagens distintas para cada processo.

O presente trabalho foi desenvolvido com o objetivo de otimizar este processamento e esta troca de mensagens, diminuindo a incidência de envio de informações e aumentando a qualidade das informações fornecidas pelos nodos.

1.1. Definição do Problema

O surgimento de redes de sensores móveis trouxe a tona diversas limitações que inexistiam em redes estáticas, tais limitações tem decorrência direta no fato dos nodos da rede possuírem vida útil limitada devido a utilização de bateria como fonte de alimentação, o que também limita o desenvolvimento de aplicações robustas, já que estas devem possuir um baixo consumo de energia. O *tradeoff* entre consumo de energia e capacidade de processamento é o estudo ao qual se cercam grande parte das pesquisas do setor, de maneira a desenvolver aplicações otimizadas que não consumam muita energia em seu processamento, assim como na busca por fontes de alimentação alternativa que possibilitem uma maior vida útil ao sistema.

Os elementos que compõem o modelo de energia de um nodo sensor são basicamente a bateria, o rádio, o processador e os sensores. A bateria possui vida útil limitada e é responsável pelo fornecimento de energia ao nodo. O rádio é responsável pelo recebimento e transmissão de pacotes, o processador é o elemento de processamento central do nodo, e seu consumo varia de acordo com a frequência do *clock*. Finalmente os sensores são os dispositivos de sensoriamento e seu consumo varia de acordo com as características e grandezas de cada medida (SOUSA, 2009).

Segundo (SILVA, 2008), a transmissão de dados é a operação que mais consome energia, representando aproximadamente 50% do consumo de energia da rede, e soluções que venham a otimizar a transmissão de dados devem ser implementadas a fim de minimizar este elevado consumo de energia. Estudos indicam que de maneira geral a execução de 3000 instruções consomem a mesma quantidade de energia que o envio de 1 bit a 100 metros via rádio (TILAK, 2002).

Como descrito anteriormente, o consumo de energia em redes de sensores móveis pode ser considerada sua maior limitação. Ao mesmo tempo em que este consumo está diretamente ligado a transmissão de pacotes, que correspondem a quase metade do consumo de energia do nodo. Desta forma se pode afirmar que algoritmos que otimizem o envio de

mensagens tem impacto direto na autonomia da rede, além de contribuírem evitando *overhead* de mensagens e congestionamentos.

Em aplicações para redes de sensores é comum a perda de mensagens em ambientes bastante congestionados, quanto maior a quantidade de mensagens trocadas entre os nodos, maiores as chances de perda, e consequentemente de falsas informações. A recorrência na perda de mensagens influencia diretamente na qualidade da rede sem fio (BAGGIO, 2010; TAN, 2006).

Como podemos observar acima, a quantidade de mensagens trocadas entre os nodos é um evento que pode ocasionar diversos problemas na rede, seja pelo aumento da possibilidade de se perderem informações, aumento do consumo de energia, congestionamento da rede, escalabilidade dos nodos, dentre outros obstáculos que advém desta comunicação.

Pensando nisso a comunidade científica busca maneiras de aperfeiçoar esta troca de mensagens, enviando dados mais significativos em um mesmo pacote. Porém o desafio de unir processos distintos em um único pode apresentar limitações consideráveis, já que cada algoritmo possui uma função específica de funcionamento.

O modelo desenvolvido neste trabalho tem o objetivo de otimizar a troca de mensagens entre os nodos, unindo as informações de localização e detecção de defeitos de forma que ambos os processos colaborem para uma diminuição na incidência de mensagens enviadas e consequente redução do consumo de energia do nodo, proporcionando assim uma melhora de desempenho na rede.

1.2. Contribuição

A proposta de um método que consiga unir as funcionalidades de localização e detecção de defeitos em uma rede de sensores móveis é por si só uma arquitetura singular em comparação aos modelos utilizados atualmente. O funcionamento conjunto destes algoritmos proporciona uma série de vantagens ao desempenho da rede, dentre as principais contribuições desta arquitetura podemos citar:

- Diminuição na troca de mensagens entre os nodos;
- Integração de algoritmos de localização e detectores de defeitos;
- Diminuição do consumo de energia dos nodos visto a redução no envio de pacotes;

Além destes impactos diretos, a topologia contribui indiretamente na diminuição de congestionamento e consequente perdas de mensagens na rede, visto que um número menor de mensagens será transmitido para a realização das mesmas tarefas. Essa redução de perdas de mensagens e consumo de energia impacta diretamente na qualidade de comunicação da RSSF.

Por fim é importante salientar que não se busca neste trabalho melhorar o desempenho dos algoritmos de detecção de defeitos ou localização separadamente. Apesar deste trabalho abrir a possibilidade de trabalhos futuros neste sentido, esta dissertação mantém o foco na melhora de desempenho da rede com a união destes dois modelos.

1.3. Organização da Dissertação

O restante do presente trabalho está organizado da seguinte forma. O capítulo 2 apresenta uma revisão sobre algoritmos detectores de defeito. Este capítulo também apresenta uma breve descrição de redes ad hoc, assim como cita as diferenças básicas de detectores para redes estáticas e móveis e os principais desafios desta migração.

O capítulo 3 apresenta uma revisão de algoritmos de localização, assim como seu objetivo, suas funcionalidades básicas e as diferentes técnicas utilizadas para estimativa de coordenadas.

No capítulo 4 é apresentada a topologia hibrida desenvolvida neste trabalho. É apresentado neste capítulo sua metodologia de funcionamento detalhada, as etapas para implementação e as diferenças que este algoritmo tem em relação aos demais modelos existentes.

O capítulo 5 descreve o ambiente de simulação que foi utilizado para validação do modelo, o simulador desenvolvido por (HU; EVANS, 2004). Assim como são apresentados os resultados de simulação, bem como a comparação destes resultados com os demais modelos existentes.

Por fim, o capítulo 6 descreve as conclusões desta dissertação, assim como as propostas de melhorias e projeções futuras do método desenvolvido neste trabalho.

2. DETECTORES DE DEFEITOS

Este capítulo apresenta uma revisão sobre detectores de defeito, assim como suas principais características e aplicações em topologias de redes distintas.

2.1. Sistemas Distribuídos e Conceitos Iniciais sobre Detectores de Defeito

Um sistema distribuído pode ser descrito como sendo um sistema que interliga diversos computadores individuais (nodos de processamento) em uma rede. De maneira similar, pode-se considerar um sistema distribuído como uma arquitetura que contem um número finito de processos $\Omega = \{p_{1,} p_2 ... p_3\}$, aonde cada processo pode se conectar a qualquer outro processo do sistema. Em um sistema assíncrono, cada nodo dentro do sistema em questão possui um relógio interno que independe do relógio interno dos demais e sua comunicação é realizado pela troca de mensagens entre os nodos através de um canal de comunicação.

É importante salientar que uma rede de computadores não é sinônimo para um sistema distribuído, porém fornece toda a infra-estrutura para se criar um (GRACIOLLI; NUNES, 2006).

Apesar de todas as vantagens que um sistema distribuído pode trazer, também fica evidente que este tipo de modelo deve ser capaz de suportar e tolerar falhas. Do contrário um pequeno desvio em algum dos nodos da rede será por si só, capaz de interromper todo o processo, causando inoperabilidade e indisponibilidade dos serviços. “Um sistema distribuído é um sistema no qual um defeito de um computador cuja existência você ignorava, pode tornar seu computador inutilizável” (LAMPORT, 1987).

Diferentemente do mundo ideal, no mundo real confiabilidade e disponibilidade são itens bastante distantes da perfeição, o que indica que falhas sempre irão ocorrer, porém as consequências que irão trazer ao serviço podem ser mais ou menos agravantes, de acordo com as técnicas de tolerância utilizadas no sistema. Como (WEBER, 2001) descreve, “Falhas são inevitáveis, mas as consequências das falhas, ou seja, o colapso do sistema, a interrupção no fornecimento do serviço e a perda de dados, podem ser evitadas pelo uso adequado de técnicas viáveis e de fácil compreensão. O conhecimento dessas técnicas habilita o administrador de sistemas a implementar as mais simples, ou exigir dos fornecedores e desenvolvedores de sistemas soluções que as incorporem”.

Por definição, neste trabalho uma falha causa um erro, que causa um defeito. Defeito é a consequência da falha perceptível ao usuário do sistema. A falha pode ficar latente no sistema gerando um erro ou defeito. Por isto detectam-se defeitos e não falhas, e toleram-se falhas e não defeitos. A Figura 1 ilustra a diferença entre falha, erro e defeito.

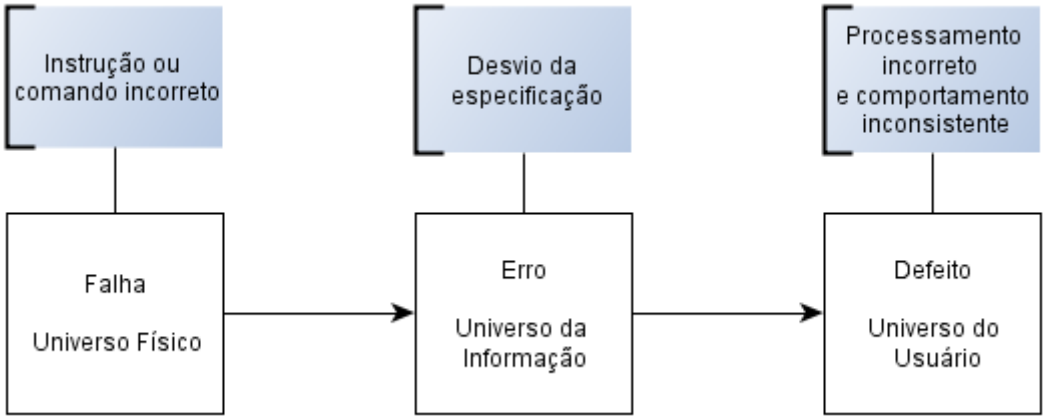

Figura 1 - Diferença entre falha, erro e defeito

As falhas em um sistema distribuído (causa inicial de um defeito) podem ser classificadas segundo (JALOTE, 1994), como:

- **Falha por colapso (*crash fault*)**: parada ou perda do estado interno, uma falha por colapso cessa o funcionamento do processo eternamente;
- **Falha por omissão (*omission fault*)**: o processo fica sem resposta para alguns pedidos, omite o envio ou recebimento de mensagens;
- **Falha por temporização (*timing fault* ou *performance fault*)**: ocorre quando o processo viola as especificações de temporização assumidas pelo sistema, ou seja, respostas atrasadas ou adiantadas (defeito irrelevante em sistemas assíncronos);
- **Falha arbitrária ou bizantina (*byzantine fault*)**: apresenta comportamento arbitrário e imprevisível, ou seja, sua ocorrência pode mudar o conteúdo da mensagem, duplicá-la, transmitir mensagens não solicitadas e até mesmo levar o sistema ao colapso.

Ao mesmo tempo em que se descreve que um sistema distribuído deve utilizar técnicas de tolerância a falhas para tornar seu processamento mais robusto, também é

essencial citar que uma de suas principais características é a utilização da redundância, o que possibilita que mesmo na presença de falhas o sistema não pare de funcionar (GARTNER, 1999).

Uma característica importante dos sistemas distribuídos é que, tanto o processamento do erro, como o tratamento das falhas é realizado através de algoritmos que se baseiem na troca de mensagens (NUNES, 2003) (POWELL, 1998). Detectores de defeito podem ser definidos como algoritmos distribuídos que fornecem informações de defeitos em componentes monitoráveis através da troca de mensagens (FELBER ET al., 1999).

2.2. Detectores de Defeito Não Confiáveis

Um detector de defeitos pode ser descrito como um algoritmo que realiza ações de monitoramento de um número finito de nodos, através da troca de mensagens entre eles. Em um modelo Push de monitoramento, define-se processo monitor aquele que recebe informações, e processo monitorado aquele que envia informações.

Caso o ambiente seja síncrono o detector de defeitos poderá dar garantia da ocorrência ou não ocorrência de falha em um dado processo, pois cada processo deverá ter um tempo estabelecido e confiável para realizar uma comunicação. Se todos os processos podem oferecer esta garantia dizemos que o detector de defeitos é confiável. Porém, se houver processos que não possam estabelecer um tempo limite para se comunicarem, o detector de defeitos não poderá afirmar com certeza que uma falha ocorreu neste processo, apenas suspeitar. Para esta situação dizemos que o detector de defeitos é não confiável.

Os detectores de defeitos não confiáveis, ou seja, que podem atuar em ambientes assíncronos, foram definidos pela primeira vez por Chandra e Toueg (CHANDRA; TOUEG, 1996). Um sistema distribuído é assíncrono quando não existe qualquer obrigação no tempo de espera, e não é possível ser feita nenhuma suposição específica de tempo garantida. Apesar de proporcionar uma semântica simplificada e facilitar aplicações programadas na base, o modelo assíncrono sujeito a falhas não consegue identificar a ocorrência de falhas deterministicamente, ou seja, não é possível afirmar se o processo realmente falhou ou se esta apenas lento (FISCHER et al., 1985).

Baseando-se, então, no comportamento dos detectores de defeitos, Chandra e Toueg (CHANDRA; TOUEG, 1996) estabeleceram sua classificação por duas propriedades, (1) abrangência (*completeness*) e (2) precisão (*accuracy*). A primeira caracteriza a capacidade do detector de defeitos de suspeitar de todos os processos defeituosos, enquanto a precisão

caracteriza a capacidade do detector de não suspeitar de processos corretos. Assim, de acordo com estas duas propriedades, o detector de defeitos pode pertencer as classes observadas na Tabela 1.

Tabela 1 - Classes dos Detectores de Defeito

Completeness	**Accuracy**			
	Strong	Weak	Eventual Strong	Eventual Weak
Strong	*Perfect* $\mathcal{P}$	*Strong* $\mathcal{S}$	*Eventually Perfect* $\diamond\mathcal{P}$	*Eventually Strong* $\diamond\mathcal{S}$
Weak	$\mathcal{Q}$	*Weak* $\mathcal{W}$	$\diamond\mathcal{Q}$	*Eventually Weak* $\diamond\mathcal{W}$

- (1) **Strong Completeness** (Abrangência Forte): todo processo falho será permanentemente suspeito por todos processos corretos.
- (1) **Weak Completeness** (Abrangência Fraca): algum processo correto suspeitará permanentemente de todos os processos falhos.
- (2) **Strong Accuracy** (Precisão Forte): nenhum processo é suspeito antes de ter realmente falhado.
- (2) **Weak Accuracy** (Precisão Fraca): pelo menos um processo correto jamais será suspeito pelos outros processos.
- (2) **Eventual Strong Accuracy** (Precisão Eventualmente Forte): todos os processos somente serão considerados suspeitos após realmente falharem.
- (2) **Eventual Weak Accuracy** (Precisão Eventualmente Fraca): algum processo correto nunca será suspeito antes de ter realmente falhado.

2.3. Algoritmos de Detecção de Defeitos em Redes Estáticas

Neste item serão revisadas as principais estratégias para detecção de defeitos para redes com nodos fixos: Push, Pull, Dual, Heartbeat e Gossip.

2.3.1. Modelo Push

No modelo Push (FELBER et al., 1999) os nodos monitorados na rede atuam enviando mensagens para os nodos monitores, que para este caso são passivos. A cada intervalo de

tempo ‘T’, os nodos monitorados devem enviar uma mensagem do tipo ‘*I am alive!*’ ao nodo monitor, caso o nodo monitor não receba a mensagem no intervalo de tempo pré-determinado (*timeout*), o nodo monitorado é considerado suspeito.

Uma das vantagens deste modelo é o fato de ser unidirecional, diminuindo assim o número de mensagens trocadas pelo sistema.

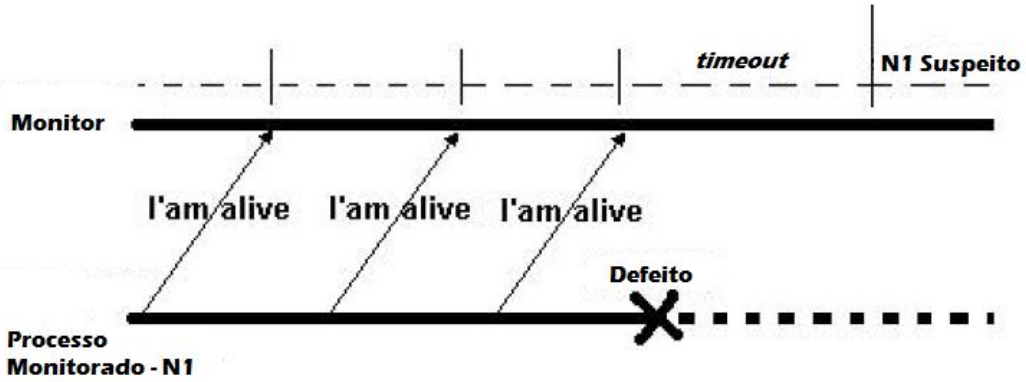

Figura 2 - Detector de Defeitos Push (adaptado de TANENBAUM, 1995)

A Figura 2 ilustra o detector Push. Nota-se que o nodo monitorado (N1) realiza 3 envios de mensagens *‘I am alive!’* ao monitor que não levanta suspeita a respeito do funcionamento do respectivo processo. A partir do momento que o nodo monitorado não envia nenhuma mensagem ao nodo monitor em um intervalo de tempo ‘*timeout’*, uma suspeita é levantada pelo processo monitor até que N1 envie outra mensagem ‘*I am alive!*’.

2.3.2. Modelo Pull

No modelo Pull os nodos monitorados são passivos, enquanto os monitores são os ativos, ou seja, o monitor envia mensagens do tipo ‘*It’s alive?’* periodicamente aos nodos monitorados, que devem responder ‘*I am alive!’*. Quando um nodo monitor não recebe a resposta em um intervalo de tempo mínimo *timeout*, o processo monitor levanta uma suspeita ao respectivo nodo.

Diferentemente do modelo Push, o modelo Pull é bidirecional, o que o torna menos eficiente quanto ao congestionamento na troca de mensagens. Porém os seus objetos monitoráveis não precisam estar ativos e nem conhecer o timeout do sistema.

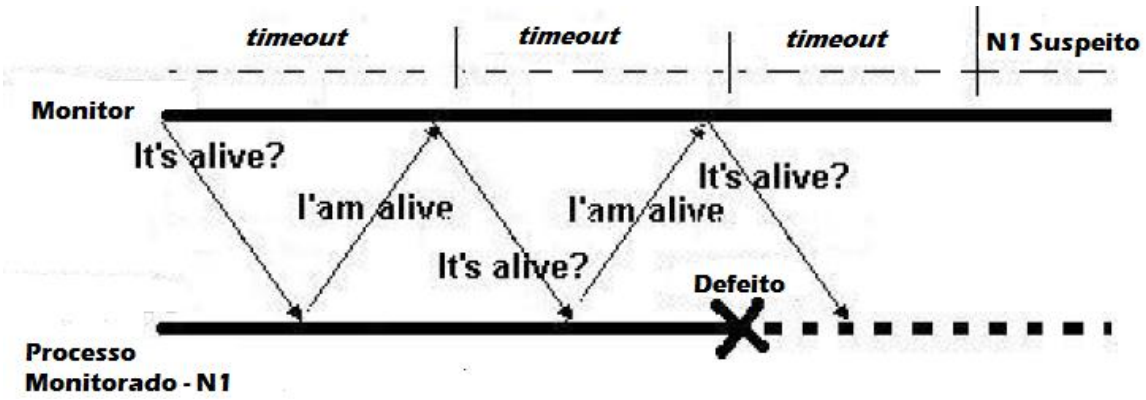

Figura 3 - Detector de Defeitos Pull (adaptado de TANENBAUM, 1995)

A Figura 3 ilustra o modelo Pull descrito anteriormente. Observe que após o segundo *timeout* o nodo monitorado 'N1' não envia a mensagem de confirmação ao monitor, ultrapassando assim tempo de *timeout* e fazendo com que o nodo monitor levante uma suspeita em relação do nodo monitorado.

2.3.3. Modelo Dual (Push/Pull)

O modelo dual nada mais é do que a combinação dos modelos Push e Pull (FELBER et al., 1999). Em uma primeira fase o algoritmo se comporta como modelo Push, ou seja, o nodo monitorado envia a cada intervalo de tempo 'T1' uma mensagem '*I am alive!*' para o nodo monitor. Caso o nodo monitor não receba a mensagem do nodo monitorado dentro do *timeout* T1 pré-definido, o modelo passa a segunda fase, baseada no modelo Pull.

Então, o nodo monitor envia uma mensagem '*Are you alive?*' ao nodo monitorado e aguarda retorno dentro de um intervalo de tempo 'T2'. Se o nodo monitorado não responder ele é considerado falho, caso o mesmo responda o algoritmo volta a fase inicial Push. A Figura 4 ilustra o modelo.

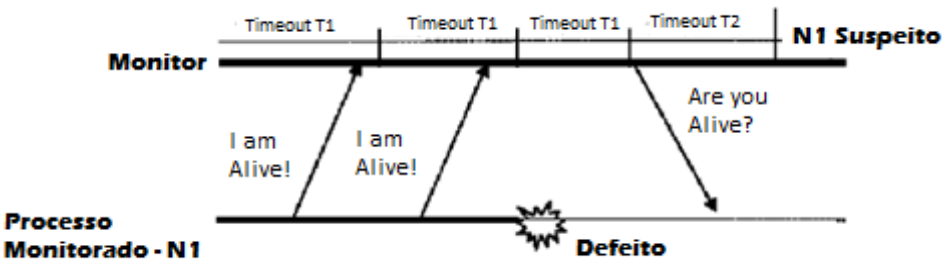

Figura 4 - Detector de Defeitos Dual

2.3.4. Modelo Heartbeat

No detector de defeitos *Heartbeat,* cada nodo monitor possui um contador de *heartbeats* para cada nodo monitorado em seu sistema. Assim, periodicamente, cada nodo monitorado envia uma mensagem "*I am alive!*" para os nodos monitores (AGUILERA; CHEN; TOUEG, 1997).

Quando os monitores recebem esta mensagem (*heartbeat*), um contador específico do nodo monitorado que enviou a mensagem é incrementado. A suspeita de falha é levantada quando o monitor não recebe a mensagem "*I am alive!*".

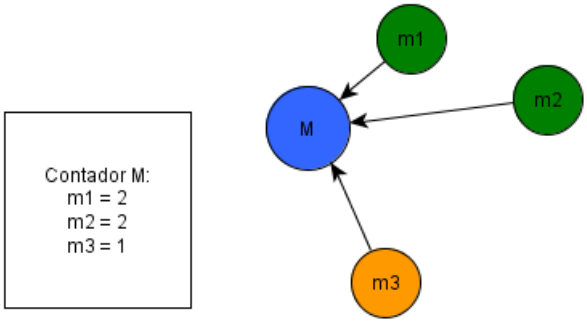

Figura 5 - Detector de Defeitos Heartbeat (AGUILERA; CHEN; TOUEG, 1997)

A Figura 5 ilustra o modelo *Heartbeat.* Nela pode-se observar que o nodo monitor é o 'M', enquanto 'm1', 'm2'e 'm3' são os nodos monitorados. Conforme nos mostra a tabela com o contador de *heartbeat* contida em 'M', os nodos 'm1' e 'm2' possuem 2 contagens, enquanto m3 possui somente 1 no mesmo período de tempo, o que indica que m3 não enviou a mensagem "*I am alive!*" durante um período de tempo, tornando-o então suspeito de falha.

2.3.5. Modelo Gossip

No modelo *Gossip* proposto por (RENESSE; MINSKY; HAYDEN, 1998), cada nodo pertencente a rede é tanto monitor quanto monitorado, e tem sua própria lista contendo o endereço e o contador de *heartbeat* para cada nodo que também pertence ao sistema.

Como o próprio nome já sugere, este modelo é conhecido por divulgar suas informações por "fofoca", ou seja, após cada intervalo de tempo 'T_G', cada nodo escolhe

aleatoriamente um ou mais nodos da rede e envia sua lista de endereços atualizada. O nodo receptor da mensagem irá então receber uma nova lista de endereços e contadores de *heartbeat*. Após comparar a lista recebida com a sua própria, o nodo irá manter os maiores valores dentre as duas listas, assim como, obviamente, incrementar o contador do nodo que enviou a mensagem.

Caso o contador de um determinado nodo não seja incrementado em um determinado intervalo de tempo 'T_{Fail}', o nodo em questão é considerado suspeito.

Ocasionalmente, cada nó faz *broadcast* de sua lista para tornar-se conhecido pela primeira vez, e para recuperar-se de partições na rede.

A principal vantagem do algoritmo Gossip é sua capacidade de tolerar a perda de algumas mensagens, porém caso muitas mensagens se percam o detector pode levantar suspeitas falsas no sistema. Outra vantagem do modelo é a não utilização de envio de mensagens por *Broadcast* como ocorre no modelo *Heartbeat*, o envio aleatório diminui drasticamente o congestionamento da rede ocasionado pela troca de mensagens.

O modelo Gossip básico, descrito acima, pode ser estendido numa versão multi-nível, permitindo ainda mais a redução no número de mensagens e uma maior escalabilidade do sistema. Neste modelo, o algoritmo utiliza a estrutura de domínios e seu mapeamento em endereço IP, o que possibilita identificar domínios e sub-redes, além de mapear diferentes níveis. Como a maioria das mensagens é enviada pelo protocolo básico na sub-rede e as mensagens trocadas entre os domínios são poucas, a versão multi-nível do modelo *Gossip* se torna mais escalável (TURCHETTI, 2006).

2.4. Redes Ad Hoc

As chamadas redes ad hoc surgiram no inicio da década de 1970 quando a US DARPA (ou *United States Defense Advanced Research Projects Agency*) iniciou o projeto PRNET (ou *Packet Radio Network*), para explorar o uso de redes na comunicação de pacotes via radio num ambiente tático militar. Basicamente redes ad hoc são arquiteturas aonde os nodos conseguem se comunicar diretamente com outros nodos sem que seja necessária a criação de uma infra-estrutura de rede. Ou seja, em uma arquitetura ad hoc os nodos formam uma rede que dispensa a utilização de um ponto de acesso central, de modo que cada dispositivo da rede atua como se fosse um roteador, atuando colaborativamente e enviando informações que vêm de dispositivos vizinhos.

Em redes convencionais temos a necessidade do ponto de acesso, pelo qual todas as informações da rede devem passar. No caso da rede ad hoc os próprios nodos podem se comunicar diretamente entre si, aumentando exponencialmente a flexibilidade e mobilidade da rede.

Como em uma rede ad hoc todos os nodos são monitores/monitorados e não temos um *gateway* ou ponto de acesso, o roteamento das mensagens passa a ser uma preocupação central da topologia e a colaboração e cooperação entre os nodos da rede passa a ser de fundamental importância na troca de informações. Assim, cada nodo possui uma participação de grande importância para que as mensagens consigam atingir o seu destino, ainda mais em redes móveis aonde a mobilidade pode dificultar bastante a criação de rotas (GRACIOLI; NUNES, 2007).

As redes ad hoc sem fio também são conhecidas por MANETS (*Mobile Ad Hoc Networks*), e atualmente vem sendo amplamente utilizadas por celulares, notebooks, redes de sensores, etc. A Figura 6 ilustra um modelo de MANET, na imagem é possível observar que não existe um nodo central, todos os nodos da rede, sejam os celulares dos soldados como o computador da base, são pontos de igual relevância, ou seja, não há hierarquia na rede.

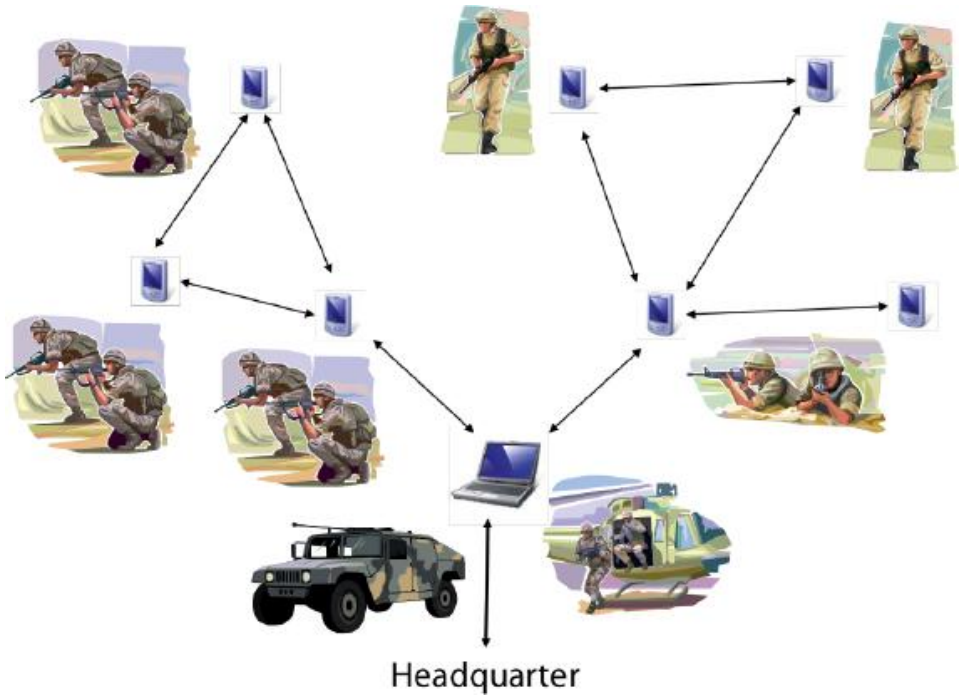

Figura 6 - Rede Ad Hoc (não possui ponto de acesso central)

2.4.1. Redes de Sensores

As redes de sensores podem ser definidas como sendo uma rede formada por diversos sensores (nodos), especializados ou não, que buscam cooperar entre si para monitorar um determinado espaço ou processo que se queira medir (PEREIRA; AMORIM; CASTRO, 2003). Os nodos especializados possuem capacidades adicionais superiores aos demais nodos da rede, sejam elas referentes a um dispositivo de rádio com maior alcance, maior poder de processamento, sensores específicos, um sistema de auto-localização ou mesmo por uma tarefa específica dentro da rede. Normalmente este tipo de elemento em uma rede de sensores ou em uma rede *ad hoc* é pequeno em relação aos demais nodos. Isto é devido a questões de custo de implementação, manutenção ou fator de forma final do hardware (OLIVEIRA, 2009) (AKYILDIZ et al., 2002).

Segundo (AKYILDIZ et al., 2002), "As redes de sensores são compostas por nós responsáveis pelo sensoriamento e pelo envio das informações coletadas a um nó que agrega informações. Esse nó pode ser um outro nó comum da rede ou um nó de maior capacidade, em todo o caso a informação tende sempre a fluir na direção de um ponto centralizador que pode ser um computador de maior porte".

Dependendo da aplicação a ser monitorada os sensores podem ser estáticos ou móveis. As redes estáticas são mais fáceis de serem monitoradas em comparação com as móveis devido aos problemas inerentes a mobilidade que serão apresentados na seção 2.4.2. Em uma rede móvel cada nodo possui restrições de processamento de dados e consumo de energia devido a bateria, que possui tempo limitado de vida útil (PANTAZIS; VERGADOS, 2007).

O nodo sensor é composto basicamente por 6 elementos: sensor responsável pela medição, processador, memória para guardar as informações, dispositivos transmissores e receptores responsáveis pela troca de mensagens com outros nodos da rede e bateria.

2.4.2. Limitações para Detectores de Defeitos em RSSF

Inicialmente os detectores de defeitos foram desenvolvidos para atuar em redes estáticas (LAN's e WAN's), porém com o surgimento e popularização das redes móveis, a necessidade de detectores de defeitos para este tipo de rede se tornou cada vez mais necessária. A reutilização dos detectores para redes estáticas exige adaptações para redes móveis, já que o ambiente desta nova topologia difere do primeiro (FRIEDMAN; TCHARNY, 2005), (HUTLE, 2004) e (TAI; TSO, 2004).

As principais diferenças são:

Qualidade da Comunicação Wireless: A qualidade na comunicação em redes sem fio é pior, se comparado a uma rede cabeada. Para o detector isto é um problema que gera a perda de mensagens e diminuição de desempenho da rede. Por exemplo, quando uma mensagem é perdida em uma troca de informações, um nodo pode ser considerado suspeito sendo que está em perfeito funcionamento, violando a propriedade da precisão. Assim como um nodo que identificou um defeito deve conseguir transmitir esta mensagem para os demais nodos operacionais da rede, do contrário estaríamos violando a propriedade da abrangência (GRACIOLI; NUNES, 2007).

Escalabilidade dos Nodos: A escalabilidade dos nodos tende ser maior em redes de sensores se fio. O detector de defeitos deve suportar um aumento significativo na quantidade de nodos e na troca de informações.

Mobilidade: Diferentemente de uma rede com nodos fixos, em uma RSSF o conceito de mobilidade se torna parte do detector de defeitos já que os nodos estão sempre em movimento. Por estarem sempre em movimento, em determinado momento 't_1' o nodo 'A' pode estar dentro do alcance de outro nodo qualquer 'B'. Porém, em um intervalo 't_2' o mesmo nodo 'A' pode se mover e ficar fora do alcance de transmissão do nodo 'B'. Neste caso o nodo 'A' não pode ser considerado suspeito por sair momentaneamente do alcance de 'B'. Em uma rede móvel o detector de defeitos deve prever uma situação como esta.

Consumo de Energia: Em uma rede móvel a única fonte de alimentação dos nodos é através de uma bateria. Diferentemente da rede estática, estas baterias possuem um limitado tempo de vida útil. Assim, é de suma importância que a troca de mensagens entre os nodos seja otimizada para o melhor desempenho e menor consumo possível, a fim de maximizar o tempo de vida da bateria.

2.5. Detectores de Defeito em Redes Móveis

Nesta seção são descritos os principais detectores de defeitos para redes móveis, assim como suas características e especificidades próprias. Os detectores de defeito baseados em formação de cluster (TAI; TSO; SANDERS, 2004) não serão abordados, pois, ocasionam maior *overhead*, o que deve ser evitado em redes que possuem recursos limitados, como as redes de sensores sem fio.

Os detectores de defeitos descritos a seguir são baseado no algoritmo Gossip.

2.5.1. Detector de Defeitos Friedman (FFD)

Em uma rede móvel sem fio um nodo móvel não pode suspeitar de outro nodo móvel somente porque um dos dois saiu da região de abrangência do outro por um determinado instante de tempo. Pensando neste problema de mobilidade, (FRIEDMAN; TCHARNY, 2005) propuseram um mecanismo para detecção de defeitos, baseando-se no modelo Gossip. Neste trabalho chamamos esta proposição de modelo Friedman, ou detector de defeitos Friedman.

Assim como no algoritmo Gossip, no detector Friedman cada nodo possui um contador de *heartbeat* para cada um dos demais nodos da rede com que manteve contato, a cada π unidades de tempo, todos os nodos da rede devem enviar aos nodos de sua região de abrangência sua lista de endereços com a identificação dos nodos ouvidos por ele e seus respectivos contadores de *heartbeat.*

Para evitar que o algoritmo levante falsas suspeitas, o algoritmo proposto por (FRIEDMAN; TCHARNY, 2005) permite a passagem de no máximo ‘ γ ’ *heartbeats* para não suspeitar de um nodo entre o recebimento de dois *heartbeats* consecutivos (GRACIOLI; NUNES, 2007). Assim, o algoritmo espera que os nodos estejam em movimento, hora estando dentro da região de abrangência de outro nodo, hora estando fora do alcance deste mesmo nodo.

No modelo Friedman o nodo que não receber uma mensagem de outro nodo que estava em sua região de abrangência não irá suspeitar deste nodo imediatamente, e sim irá aguardar a omissão de ‘ γ ’ *heartbeats* para levantar uma suspeita de falha, esta técnica é utilizada levando-se em conta o problema da mobilidade. O desafio deste modelo é encontrar o valor de ‘ γ ’ para cada nodo.

O detector de defeitos Friedman mantém as propriedades aonde garante que todos os processos serão eventualmente suspeitos (precisão eventualmente forte) e tem seu desempenho otimizado quanto maior a conectividade e o número de nodos da rede. Isto se deve ao fato de uma baixa conectividade resultar em um menor alcance de disseminação de detecção entre os vizinhos de cada nodo, aumentando a probabilidade de falsas detecções. Logo, quando maior o tamanho da rede, maior a conectividade entre os nodos, menor a probabilidade de falsas detecções.

Nota-se também que graças ao modelo Gossip é possível que os nodos detectem quais deles estão se movendo, desta forma um nodo pode receber um *heartbeat* atualizado de outro nodo que não pertence mais a sua região de abrangência. As Figuras 7 e 8 ilustram esta interação. Mesmo que o nodo 3 não esteja na região de abrangência do nodo 1, o seu status de funcionamento (contador de *heartbeat*) é repassado pelo nodo 2, que abrange ambos os nodos e vice-versa.

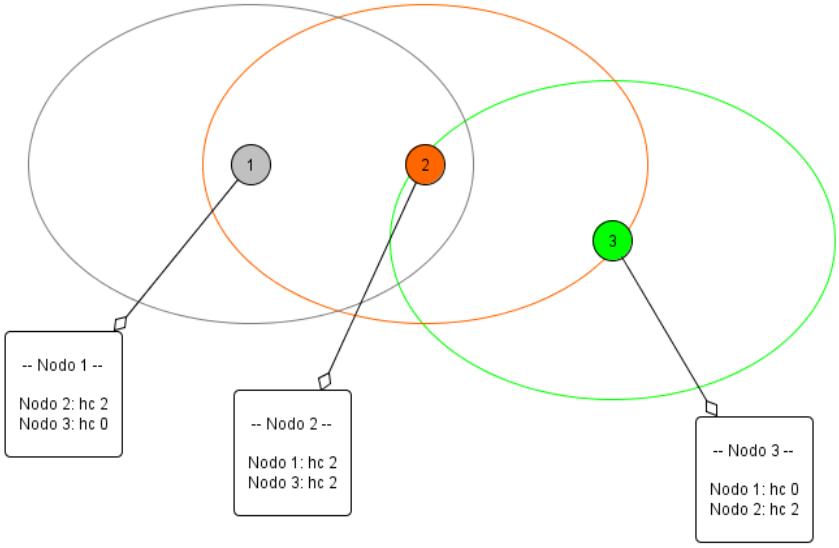

Figura 7 - Modelo Friedman: Interação entre os nodos em 't1'

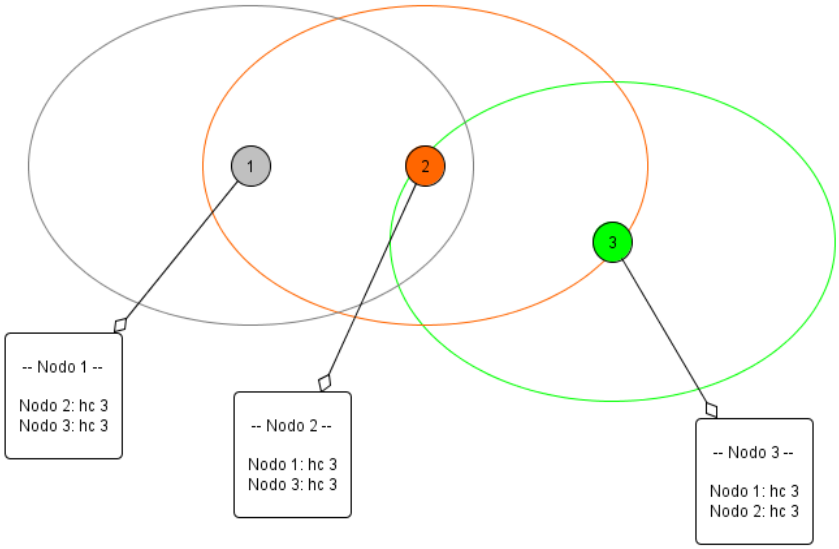

Figura 8 - Modelo Friedman: Interação entre os nodos em 't2'

Observe nas Figuras 7 e 8 que mesmo os nodos 1 e 3 não estabelecendo contato direto durante o intervalo de tempo ‘t1’, seu contador de *heartbeat* é atualizado em ambos em um intervalo ‘t2’ graças ao nodo 2 que envia sua tabela de *heartbeat* completa (e que contém ambos) por *broadcast.*

A Figura 9 ilustra a descrição do algoritmo Friedman básico para um nodo i qualquer.

1: **Notação do nodo i:**
2: $alive_i$ vetor dos contadores de *heartbeat* para cada nodo
3: $suspected_i$ vetor de *bitmap* dos nodos suspeitos
4:
5: **Inicialização do nodo i:**
6: $alive_i = [0,0...0]$
7: $suspected_i = [0,0...0]$
8: para todo j faça $suspect\_timer_i[j].set(\beta(\gamma_0))$ /* tempo inicial para considerar um nodo suspeito */
9:
10: **A cada π unidades de tempo:**
11: $alive_i[i] = alive_i[i] + 1$
12: $broadcast(alive, alive_i)$
13:
14: **No recebimento** $receive(alive, alive_j)$ **do nodo** j
15: $\forall k$, if$(alive_i[k] < alive_j[k])$
16: $suspected_i[k] = 0$
17: $suspect\_timer_i[k].set(\beta(\gamma_f)$
18: $alive_i = ArrayMax(alive_i, alive_j)$
19:
20: **Ao** $suspect\_timer_i[j]$**.timeout**:
21: $suspected_i[j] = 1$

Figura 9 - Algoritmo de detecção de defeitos Friedman (adaptado de FRIEDMAN; TCHARNY, 2005)

Observe na Figura 9 que o tempo para considerar um nodo suspeito é inicializado com o valor $\beta(\gamma_0)$, onde γ_0 representa o tempo máximo (*timeout*) para o recebimento do primeiro *heartbeat* do nodo mais distante no sistema.

Assim que o nodo i receber a lista de outro nodo qualquer (neste caso representado por j), o mesmo irá comparar os contadores de *heartbeats* das listas mantendo os maiores valores. No caso de receber um contador maior que o local um novo valor para o tempo de suspeita do membro é calculado.

Por fim, caso o nodo i não receba um valor de *heartbeat* do nodo j após a passagem do último tempo de suspeita calculado para este nodo, o mesmo é considerado suspeito.

2.5.2. Modelo Hutle

O algoritmo proposto por (HUTLE, 2004) é um modelo de detector de defeitos eventualmente perfeito para redes que não necessitem estar completamente conectadas. Uma

característica interessante deste modelo é o de que o detector não necessita conhecer todos os nodos da rede e sim o *jitter* (ou variação estatística do atraso na entrega de dados em uma rede) que existe na comunicação entre eles.

Assim como o modelo *Gossip* básico, cada nodo possui uma tabela (ou lista), que contém as seguintes informações:

- Contador de h*eartbeats* atualizado para os demais nodos com que teve contato;
- Contador da distância estimada entre os nodos;
- *Timestamp* contendo o último período (*round*) em que aquele nodo recebeu a mensagem dos demais nodos da rede.

Esta tabela é enviada pelos nodos aos seus vizinhos a cada intervalo de tempo pré-determinado. Se um nodo não é atualizado por um dado intervalo de tempo (*timeout*), uma suspeita sobre ele é levantada.

Outra característica interessante do modelo Hutle é o de conseguir reduzir a freqüência de encaminhamentos de mensagem pela distância, tornando assim a informação sobre os vizinhos próximos mais precisa do que informações sobre os vizinhos mais distantes, isto sem que seja necessário o aumento do tamanho das mensagens (SILVA et al., 2007).

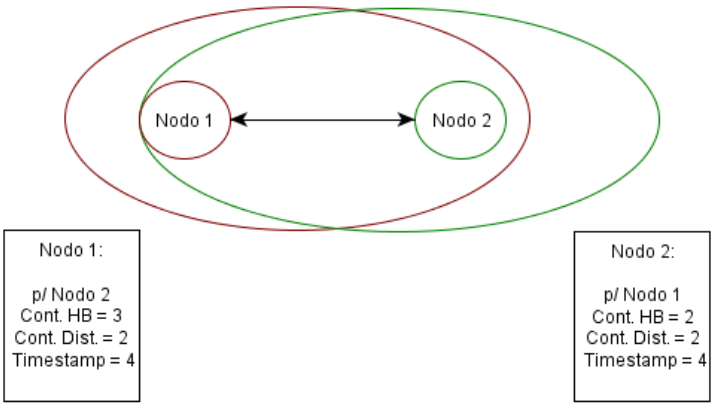

Figura 10 - Ilustração do modelo proposto por Hutle

Na Figura 10 é possível verificar uma ilustração do modelo proposto por Hutle, aonde cada nodo possui uma tabela com informações de *heartbeat*, distância e *timestamp* para cada outro nodo pertencente a rede.

2.5.3. Modelo Sridhar

O detector de defeitos proposto por Sridhar (SRIDHAR et al., 2006) possui características diferentes dos demais modelos apresentados anteriormente. Este modelo é voltado para detecção de defeitos em redes locais, que tolera a mobilidade dos nodos.

No detector proposto por Sridhar os nodos só mantém informações sobre seus vizinhos imediatos, reduzindo a quantidade de memória necessária para armazenar dados sobre os nodos vizinhos (BAGGIO, 2010).

O modelo de Sridhar possui a capacidade de identificar quando um nodo realmente falhou ou simplesmente migrou para a região de outro nodo. E pode ser implementado em duas etapas distintas (SRIDHAR et al., 2006; BAGGIO, 2010; SILVA et al., 2007):

- Etapa de detecção de falhas local: Nesta etapa deve ser implementado um detector de defeitos, que pode ser qualquer um dentre os já estudados, desde que o mesmo mantenha um *timestamp* para cada nodo na lista de suspeitos.
- Etapa de detecção de mobilidade: Esta etapa é executada por cada nodo da rede, que compartilha suas informações com outros nodos, a fim de verificar ou corrigir nodos suspeitos, ou seja, a partir das informações de suspeita de falhas levantadas pelos nodos vizinhos, um determinado nodo consegue determinar se um nodo específico ‘X’ esta realmente falho ou somente moveu-se para outra região.

Estas duas camadas satisfazem as especificações das redes móveis para forte abrangência local, precisão local eventualmente forte e localização suspeita.

2.6. Conclusões Parciais

Este capítulo abordou algoritmos detectores de defeitos para redes de sensores estáticas e móveis, assim como apresentou as limitações a detecção de defeitos em RSSF. Limitações quanto a consumo de energia, escalabilidade dos nodos e qualidade da rede sem fio são comuns para qualquer aplicação, além de problemas decorrentes da mobilidade dos nodos que podem ocasionar falsos alarmes na detecção de defeitos.

Apesar de seu grande crescimento na última década, as aplicações para redes de sensores sem fio se limitam ao *tradeoff* entre consumo de energia e capacidade de

processamento e monitoração. Limitações estas que acabam interferindo na qualidade da rede e escalabilidade dos nodos. As técnicas de detecção de defeitos tiveram de se adaptar as mudanças entre redes estáticas e móveis, e com este propósito, (HUTLE, 2004), (SRIDHAR et al., 2006) e (FRIEDMAN; TCHARNY, 2005) propuseram algumas das estratégias utilizadas para detecção de defeitos para redes sem fio.

Todos estes algoritmos de detecção de defeitos apresentam suas próprias propriedades e características, com focos distintos. No modelo proposto por (HUTLE, 2004), o detector de defeitos possui mais informações a respeito da distância de seus vizinhos, assim como dados de *timestamp* dos mesmos, possibilitando um sincronismo entre os vizinhos dos nodos.

O modelo de (SRIDHAR et al., 2006) é dividido em duas camadas, detecção local e camada de mobilidade, sua proposta permite que sejam usados detectores de defeitos de classe $\Diamond P$ para detecção local. Por fim, o detector proposto por (FRIEDMAN; TCHARNY, 2005) tem seu foco voltado para o problema da mobilidade dos nodos. Ao invés de levantar falsas suspeitas o detector permite a passagem um determinado número de *heartbeats* (γ) até levantar suspeita a nodos suspeitos.

A academia possui diversos trabalhos que comparam o desempenho destes e outros detectores, como em (LORENZI et al., 2007; BAGGIO, 2010), porém, não é foco deste trabalho interferir no desempenho do detector de defeitos e sim adapta-lo da melhor forma possível a uma arquitetura hibrida. Neste sentido o modelo Friedman foi escolhido como modelo base para a integração na topologia hibrida devido as suas características de atuação em redes com muitos nodos (alta escalabilidade) e por possuir pacotes de mensagens menores que os demais modelos citados, o que impacta diretamente nos benefícios de redução de mensagens enviadas e consumo de energia que buscamos com esta abordagem.

3. ALGORITMOS DE LOCALIZAÇÃO DE NODOS

Um algoritmo de localização de nodos para uma aplicação em rede de sensores sem fio pode ser descrito como sendo um algoritmo de calculo de coordenadas de uma posição em um sistema de coordenadas (OLIVEIRA, 2009). Ou seja, a partir de um ou mais pontos de referência, este algoritmo estima a posição dos outros nodos desta rede.

O sistema GPS pode ser utilizado para se obter a localização de nodos em uma rede móvel. Seu sistema de navegação se utiliza da triangulação de satélites para estabelecer posições, ou seja, os receptores GPS captam o sinal de pelo menos 4 satélites e utilizam essa triangulação para determinar a posição exata do nodo em um sistema de coordenadas 3D (PARKINSON; GILBERT, 1983) (GETTING, 1993). Os receptores GPS utilizam o método do tempo de chegada para estimar o quão longe estão do satélite, ou seja:

$$Distância = Velocidade * Tempo_{Chegada} \quad \textbf{(1)}$$

Partindo-se deste pressuposto é obvia a necessidade da utilização de relógios de alta exatidão, do contrário os erros de cálculo podem ser inaceitáveis (OLIVEIRA, 2009).

Se levarmos em consideração que o sistema GPS está disponível 24 horas por dia e não possui nenhum custo financeiro para a contratação do serviço, se poderá definir que o problema de localização esta resolvido, porém, mesmo como uma tecnologia atrativa em um primeiro momento, sua utilização em algumas aplicações podem possuir elevado custo financeiro na compra do equipamento em larga escala, sem falar no seu alto consumo de potência. Em aplicações que exijam locais subterrâneos ou que possuam muitas obstruções a medição por GPS não é garantida (OH-HEUM; HA-JOO, 2007).

Estas limitações levaram ao estudo e desenvolvido de metodologias alternativas para algoritmos de localização, baseados na distância entre dois nodos. Basicamente, estes métodos de localização necessitam que o sistema possua dois tipos diferentes de nodos na rede, os chamados ‘nodos de referência’ ou ‘âncoras’ e os ‘nodos desconhecidos’.

Os ‘nodos de referência’ ou ‘âncoras’ são nodos que conhecem suas coordenadas no espaço, seja por serem estáticos e possuírem no seu código estas informações, ou por serem móveis e possuírem um dispositivo que indique sua posição no espaço.

Diferentemente, os ‘nodos desconhecidos’ não conhecem sua localização na rede e dependem da informação repassada a eles para estimarem suas coordenadas no espaço. Estes

nodos podem ser móveis ou estáticos, dependendo da aplicação que se deseja. Os métodos de localização alternativos que não utilizam GPS podem ser divididos em 2 grupos: Métodos baseados na medição da distância (*ou Range-Based*) e Métodos baseado na conectividade (*ou Range-Free*)

3.1. *Range-Based* (Métodos Baseados na Medição da Distância)

Métodos que utilizem uma abordagem *Range-Based* utilizam o alcance (ou *range*) do sinal de rádio para medir a distância entre o nodo âncora e o nodo desconhecido, estes métodos por ser classificados em:

3.1.1. *Time of Arrival* - TOA (Tempo de Chegada)

O Método TOA (KAPLAN, 1996) utiliza o tempo de chegada do sinal de um nodo qualquer desconhecido até um outro nodo âncora, ou vice-versa. Ou seja, podemos definir que esta técnica leva em consideração o tempo que o sinal leva para ser transmitido de um nodo a outro, a Figura 11 ilustra o processo.

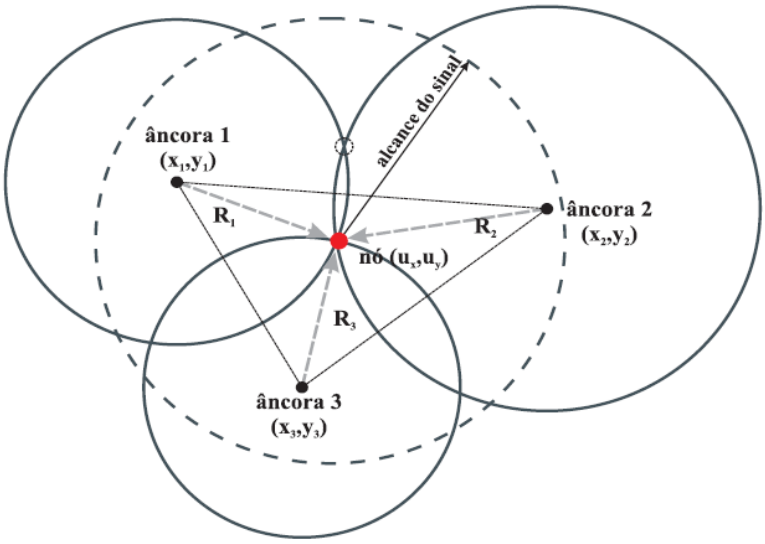

Figura 11 - Método Time Of Arrival de Localização (OLIVEIRA, 2009)

Como podemos observar acima, as circunferências em torno dos nodos pretos representam o alcance de abrangência de cada nodo âncora, aonde R_1, R_2 e R_3 são os raios, respectivamente. O ponto vermelho representa o nodo desconhecido e a linha tracejada o seu alcance.

Assim, a partir das informações de localização dos nodos âncoras o nodo desconhecido consegue identificar sua posição no espaço, quando mais nodos âncoras fizerem parte do processo mais preciso será a estimativa realizada pelo nodo desconhecido. Obviamente, como este método é baseado em informações de tempo, a sincronização entre todos os nodos do sistema deve ser muito precisa, do contrário poderão ser encontrados erros consideráveis.

3.1.2. *Time Diference of Arrival* - TDOA (Diferença no Tempo de Chegada)

O método por diferença no tempo de chegada, ou TDOA (KRIZMAN; BIEDKA; RAPPAPORT, 1997) (SPENCER, 2007) utiliza a correlação cruzada de um sinal que chega a dois nodos âncoras como um método indireto para calcular a distância e triangular a posição de um nodo desconhecido (OLIVEIRA, 2009).

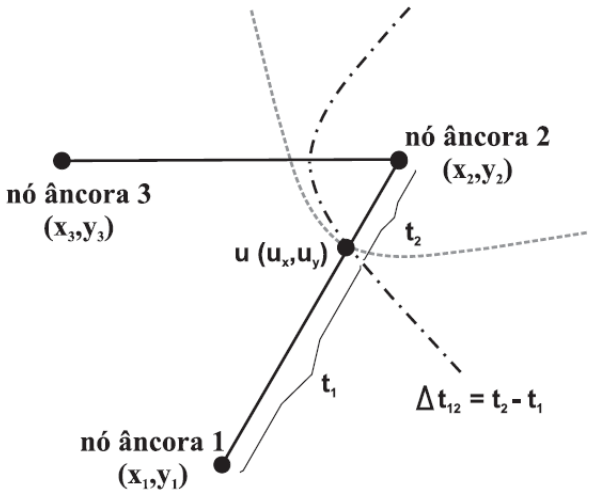

Figura 12 - Método Time Diference of Arrival de Localização (OLIVEIRA, 2009)

A Figura 12 ilustra dois pares de nodos âncoras $(1, 2)$ e $(2, 3)$, que são os focos das hipérboles. O nodo desconhecido μ esta localizado na intersecção entre as hipérboles, t_1 e t_2 são os tempos que os sinais levam para alcanças os nodos âncoras 1 e 2 a partir do nodo u.

Segundo (SPENCER, 2007) e (OLIVEIRA, 2009) “Cada medição TDOA remonta a uma curva hiperbólica ao longo da qual o nó desconhecido pode estar posicionado e, através da intersecção de duas hipérboles é possível detectar a localização do nó desconhecido. Na verdade para um espaço em duas dimensões (2D), caso nenhuma restrição de área for imposta

à rede, é possível encontrar duas soluções com apenas duas hipérboles, e uma terceira pode ser necessária para estabelecer o cálculo de forma mais precisa e encontrar o ponto único de interseção (localização do nó desconhecido) destas curvas".

3.1.3. *Angle of Arrival* - AOA (Ângulo de Chegada)

O método AOA (NICULESCU; NATH, 2003) requer a utilização de um equipamento especial para a medição das coordenadas dos nodos. Nesta topologia cada nodo sensor deve possuir um conjunto de antenas direcionais que irão habilitar a medição do ângulo de incidência dos sinais recebidos em relação ao seu próprio eixo.

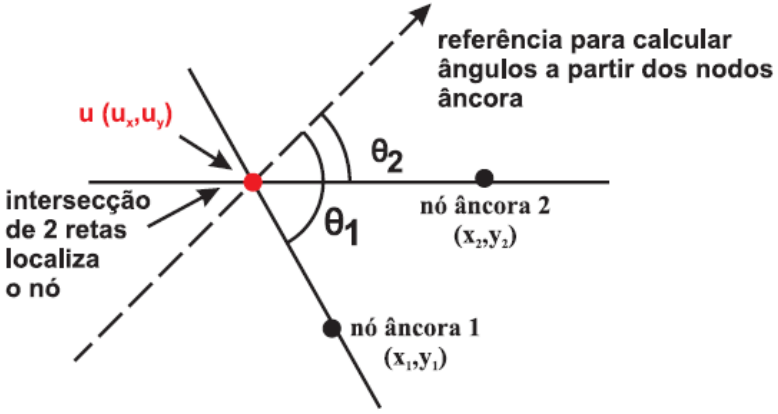

Figura 13 - Método Angle of Arrival de Localização (OLIVEIRA, 2009)

A Figura 13 detalha o modelo de localização por ângulo de chegada. Nesta topologia o nodo desconhecido possui sua posição estimada a partir dos ângulos medidos para os sinais que estão chegando. A seta pontilhada representa a orientação do nodo, os quais todos os outros ângulos são comparados. A localização do nodo a ser determinado é restrita ao longe de uma linha que inicia no nodo âncora, assim, através de no mínimo dois nodos âncoras, a localização do nodo desconhecido por ser calculada utilizando trigonometria básica. Este método pode apresentar erros de reflexão, difração e dispersão (OLIVEIRA, 2009).

3.1.4. *Received Signal Strength Indicator* – RSSI (Intensidade do Sinal Recebido)

O método de intensidade do sinal recebido ou RSSI (MONDINELLI, 2004) e (PATWARI et al., 2003), utiliza o nível de potência do sinal recebido para estimar a

localização do nodo desconhecido. Ou seja, o nodo desconhecido envia um sinal ao nodo âncora, através da força ou intensidade deste sinal o nodo âncora utiliza a equação 2 para estimar a posição do respectivo nodo.

$$P_t = P_r.k.r^{-2} \quad (2)$$

Aonde, P_t é a potência transmitida em Watts, P_r é a potência recebida em Watts, k é um fator relacionado as características das antenas, cumprimento de onda e meio transmissão, por fim r é a distância entre os nodos.

O presente método é considerado bastante preciso para distâncias pequenas (até 3 metros), porém pode ter bastantes problemas em redes que necessitem de algum tipo de interferência ou obstrução. Sua grande vantagem é de não necessitar de sincronização de tempo ou de algum hardware especial, algo impensável nos modelos anteriores.

3.2. Range-Free (Métodos Baseados na Conectividade)

O método *Range-Based* utiliza a distância para estimar a localização de nodos desconhecidos, diferentemente deste desta metodologia o modelo *Range-Free* se utiliza da conectividade entre os nodos para estimar sua localização, ou seja, ou conteúdo das mensagens trocadas entre os nodos.

Existem duas técnicas baseados em *Range-Free:*

- Técnicas locais – que requer uma quantidade bastante elevada de nodos âncoras na rede, aonde cada nodo desconhecido possui em seu alcance de sinal diversos nodos âncoras, como exemplos de algoritmos que usam esta técnica temos o MCL e o Centroid.

- Técnica de contagem de saltos (*ou hop counting*) – nesta técnica as mensagens enviadas pelos nodos de referência se propagam pela rede através da contagem de saltos (ou *hops*). Este processo também é chamado de "*flooding of the network".*

No algoritmo de localização CAB (do inglês, *Concentric Anchors-Beacons*) proposto por (VIVEKANANDAN, 2006), cada nodo de referência possui um sistema GPS que calcula sua localização instantaneamente no espaço. Os nodos de referências enviam sinais chamados de *beacons*, em dois níveis diferentes de potência, estes dados são coletados pelo nodo desconhecido que deve ser cercado por no mínimo 3 nodos âncoras, apesar de utilizar somente 2 para calcular os pontos de intersecção válidos, a Figura 14 ilustra o método CAB.

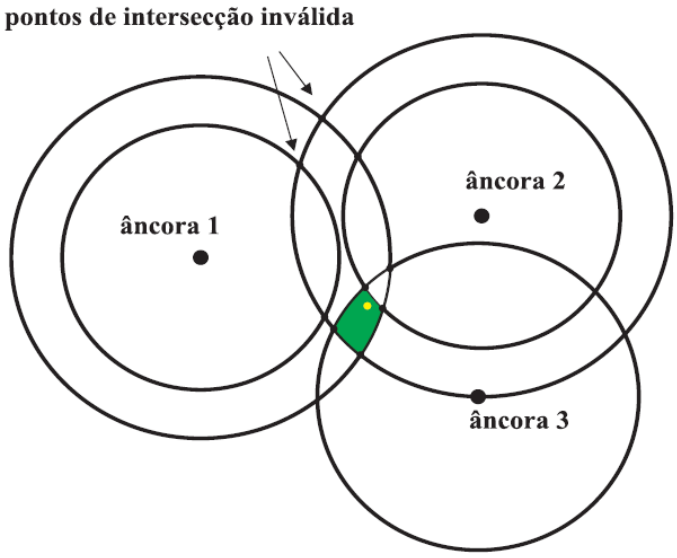

Figura 14 - Método CAB de Localização (VIVEKANANDAN, 2006)

Na Figura 14, o espaço em verde representa o intervalo entre os pontos de intersecção válidos e o ponto amarelo representa o nodo desconhecido.

Dentre as técnicas de contagem de saltos, uma bastante conhecida é a DV-HOP proposta por (NICULESCU; NATH, 2001). Nesta técnica os nodos âncoras sabem suas localizações e descobrem as localizações dos nodos desconhecidos através da contagem de saltos (*hops*) das mensagens que enviam até os nodos desconhecidos, ou seja, através do número de saltos que uma mensagem leva de um nodo até outro, o âncora consegue converter esta contagem para um valor em metros, estimando assim a distância do nodo desconhecido até ele, e assim sua localização no espaço. A Figura 15 ilustra o modelo DV-HOP de localização proposto por (NICULESCU; NATH, 2001).

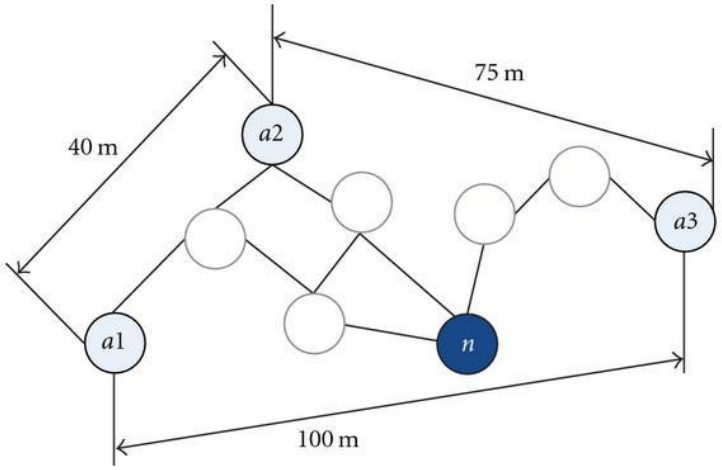

Figura 15 - Modelo de Localização DV-HOP (NICULESCU; NATH, 2001)

Assim como o modelo CAB e o DV-HOP, temos na literatura diversos modelos de algoritmos de localização, cada um com foco na sua própria topologia de rede. Como não é foco deste trabalho detalhar os algoritmos de localização de nodos e sim introduzi-los para que o leitor conheça os métodos utilizados na topologia proposta de integração entre algoritmos de localização e detectores de defeitos, o único algoritmo a ser detalhado serão os métodos Centroid e Monte Carlos, que são modelos *Range-Free* que utilizam técnicas locais.

3.2.1. Algoritmo Centroid de Localização

O algoritmo Centroid foi proposto por (BULUSU; HEIDEMANN; ESTRIN, 2000), como um modelo que se baseia na conectividade entre os nodos, e utiliza técnicas locais para determinar a localização de nodos desconhecidos. Ou seja, o presente método utiliza o conteúdo das mensagens trocadas em uma rede aonde é necessária uma grande quantidade de nodos âncoras para a estimativa da posição de nodos desconhecidos.

Ainda sobre o modelo Centroid, supõe-se que os nodos âncoras são estáticos e que seu raio de abrangência consiga cobrir toda a área dos nodos, estes nodos de referência ainda devem enviar periodicamente mensagens contendo sua posição aos nodos desconhecidos de seu alcance. Uma característica deste algoritmo é o de não necessitar manter uma tabela prévia com as coordenadas dos nodos âncoras, o que aumenta a escalabilidade da rede.

É importante salientar que cada nodo âncora deve enviar um número "S" de mensagens em um determinado espaço de tempo "t", ao término deste período de coleta de dados um processo de estimativa de localização, com os dados armazenados, é iniciado.

Diferentemente dos nodos de referência, os nodos desconhecidos são móveis e possuem movimentação aleatória na rede, periodicamente recebem as informações da localização dos nodos âncoras e a partir da intersecção das regiões de conectividade cobertas pelos nodos de referência (ou âncoras), conseguem estimar suas coordenadas na rede.

Esta intersecção de regiões de conectividade recebe o nome de centro ou **Centroid** dos nodos âncoras participantes, daí o nome do algoritmo (OLIVEIRA, 2009). Tais nodos de referência são escolhidos para participar deste calculo quando o resultado da métrica de conectividade proposta por (BULUSU; HEIDEMANN; ESTRIN, 2000) ultrapassar o valor de 90%. Pode-se observar a 3 para calculo destas métricas.

$$CM_i = \frac{N_{recv}(i,t)}{N_{sent}(i,t)} * 100 \qquad \textbf{(3)}$$

Aonde, CM_i é o resultado da equação para um determinado nodo âncora "i", $N_{recv}(i,t)$ é o número de informações de localização que foram enviadas por este nodo "i" em um espaço de tempo "t", e $N_{sent}(i,t)$ é o número de informações de localização que foram enviadas pelo nodo "i" e recebidas no período de tempo "t".

Logo, todos os nodos âncoras que obtiverem um valor CM_i maior ou igual a um valor pré-determinado CM_{thresh}, irão participar do calculo final de localização daquele nodo desconhecido em específico. A equação para o calculo final de estimativa de localização de um nodo "i" pode ser observada em 4.

$$(X_{est}, Y_{est}) = \left(\frac{XAi_1 + XAi_2 + ... + XAi_k}{k}, \frac{YAi_1 + YAi_2 + ... + YAi_k}{k} \right) \qquad \textbf{(4)}$$

Aonde, (X_{est}, Y_{est}) são as coordenadas do nodo a serem descobertas e k é o número total de nodos âncoras que irão participar do calculo final.

Outra característica interessante do algoritmo Centroid é o de operar com o cálculo do centro de regiões geométricas retangulares, diferentemente de outros métodos, a grande vantagem desta característica é a simplificação matemática que esta resolução trás ao método, deixando assim o algoritmo mais rápido e simples. Porém, a grande desvantagem deste ponto é o de o Centroid se tornar dependente da sobreposição do alcance de rádio dos nodos

âncoras. Assim fica óbvio que neste algoritmo a deposição dos nodos de referência na rede, tem papel fundamental para o melhor rendimento e exatidão do modelo.

Mesmo com a limitação descrita acima, a baixa complexidade computacional do Centroid permite a ele ser um dos algoritmos mais rápidos na estimativa de localização de nodos.

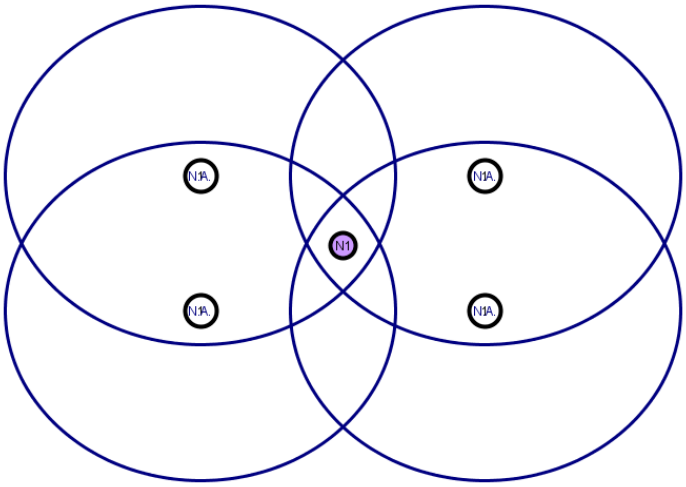

Figura 16 - Modelo Centroid de Localização

A Figura 16 ilustra o modelo Centroid de localização, aonde "NA" representa os nodos âncoras ou de referência, a circunferência em volta de cada nodo âncora representa sua região de abrangência e, por fim, "$N1$" representa o nodo que se deseja localizar as coordenadas.

O modelo proposto por (BULUSU; HEIDEMANN; ESTRIN, 2000) possui características determinantes que o tornam até hoje parâmetro para diversas aplicações e melhorias, dentre elas podemos citar:

Os nodos desconhecidos não necessitam enviar mensagens, eles apenas recebem mensagens dos nodos âncoras e com base nelas, estimam sua localização, economizando assim a energia da transmissão de mensagens e eliminando o tempo que seria gasto na troca de mensagens entre os nodos moveis.

No modelo Centroid só há ocorrência de envio de mensagens pequenas, contendo a identificação do nodo âncora e suas coordenadas (X_i, Y_i). O Centroid não utiliza equações complexas ou estatísticas, esta baixa complexidade diminui consideravelmente o tempo de processamento dos dados.

3.2.2. Algoritmo MCL (Monte Carlo Localization)

O algoritmo de localização MCL (ou Monte Carlo Localization) para redes de sensores móveis é uma adaptação do método Monte Carlo desenvolvido para utilização em robôs (DELLAERT et al., 1999) (THRUN et al, 2001). O presente método é um filtro de partículas combinado com modelos probabilísticos de percepção e movimentação do robô. A ideia central por trás deste algoritmo é o de conseguir representar possíveis localizações posteriores a atual utilizando um conjunto de amostras.

Assim, cada etapa é dividida entre as fases de predição e atualização, na primeira o robô realiza um movimento aleatório e não tem certeza quando a sua localização, na fase de atualização novas observações são acrescentadas ao filtro e os dados são atualizados, este processo se repete e o robô fica continuamente atualizando suas coordenadas de localização.

Obviamente, existem diferenças significativas na aplicação do método de localização MCL para robôs e redes de sensores, por exemplo, enquanto o primeiro se utiliza de um mapa pré-definido, os nodos de uma rede sem fio se movimentam aleatoriamente em um espaço livre e desconhecido. Assim como, os robôs possuem controle e conhecimento probabilísticos da sua movimentação no mapa pré-definido, enquanto os nodos possuem pouco ou nenhum controle da sua mobilidade, além de desconhecer sua velocidade e direção. Por outro lado uma aplicação em redes de sensores nos permite que as grandes quantidades de nodos possam interagir e cooperar entre eles para a troca de informações de localização.

A grande vantagem da adaptação do método MCL para redes de sensores é a de que, diferentemente da maioria dos outros algoritmos, o método se aproveita da mobilidade dos nodos da rede para otimizar a precisão de localização. Ou seja, o que é um problema para os demais algoritmos acaba se tornando peça fundamental para que o método MCL consiga o melhor rendimento. Esta característica torna o MCL ideal para aplicações em que haja muita movimentação dos nodos, como por exemplo, rastreamento de animais e veículos.

O algoritmo de localização de Monte Carlo para redes de sensores móveis segue o mesmo padrão do utilizado com robôs, com a diferença de que os nodos podem cooperar entre si para trocar informações de localização, desta forma (HU; EVANS, 2004) optou por também segmentar o modelo entre as etapas de Predição e Filtragem. Segundo as definições de (HU; EVANS, 2004), inicialmente o nodo não possui conhecimento da sua localização e a variável N representa o número de amostras mantidas em cada intervalo de tempo. A cada

nova etapa um conjuntos de novas possíveis localizações L_t são calculadas com base no conjunto de coordenadas anterior L_{t-1}, na possível localização definida a partir da etapa anterior e em novas observações o_t. Esta descrição pode ser observada com maiores detalhes nos equacionamentos da Figura 17.

L_0 = {conjunto de N localizações aleatórias na área em questão}

L_t = { }

while (size L_t < N) **do**

R = { $l_t^i \mid l_t^i$ is selected from p($l_t \mid l_{t-1}^i$), $l_{t-1}^i \in L_{t-1}$ for all $1 \leq i \leq$N} **Predição**

$R_{filtered}$ = { $l_t^i \mid l_t^i$ where $l_t^i \in R$ and $p(o_t \mid l_t^i) > 0$ } **Filtragem**

L_t = choose($L_1 \cup R_{filtered}$, N)

Figura 17 - Algoritmo do modelo de localização MCL (HU; EVANS, 2004)

Predição: Na etapa de predição o nodo inicia conhecendo um conjunto de possibilidades de localização computadas anteriormente "L_{t-1}" e que aplicadas ao modelo de mobilidade de cada amostra resultará em um novo conjunto de possíveis localizações "L_t". Assume-se que o nodo não conhece sua velocidade ou direção de deslocamento, a única informação que possui é de que sua velocidade estará entre 0 e o velocidade máxima "v_{max}". Então, se na etapa anterior "l_{t-1}" temos a possível posição anterior do nodo, as possíveis posições do estado atual estão contidas em uma região circular com origem em "l_{t-1}^i" e de raio "v_{max}". Ainda segundo (HU; EVANS, 2004), foi utilizado $d(l_t, l_{t-1})$ para representar a distância Euclidiana entre dois pontos l_1 e l_2. Se a velocidade dos nodos é distribuída uniformemente no intervalo [0, v_{max}], a probabilidade da localização do nodo é dado pela distribuição uniforme.

$$p(l_t \mid l_{t-1}) = \begin{cases} \dfrac{1}{\pi v_{\max}^{\,2}}, \text{ se } d(l_t, l_{t-1}) < v_{\max} \\ 0, \text{ se } d(l_t, l_{t-1}) \geq v_{\max} \end{cases} \quad (5)$$

Com isso podemos observar que o conjunto R calculado na fase de predição contém uma estimativa de localização selecionada aleatoriamente a partir do circulo de raio v_{max} em torno de cada ponto L_{t-1}. Isto reflete a incerteza sobre a localização do nodo devido a sua movimentação aleatória.

Filtragem: Na fase de filtragem o nodo filtra as localizações impossíveis baseando-se em novas observações de âncoras (ou sementes) e/ou vizinhos. Nodos sementes ou âncoras são aqueles que conhecem suas coordenadas no tempo, seja por possuírem GPS ou outro equipamento que possibilite estas informações. Assim, a cada intervalo de tempo, todos os nodos que estiverem dentro do alcance do radio de uma semente vão receber o anúncio de localização (coordenadas) daquele nodo. Basicamente existem 4 tipos de nodos âncoras, são eles:

- Outsiders – nodos âncoras que não foram ouvidos tanto na etapa atual como na anterior.
- Arrivers – nodos âncoras que são ouvidos no tempo atual, porém não foram ouvidos na etapa anterior.
- Leavers – nodos âncoras que foram ouvidos na etapa anterior, mas não na etapa atual.
- Insiders – nodos âncoras que foram ouvidos nas duas etapas.

Os âncoras dos tipos *arrivers* e *leavers* fornecem a informação mais útil, visto que o nodo vai saber que estava dentro do alcance ‘r’ de l_0 no tempo t_0, mas não do alcance r de l_1 no tempo t_1.

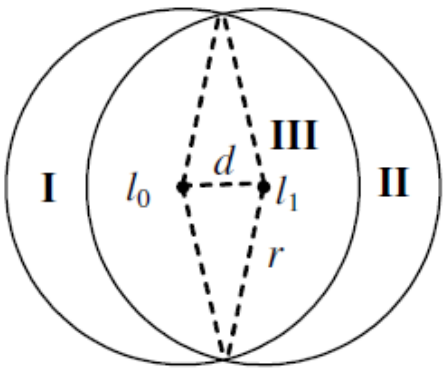

Figura 18 - Movimentação dos âncoras

A Figura 18 ilustra a movimentação dos nodos sementes, na imagem podemos observar que o semente se move de l_0 no tempo 0, para l_1 no tempo 1. Ele será classificado como *insider* para nodos na região III, *arriver* para nodos na região II, *leaver* para nodos na região I e *outsider* para todos os demais nodos.

Mesmo assim, se um nodo se utilizar apenas da informação provinda diretamente de um âncora, o nodo não conseguirá saber a localização anterior de um *arriver* ou a atual de um *leaver*. Para resolver este problema podem-se utilizar duas possibilidades:

O nodo âncora (S) envia para os nodos de seu alcance tanto a suas coordenadas atuais como anteriores em cada anúncio de localização.

S → Region $\quad HELLO \mid ID_S \mid loc_t \mid loc_{t-1}$

Os nodos vizinhos (N) também transmitem informações sobre a localização dos âncoras que conseguem ouvir.

S → Region $\quad HELLO \mid ID_S \mid loc_t$

N → Region $\quad HELLO \mid ID_N \mid \{ID_S, loc_{S_t}\}$

O envio de mensagens por nodos vizinhos, apesar de possuir um custo mais elevado que o primeiro método possibilita aos nodos obterem informações de âncoras que estão fora de sua região de abrangência, a até uma distância de duas vezes o seu raio ($2.r$). A Figura 19 ilustra a abrangência obtida pelos âncoras utilizando-se os dois métodos descritos anteriormente e a condição do filtro para âncoras *insiders* e *outsiders.*

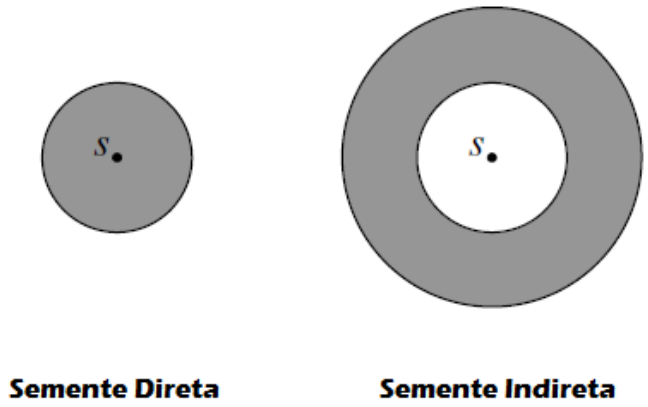

Figura 19 - Abrangência dos âncoras

Como nos mostra a Figura 19, se um nodo estiver dentro da região de abrangência (r), o semente será ouvido diretamente. Caso o nodo esteja a uma distancia entre [r, 2r] o semente será ouvido indiretamente, ou seja, ouvido por um nodo vizinho. A condição de filtragem para âncoras *insiders* e *outsiders* pode ser dada pela equação 6.

$$filter(l) = \forall_S \in S, d(l,s) \leq r \wedge \forall_S \in T, r \leq d(l,s) \leq 2r \quad \textbf{(6)}$$

Aonde 'S' representa o conjunto de todos os nodos âncoras ouvidos por 'N', e 'T' representa o conjunto de todos os âncoras ouvidos somente pelos vizinhos de 'N'.

A distribuição de probabilidade da posição atual do nodo com base nas observações (o_t), representada por $p(l_t \mid o_t)$, será 0 se a condição do filtro destas observações for falsa. Assim, pode-se eliminar coordenadas inconsistentes com as observações do conjunto de localizações possíveis, podendo existir até mesmo menos de N possíveis localizações pós-filtragem. As etapas de predição e filtragem se repetem até que haja pelo menos N possíveis localizações.

3.3. Conclusões Parciais

O capítulo 3 apresentou o conceito básico de algoritmos de localização, assim como os principais algoritmos utilizados.

Tais algoritmos foram classificados quanto ao seu método de funcionamento, seja baseado na medição da distância entre dois ou mais nodos (*range-based*), como também na conectividade entre eles (*range-free*). Os métodos baseados na conectividade se utilizam do

conteúdo das mensagens trocadas entre os nodos para estimar suas coordenadas de localização. Dentre eles podem-se citar os métodos baseados na contagem de saltos (hops) e também os que se utilizam de técnicas locais, como é o caso do Centroid e do Monte Carlo Localization, algoritmo escolhido para ser utilizado na topologia desenvolvida neste trabalho.

Como o propósito desta dissertação é o trabalho conjunto de dois algoritmos com objetivos distintos e que nunca colaboraram entre si em um ambiente totalmente móvel e aleatório, optou-se pela escolha do MCL por possuir um elevado número de troca de mensagens entre os nodos com seus âncoras e vizinhos, assim como por ser um algoritmo que toma vantagem da mobilidade dos nodos para melhorar sua precisão de localização, ideal para aplicações que necessitem de grande movimentação dos nodos como por exemplo o monitoramento de animais.

4. TOPOLOGIA HIBRIDA

A Topologia Hibrida proposta, integra o algoritmo de localização MCL e o detector de defeitos Friedman, com o objetivo de otimizar o conteúdo das mensagens enviadas pelos nodos, diminuindo assim a sua incidência e otimizando o desempenho da rede. É importante ressaltar novamente que não é objetivo deste trabalho interferir na precisão e no desempenho de ambos os algoritmos separadamente, apesar do algoritmo hibrido proporcionar esta possibilidade em trabalhos futuros, a integração proposta tem seu foco na melhoria de desempenho geral da rede através da diminuição na troca de mensagens.

A escolha pelos algoritmos de localização MCL e Friedman deveu-se ao fato de ambos serem propícios a integração em um modelo hibrido. O MCL, por ser um algoritmo R*ange-Free* utiliza o conteúdo das mensagens enviadas para determinar a localização dos nodos, assim como possui uma baixa densidade de nodos âncoras e sua arquitetura se aproveita da mobilidade dos nodos para melhorar sua precisão, o que é ideal para redes com grande movimentação dos nodos.

O detector de defeitos Friedman, por sua vez, deverá ser implementado dentro do algoritmo de localização, para isto o detector deve conseguir trabalhar em redes com um grande número de nodos sem que haja perda de desempenho. O modelo proposto por Friedman vai ao encontro a esta característica, aumentando o desempenho do detector a medida que a conectividade dos nodos também aumenta. O tamanho reduzido das mensagens enviadas pelo detector de defeitos Friedman também pode ser considerada outra vantagem, já que tem impacto direto no consumo de energia utilizado durante a transmissão de mensagens.

Por fim, buscou-se que ambos os algoritmos implementados na Topologia Hibrida conseguissem ser executados no ambiente mais desconhecido e aleatório possível em uma rede de sensores móvel, mantendo as suas propriedades, além de otimizar o desempenho da rede através da redução do número de mensagens enviadas e consequente diminuição do consumo de energia.

4.1. Breve revisão dos algoritmos MCL e Friedman

Neste item será realizada uma breve revisão dos algoritmos MCL e Friedman que basearam a topologia mista a ser detalhada posteriormente. Para um maior detalhamento de ambos deve-se observar os capítulos anteriores.

4.1.1. Detector de Defeitos Friedman:

O Detector de Defeitos para redes móveis proposto por Friedman (FRIEDMAN; TCHARNY, 2005) é uma variação do modelo Gossip para redes estáticas. Assim como no modelo Gossip (RENESSE, 1998) todos os nodos da rede são monitores e monitorados, ou seja, não existe uma hierarquia entre os nodos. Cada um destes nodos possui uma tabela contendo um contador de *heartbeat* para cada um dos demais nodos da rede com que teve alguma interação.

A cada intervalo de tempo "*t*", os nodos da rede enviam sua lista de endereços atualizada para um ou mais nodos de sua região de abrangência. Cada nodo receptor irá comparar a tabela que recebeu com a sua própria, atualizando sua lista de endereços e mantendo os valores de *heartbeats* mais altos. Caso o valor de *heartbeat* recebido para um determinado nodo seja maior que o da lista local, um novo tempo de suspeita (*timeout*) é calculado para o nodo em questão.

Um nodo será considerado suspeito se o seu contador de *heartbeats* não for incrementado até a passagem do seu tempo de suspeita.Vide seção 2.5.1 para mais detalhes.

4.1.2. Algoritmo de Localização MCL:

O algoritmo Monte Carlo de Localização utiliza o conteúdo das mensagens trocadas entre os nodos para estimar a localização de nodos desconhecidos, em uma rede aonde é necessária uma quantidade mínima de nodos sementes (ou âncoras) que conhecem previamente as suas coordenadas no espaço, seja por possuírem um GPS ou outro equipamento para este fim.

Esta técnica de estimativa de localização se divide em duas etapas: predição e filtragem. Na etapa de predição o nodo desconhecido prediz sua posição atual baseando-se nas suas possíveis localizações prévias. Na etapa de filtragem o nodo recebe mensagens de observação de nodos âncoras e vizinhos e atualiza os pesos de seu conjunto de amostras. Por fim as amostras válidas (de peso 1) são utilizadas pelo algoritmo para calcular a posição posterior do nodo. Mais detalhes sobre o funcionamento do Algoritmo de Localização Monte Carlo são descritas na seção 3.2.2.

4.2. Arquitetura do Algoritmo de Topologia Hibrida

O algoritmo de topologia hibrida busca a integração entre algoritmo de localização e detector de defeitos em redes de sensores móveis. Porém antes de se detalhar o funcionamento da topologia proposta neste trabalho é necessário que se defina o ambiente em que o método foi desenvolvido.

Mesmo para aplicações em redes de sensores móveis é possível que se encontre três tipos diferentes de cenários (HU; EVANS, 2004), são eles:

- Cenário com nodos estáticos e âncoras móveis – Pode ser utilizado em aplicações aonde os nodos de medição possam ficar parados enquanto os nodos âncoras se movem. Este cenário é mais usualmente utilizado em aplicações militares.
- Cenário com nodos móveis e âncoras estáticos – Neste cenário os nodos desconhecidos se movimentam enquanto os âncoras permanecem estáticos, um exemplo de aplicação para este fim é o monitoramente de peixes em uma determinada lagoa ou rio, enquanto os nodos de monitoramento estão sempre móveis presos em cada peixe, os nodos âncoras permanecem estáticos nas margens dos rios.
- Cenário com nodos e âncoras móveis – O cenário com ambos os nodos, desconhecidos e âncoras se movimentando representa uma aplicação menos controlada que as anteriores. De forma geral, as aplicações ad hoc (nodos independentes) são mais comuns em redes de sensores móveis, neste caso é necessário que os nodos âncoras possuam um sistema de GPS ou outro equipamento que lhe informe suas coordenadas.

Para a análise da topologia hibrida buscou-se considerar as características individuais dos dois algoritmos, de forma que eles não perdessem suas propriedades individuais ao mesmo tempo em que sua aplicação era realizada no cenário mais desconhecido possível. Alguns itens que podemos levantar quanto ao ambiente escolhido para a implantação da arquitetura são:

- A rede é dividida em nodos desconhecidos e nodos âncoras (ou sementes);

- Não existe hierarquia na rede, ou seja, todos os nodos são monitores e monitorados;
- Os nodos desconhecidos não dispõem de nenhum hardware adicional;
- Os nodos âncoras possuem um sistema GPS para obtenção de suas coordenadas;
- A implantação dos nodos âncoras é desconhecida;
- A distribuição dos nodos é irregular;
- Nodos desconhecidos e âncoras se movimentam aleatoriamente, sem controle sobre velocidade ou direção;

Os parâmetros citados descrevem que o ambiente utilizado para a implementação da topologia hibrida é totalmente móvel e de mobilidade descontrolada e aleatória, ao mesmo tempo em que se mantiveram as propriedades individuais dos modelos de localização e detecção de defeitos.

Em aplicações específicas para detectores de defeitos uma rede pode ou não ter hierarquia, tendo ou não nodos centrais que realizam o consenso através da informação que chega até eles. No ambiente simulado para o algoritmo hibrido buscou-se um ambiente aonde todos os nodos fossem tanto monitores quanto monitorados e dependessem da troca de informações com outros nodos para detectar possíveis falhas de seus vizinhos, ou seja, um ambiente sem hierarquia de nodos. Nota-se que esta característica mantém as propriedades do modelo proposto por Friedman e outras arquiteturas baseadas em Gossip, indo ainda ao encontro às propriedades de mobilidade dos algoritmos de localização.

Os algoritmos de localização necessitam obrigatoriamente que a rede possua nodos âncoras (ou sementes), que conhecerão suas coordenadas no espaço, seja por possuírem um GPS ou qualquer outro dispositivo para este fim. Os demais nodos da rede, também chamados por nodos desconhecidos, não possuem qualquer hardware adicional e dependem da informação repassada a eles pelos âncoras para calcular sua posição. É importante salientar que qualquer algoritmo de localização necessita de informações de coordenadas de nodos âncoras para estimar sua localização no espaço, se o nodo não obtiver nenhuma informação de localização da rede não conseguirá mapear as suas coordenadas no meio.

Com isso é possível observar que todas estas características descritas para a topologia hibrida seguem as propriedades dos algoritmos MCL e Friedman, obviamente ambas as

arquiteturas foram escolhidas por apresentarem ambientes de implantação compatíveis a uma rede ad hoc.

O algoritmo de localização de Monte Carlo, ou MCL, como foi exaustivamente descrito neste trabalho, faz uso do conteúdo das mensagens trocadas entre os nodos para determinar sua localização. A escolha por uma integração utilizando o algoritmo MCL se deveu ao fato da arquitetura se utilizar da mobilidade dos nodos para aumentar sua precisão de localização, o que é ideal para aplicações aonde seja necessária uma grande movimentação por parte dos nodos como monitoramento de peixes e animais silvestres. Outra característica interessante deste algoritmo é o de possuir uma baixa densidade de nodos âncoras o que diminui os seus custos de implantação em relação a outros modelos.

Como a ideia base desta integração é a implementação do detector de defeitos dentro do algoritmo de localização, é necessário que o primeiro consiga trabalhar em redes relativamente grandes e que possuam alto número de trocas de mensagens sem que haja perda de desempenho, como é o caso do modelo proposto pro Friedman aonde quanto maior a rede e a conectividade dos nodos, maior o desempenho do algoritmo. Uma vantagem a ser considerada do detector de defeitos Friedman em comparação aos demais é a utilização de mensagens menores, o que não ocasiona pacotes de maior tamanho e um consumo de potência mais elevado durante a transmissão de mensagens.

A partir da descrição do ambiente de implementação da topologia hibrida, assim como das propriedades e vantagens dos algoritmos de localização e detecção de defeitos utilizados, pode-se definir a arquitetura do algoritmo proposto.

Na arquitetura proposta o detector de defeitos irá aproveitar as trocas de mensagens realizadas pelo algoritmo de localização para realizar sua verificação de falhas. A Topologia Hibrida possui três estágios: predição, filtragem e detecção. Na etapa de predição o nodo não troca mensagens com os nodos vizinhos, ele somente prediz sua posição atual a partir das possíveis posições anteriores. Apesar de possuir movimentação aleatória e não conhecer sua velocidade ou direção, o nodo conhece suas velocidades mínima e máxima [0, $v_{máx}$]. Logo, a partir destes dados é possível que o nodo afirme que partindo de uma posição anterior l_1, ele poderá no máximo atingir uma posição dentro de uma circunferência de raio $v_{máx}$ com origem na posição l_1.

A Figura 20 ilustra o raio que determina as possíveis localizações do nodo no processo de predição. Apesar de não conhecer sua localização futura, velocidade ou direção, o nodo

sabe que partindo de sua posição anterior não poderá ultrapassar o raio de abrangência de sua velocidade máxima, assim sua possível posição atual deve estar contida dentro deste raio.

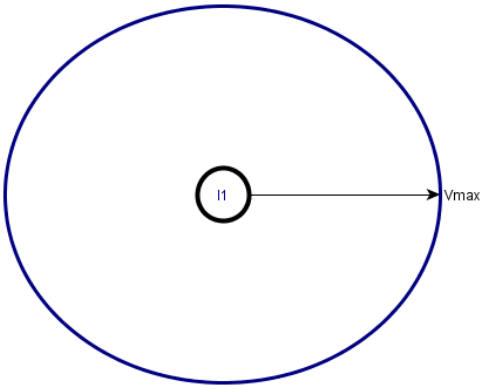

Figura 20 - Etapa de predição

Com isto, um conjunto de amostras de estimativa de posição é selecionado aleatoriamente de dentro do circulo com raio $v_{máx}$ e origem na posição l_1. A partir deste conjunto de amostras e da incerteza quanto a real posição do nodo, começa a segunda etapa do algoritmo hibrido.

A segunda etapa da topologia híbrida, também conhecida por etapa de filtragem ou atualização, toma como base as informações provindas da etapa anterior para que o nodo, a partir da troca de mensagens com nodos âncoras e vizinhos, filtre posições impossíveis e consiga definir a sua provável posição posterior. É durante a etapa de filtragem que os nodos trocam mensagem com seus âncoras e vizinhos, desta forma as informações relativas a detecção de defeitos (lista de *heartbeat*) de cada nodo também serão enviada nesta etapa, porém sua análise quanto verificação de defeitos só será executada na última etapa da topologia hibrida.

Durante o algoritmo de localização MCL tradicional os nodos recebem dos âncoras (S) e vizinhos (N) as seguintes mensagens:

S → Region $\quad HELLO \mid ID_S \mid loc_t$

N → Region $\quad HELLO \mid ID_N \mid \{ID_S, loc_{S_t}\}$

Observe que o nodo semente (ou âncora) envia uma mensagem para todos os demais nodos de sua região de abrangência (S → Region) contendo seu identificador (ID_S) e suas

coordenadas de localização (loc_t), enquanto os vizinhos enviam além de seus dados de identificação (ID_N), informações a respeito da identificação e posição de todos os nodos âncoras presentes em sua região de abrangência ($\{ID_S, loc_{S_t}\}$). As Tabelas 2 e 3 ilustram o formato resumido dos pacotes de dados enviados por âncoras e nodos vizinhos no algoritmo MCL.

Tabela 2 - Formato do pacote de dados enviado pelos nodos Sementes (Âncoras) no algoritmo MCL

ID	Coordenadas de localização (x,y)

Tabela 3 - Formato do pacote de dados enviado pelos nodos vizinhos no algoritmo MCL

ID	
Nodo Semente 1	Coordenadas do Nodo Semente 1
Nodo Semente 2	Coordenadas do Nodo Semente 2
Nodo Semente n	Coordenadas do Nodo Semente n

Salienta-se que no pacote do nodo semente todos os dados são referentes somente ao próprio nodo, ou seja, identificação e coordenadas do próprio âncora. Diferentemente, os pacotes enviados pelos nodos vizinhos, não possuem as suas coordenadas, pois os mesmos as desconhecem e sim as identificações e coordenadas dos nodos âncoras que se encontrão dentro de sua região de abrangência. O único dado enviado pelo nodo vizinho que faz referência a ele próprio é o seu ID.

A Figura 21 ilustra a troca de mensagens que é feita pelo MCL durante a fase de filtragem.

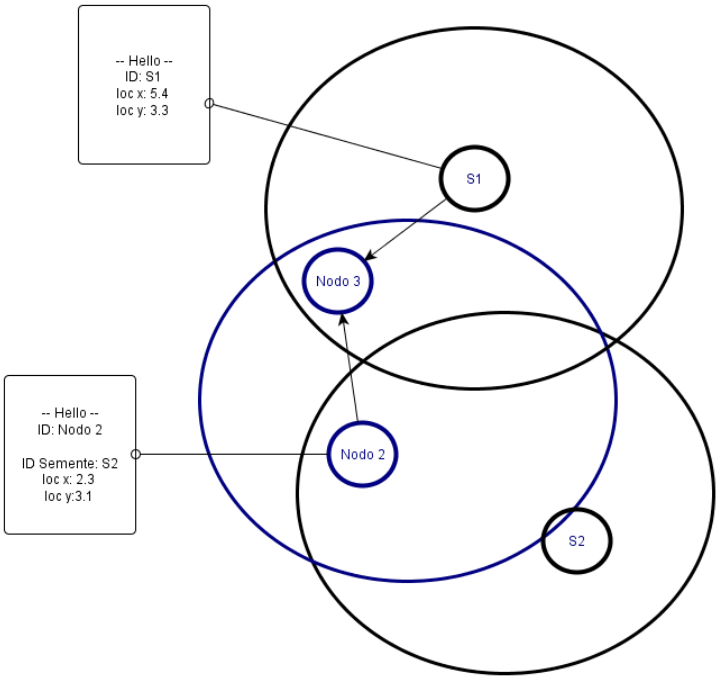

Figura 21 - Conteúdo das mensagens enviadas na etapa de filtragem do algoritmo MCL

Como podemos observar na figura acima, a cada turno (ou *round*) os nodos âncoras enviam sua identificação 'ID' e coordenadas (loc_x, locy) aos nodos desconhecidos de seu alcance. Note na figura que o nodo semente S1 envia para o nodo 3 suas informações. Da mesma maneira o nodo 2 vizinho do nodo 3 envia seus dados de identificação e as informações que foram repassadas a ele pelos nodos âncoras de sua região de abrangência.

Durante a fase de filtragem o algoritmo hibrido irá anexar a estas mensagens transmitidas segundo o modelo MCL, as informações referentes a detecção de defeitos.

No algoritmo detector de defeitos os nodos enviam sua tabela de *heartbeat* a cada intervalo de tempo para todos os nodos vizinhos pertencentes a sua região de abrangência. Em detectores baseados no modelo Gossip para redes móveis, os nodos não possuem hierarquia e todos são monitores e monitorados. Assim as mensagens de todos os nodos pertencentes a rede tem a mesma forma padrão.

$$\mathrm{N} \rightarrow \mathrm{Region} \qquad HELLO \mid ID_N \mid Hb_N$$

$$Hb_N = \{hc_1, hc_2, hc_3, ..., hc_n\}$$

Logo, no algoritmo detector de defeitos Friedman, os nodos enviam a seus vizinhos uma mensagem contendo sua identificação (ID_N) e sua tabela de *heartbeats* (Hb_N), que

contém os contadores de *heartbeats* de cada nodo com que teve alguma interação. Note que esta tabela possui tamanho variável de acordo com o número de nodos que a rede possui e com os quais cada um teve interação. A Tabela 4 ilustra o formato do pacote transmitido por cada nodo durante o algoritmo detector de defeitos, e a Figura 22 ilustra o conteúdo das mensagens em um cenário específico.

Tabela 4 - Formato do pacote de dados enviados pelos nodos durante o processo de detecção de defeitos

ID	
Nodo 1	Contador Heartbeat do Nodo 1
Nodo 2	Contador Heartbeat do Nodo 2
Nodo n	Contador Heartbeat do Nodo n

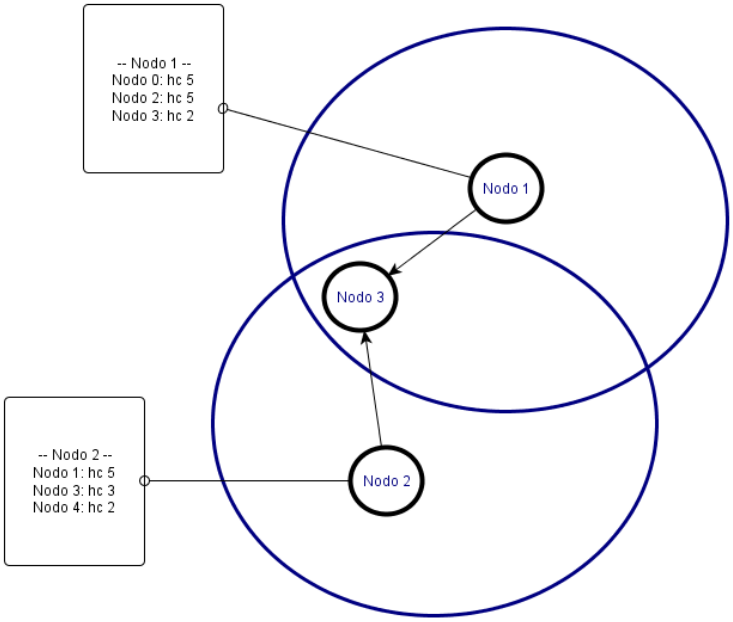

Figura 22 – Conteúdo das mensagens enviadas durante a detecção de defeitos – FFD

A Figura 22 representa a interação entre os nodos em um cenário básico de detecção de defeitos, aonde a variável *hc* representa o contador de *heartbeat* de cada nodo na visão do nodo que envia a mensagem. Observe também que tanto o nodo 1 quanto o nodo 2, enviam ao nodo 3 sua tabela de contadores de *heartbeat* atualizada. Esta tabela será comparada e atualizada pelo nodo receptor, que manterá os maiores valores. Caso o contador de um determinado nodo seja maior que o seu tempo de suspeita ele é considerado falho.

Estas trocas de mensagens realizadas tanto no momento de estimativa de localização quanto para detecção de defeitos ocorrem em momentos separados, e acabam exigindo que o radio do nodo seja utilizado em ambos os momentos, o que aumenta o custo de consumo de potência e acarreta no envio de um número maior de mensagens que podem acabar causando o congestionamento da rede.

O algoritmo híbrido propõem um método de integração entre as técnicas, aonde se aproveite a mensagem já enviada pelo algoritmo de localização para a detecção de defeitos. Obviamente tem-se um aumento no tamanho do pacote enviado, mas a utilização do rádio, que é a maior fonte de consumo de energia do nodo não necessitará ser utilizada em momentos distintos, ou seja, o nodo não necessitará enviar mensagens distintas para detecção de defeitos e estimativa de localização, aproveitando-se de uma única mensagem para o envio destas informações.

É importante observar que a tabela de *heartbeats* de cada nodo não possui um tamanho característico padrão, pois ela depende da quantidade de nodos com que o nodo em questão realizou alguma interação e do tamanho da rede. Logo, não se pode afirmar o quão grande o pacote enviado pelo algoritmo hibrido pode se tornar. Porém, mesmo para pacotes muito grandes o consumo de energia utilizado para o envio de uma mensagem será o mesmo, exceto haja fragmentação da mensagem em mais pacotes de dados, neste caso o nodo enviará um novo pacote de dados contendo o restante da informação. Além disto, a topologia hibrida evita que se tenha mais um cabeçalho a ser transmitido em caso de nova mensagem, como no caso tradicional em que o detector de defeitos ocorre sequencialmente ao algoritmo de localização.

Baseando-se nas mensagens transmitidas durante a fase de filtragem do MCL, a topologia hibrida adiciona o conteúdo que seria enviado pelo detector de defeitos em sua execução, ou seja, a lista de *heartbeats* do nodo, tal qual segue.

$$Hb_x = \{hc_1, hc_2, hc_3, ..., hc_n\}$$

S → Region $\quad HELLO \mid ID_S \mid loc_t \mid Hb_S$

N → Region $\quad HELLO \mid ID_N \mid \{ID_S, loc_{S_t}\} \mid Hb_N$

O valor representado por Hb_x representa a tabela de contadores de *heartbeat* de um nodo ‘x’ qualquer. Logo, hc_1 representa o contador de *heartbeat* do nodo 1, e assim respectivamente até o último nodo com que ‘x’ teve contato. Desta maneira, a cada intervalo

de tempo, os nodos da rede receberão uma mensagem de todos os âncoras presentes dentro de seu alcance contendo sua identificação, coordenadas de localização e sua tabela de contadores de *heartbeat.* O mesmo vale para os nodos vizinhos que além de seus identificadores e informações de outros âncoras, também irão enviar sua tabela de *heartbeats.* As tabelas 5 e 6 ilustram os formatos dos pacotes para os nodos âncoras e vizinhos, respectivamente.

Tabela 5 - Formato do pacote de dados enviado pelos nodos âncoras na Topologia Hibrida

ID	Coordenadas de localização (x,y)
Nodo 1	Contador Heartbeat do Nodo 1
Nodo 2	Contador Heartbeat do Nodo 2
Nodo n	Contador Heartbeat do Nodo n

Tabela 6 - Formato do pacote de dados enviado pelos nodos vizinhos na Topologia Hibrida

ID	
Nodo Semente 1	Coordenadas do Nodo Semente 1
Nodo Semente 2	Coordenadas do Nodo Semente 2
Nodo Semente n	Coordenadas do Nodo Semente n
Nodo 1	Contador Heartbeat do Nodo 1
Nodo 2	Contador Heartbeat do Nodo 2
Nodo n	Contador Heartbeat do Nodo n

Observe que tanto nodos âncoras quanto vizinhos possuem um acréscimo no tamanho de suas mensagens. Este aumento deve-se a incorporação da tabela de *heatbeat* de cada nodo que seria enviada durante a fase de detecção de defeitos. Observe que apesar deste incremento, os nodos não irão mais necessitar enviar uma mensagem para estimar localização e outra para detecção de defeitos, podendo realizar as mesmas tarefas com um número reduzido de mensagens. A Figura 23 ilustra a troca de mensagens realizada pelo algoritmo hibrido.

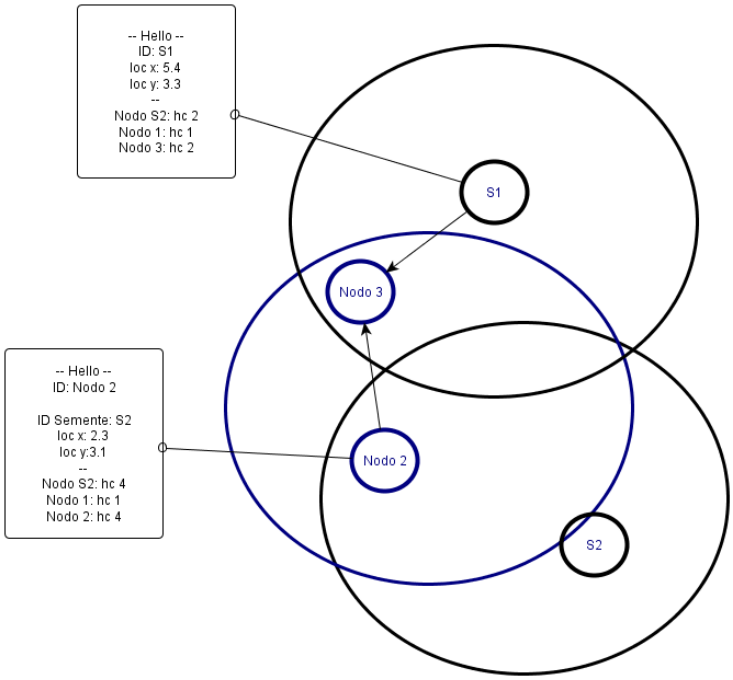

Figura 23 - Mensagens enviadas segundo a topologia hibrida

Observando a Figura 23 pode-se perceber que as mensagens enviadas possuem, além das informações de localização, a tabela com os contadores de *heartbeat* do nodo que enviou a mensagem. É importante salientar também que além de estar se aproveitando da troca de mensagens efetuada pelo algoritmo de localização, o detector de defeitos também deixa de estar implementado diretamente na camada de aplicação para estar implementado dentro do algoritmo de localização, reservando mais capacidade de memória e processamento para aplicações que busquem o propósito funcional do nodo na rede.

Após a troca de mensagens realizada pelos nodos, as informações de localização (coordenadas) são filtradas, enquanto os dados de detecção de defeitos (tabela de *heartbeat*) são repassados ao último estágio da Topologia Híbrida, chamado de estágio de detecção. Porém, antes do inicio do estagio de detecção, os dados de localização recebidos são selecionados a fim de eliminar dados incoerentes. A condição de filtragem de nodos âncoras é dada pela equação 7, aonde ‘S’ representa o conjunto de todos os âncoras ouvidos por um nodo ‘N’ e ‘T’ representa o conjunto de todos os âncoras ouvidos pelos vizinhos de ‘N’.

$$filter(l) = \forall_S \in S, d(l,s) \leq r \wedge \forall_S \in T, r \leq d(l,s) \leq 2r \quad (7)$$

A probabilidade de encontrar sua posição atual com base nas informações recebidas dos nodos âncoras será 0 se a condição do filtro for falsa, desta maneira elimina-se coordenadas inconsistentes com as observações do conjunto de localizações possíveis. As etapas de predição de filtragem irão se repetir até que haja pelo menos um número mínimo ‘n’ de amostras de possíveis localizações.

Nesta abordagem de localização é imprescindível que se discuta a importância da amostragem e re-amostragem. Uma vez que estamos interessados em descobrir a posição posterior do nodo, o algoritmo utiliza uma abordagem de níveis de importância para suas amostras, desta forma consegue-se selecionar amostras de maior peso e traçar um possível trajeto do nodo.

Pode-se dizer então que durante a etapa de filtragem o nodo atualiza os pesos de suas amostras (possíveis localizações) com base nas informações recebidas dos nodos âncoras e vizinhos. A equação 8 representa esta etapa, aonde ‘ $\tilde{w}_t^i$ ’ representa a atualização dos pesos das amostras anteriores ‘ $\tilde{w}_{t-1}^i$ ’, com base nas observações recebidas ‘ $p(o_t \mid l_t^i)$ ’.

$$\tilde{w}_t^i = \tilde{w}_{t-1}^i . p(o_t \mid l_t^i) \qquad \textbf{(8)}$$

Após a atualização dos pesos das amostras, os mesmos são normalizados de $\tilde{w}_t^i$ para w_t^i, conforme sua importância e um conjunto ponderado é utilizado na simulação da posição posterior do nodo, conforme equação 9.

$$w_t^i = \frac{\tilde{w}_t^i}{\sum_{k=1}^{N} \tilde{w}_t^k} \qquad \textbf{(9)}$$

Após a amostragem, a re-amostragem (do inglês, *re-sampling*) é responsável por eliminar trajetórias que possuam amostras de pouco peso. A medida de degeneração das amostras pode ser dada pelo tamanho da amostra efetiva N_{eff} ilustrada na equação 10.

$$N_{eff} \approx \frac{1}{\sum_{i=1}^{N} (w_t^i)^2} \qquad \textbf{(10)}$$

O valor de N_{eff} é dado aproximadamente pelo inverso do somatório dos pesos das amostras ao quadrado. Quando seu valor esta abaixo de um valor fixo de *threshold* ($N_{threshold}$) é necessária a re-amostragem. O valor de N_{eff} é exatamente o número de amostras válidas, logo só é necessário que se mantenha um número suficiente de amostras válidas (peso elevado) para que N_{eff} seja maior que $N_{threshold}$. Como dito anteriormente as etapas de predição e filtragem irão se repetir até que haja pelo menos um número mínimo de amostras para possíveis localizações, porém os dados de detecção somente serão transmitidos uma única vez a cada novo calculo de coordenadas do algoritmo de localização, a fim de evitar reenvio de informações, diminuir o tamanho da mensagem e consequente consumo de energia.

A etapa de detecção é o terceiro e último estágio da topologia hibrida, neste estágio, caso haja novas informações de detecção (nova mensagem com lista de *heartbeat*) o nodo irá atualizar sua tabela e verificar possíveis nodos falhos. Para isto irá comparar a nova tabela com os contadores de *heartbeats* com a sua mantendo os maiores valores.

Caso algum nodo vizinho não esteja mais dentro de sua região de abrangência o detector não levantará suspeita sobre ele automaticamente, e sim aguardará um período de ‘γ’ *heartbeats*, levando em consideração que este nodo pode estar apto e somente ter saído de sua região de abrangência.

O tempo para se considerar um nodo suspeito é inicializado pelo valor de ‘$\beta(\gamma_o)$’, onde ‘γ_o’ representa o *timeout* de recebimento do primeiro *heartbeat* do nodo mais distante. O valor de ‘γ_o’ pode ser dado pela equação 11, aonde ‘D’ representa o diâmetro da rede e ‘r’ a distância entre os nodos.

$$\gamma_0 = \left\lceil \frac{D}{D-r} \right\rceil \qquad \textbf{(11)}$$

Caso o nodo receba uma tabela com um contador maior que o local um novo valor de tempo de suspeita é calculado, vide equação 12, aonde σ representa o atraso da rede para a distância de um hop e π o tempo necessário para se enviar um *heartbeat.*

$$\beta(\gamma) = \gamma * (\pi + \sigma) \qquad \textbf{(12)}$$

Porém, se o nodo não receber um valor de *heartbeat* de outro nodo com que realizou alguma interação, após a passagem do último tempo de suspeita calculado, ele irá levantar uma suspeita de funcionamento ao nodo em questão.

A imagem a seguir ilustra os 3 estágios da topologia hibrida.

Figura 24 - Estágios da Topologia Hibrida

Como pode-se observar a detecção de defeitos é realizada no momento pós-filtragem. De forma mais detalhada, o algoritmo hibrido possui uma primeira fase de predição aonde irá predizer sua localização baseando-se no seu ponto anterior e sua velocidade máxima. Após esta etapa, diversos pontos de localização serão sugeridos e filtrados pelas mensagens recebidas de nodos âncoras e vizinhos durante a etapa de filtragem. No processo de filtragem e atualização, o nodo recebe mensagens de outros nodos e estima sua posição momentânea e posterior através da estimativa de N_{eff} e de um sistema de pesagem das amostras recebidas. É importante ressaltar que a troca de informações com nodos vizinhos e âncoras só é feita na etapa de filtragem. A detecção de defeitos é executada posteriormente ao processo de filtragem, e irá comparar as tabelas de *heartbeats* recebidas de outros nodos e verificar se há ou não nodos suspeitos na rede.

Desta forma buscou-se implementar uma arquitetura que conseguisse integrar tanto um algoritmo de localização quanto um detector de defeitos, no cenário mais desconhecido e aleatório possível, utilizando-se de técnicas que permitissem uma grande mobilidade e elevado número de nodos na rede sem que houvesse uma diminuição na precisão dos algoritmos integrados e permitisse a ambos atuarem colaborativamente a fim de reduzir o número de pacotes enviados pelos nodos.

4.3. Conclusões Parciais

Este capítulo apresentou a arquitetura da Topologia Hibrida aonde se implementou um detector de defeitos dentro de um algoritmo de localização de nodos. Tal integração visa a

otimização do conteúdo das mensagens trocadas entre os nodos de maneira a reduzir o número de mensagens enviadas e o consequente consumo de energia ocasionado pelo envio de um número maior de pacotes.

A Topologia Hibrida foi implementada da camada de aplicação, porém atualmente já existem algoritmos de localização implementados diretamente na camada de hardware como em (OLIVEIRA, 2009). Para detectores de defeitos ainda não há registros de uma aplicação em tão baixo nível. Sendo assim, a topologia proposta nesta dissertação também abre caminho para trabalhos futuros que busquem o desenvolvimento de aplicações hibridas em hardware visando um maior desempenho para a rede, como uma diminuição ainda maior no consumo de energia e o aumento da capacidade de processamento em alto nível. Além das considerações realizadas anteriormente, é a primeira vez na bibliografia que um trabalho busca a integração destes dois algoritmos para redes de sensores móveis.

5. AVALIAÇÃO E VALIDAÇÃO

Este capítulo apresenta os experimentos e testes realizados na implementação da arquitetura hibrida, bem como o seu impacto na redução do número de mensagens enviadas na rede e consumo de energia. Na avaliação foram utilizadas algumas das métricas de análise de desempenho de rede propostas em (TAN, 2006).

O capítulo também apresenta o simulador utilizado nos experimentos e o cenário em que os testes foram realizados.

5.1. AMBIENTE DE DESENVOLVIMENTO E VALIDAÇÃO

O ambiente de simulação utilizado para a validação do algoritmo de topologia hibrida proposto neste trabalho foi desenvolvido por Hu e Evans em (HU; EVANS, 2004), para realizar a validação do algoritmo de localização MCL. Os arquivos e pastas que compõem o simulador são apresentados na Figura 25.

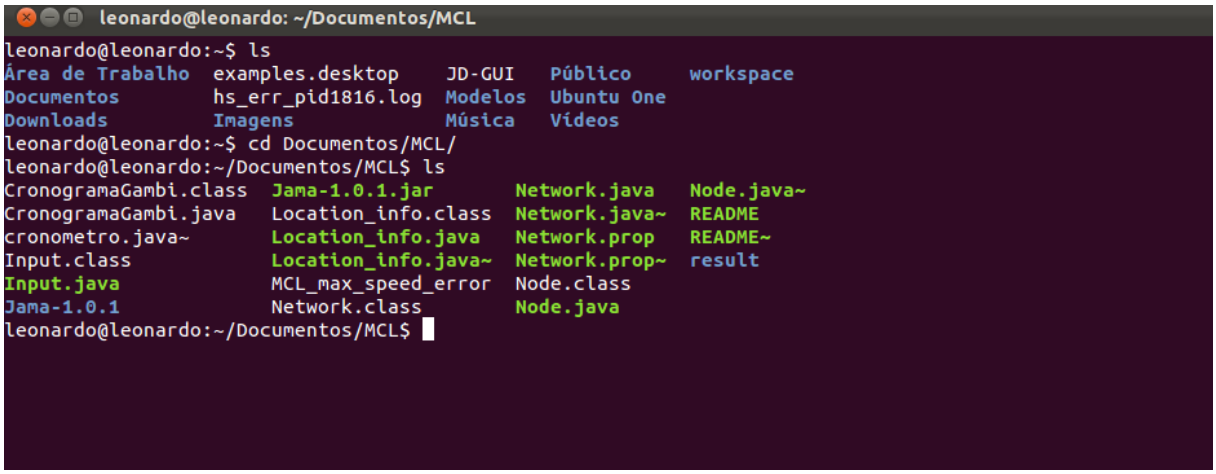

Figura 25 - Arquivos de simulação do simulador MCL (HU; EVANS, 2004)

Tais arquivos e pastas representam:

- *Network.prop: Arquivo de configuração da rede*
- *Input.java: Lê e atualiza parâmetros de rede*
- *Network.java: Classe da rede*
- *Node.java: Classe dos nodos individuais*

- *Location_info.java: Estrutura de dados utilizada para gravar informações de localização*
- *Jama-1.0.1.jar: Biblioteca utilizada para operações com matrizes*
- *result/: Pasta com os resultados experimentais*

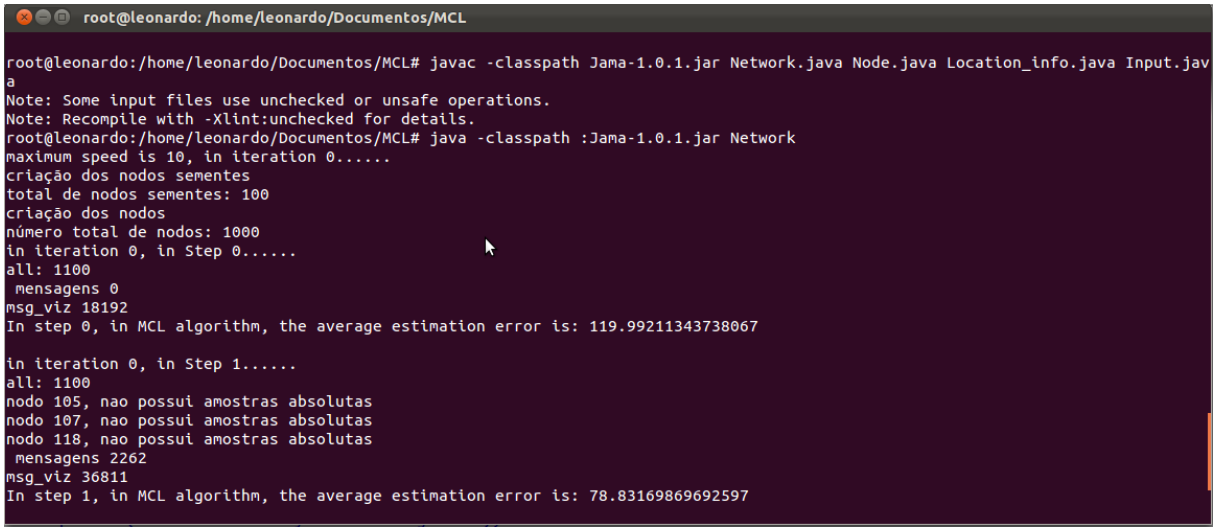

Figura 26 - Simulador MCL em funcionamento

A Figura 26 ilustra o simulador em funcionamento, além das análises relativas ao próprio funcionamento do simulador quanto ao MCL, ele também foi adaptado para detectar falhas e analisar a troca de mensagens entre os nodos. Para compilar e executar o simulador seguem os comandos, respectivamente:

- *javac –classpath Jama-1.0.1.jar Network.java Location_info.java Input.java*
- *java –classpath :Jama-1.0.1.jar Network*

O simulador foi desenvolvido em ambiente Java e pode ser encontrado em *http://www.cs.virginia.edu/mcl*. É importante ressaltar também que para a validação do algoritmo hibrido foi utilizado o software MATLAB (GILAT, 2006) para obtenção e análise dos dados obtidos.

Inicialmente é importante ressaltar que para todas as simulações foi adotado o método de mobilidade "*random waypoint mobility model*" (CAMP; BOLENG; DAVIES, 2002), através do qual todos os nodos desconhecidos e âncoras da rede se movem aleatoriamente, sem especificar sua velocidade ou direção.

Além desta característica, a rede de sensores móveis simulada neste trabalho possui como propriedades:

- Distribuição dos nodos é feita de forma aleatória
- Tamanho da rede é de 600m x 600m
- O alcance de transmissão de nodos e âncoras é de 50 metros
- A velocidade dos nodos varia de [0, v_{max}]

5.2. RESULTADOS OBTIDOS E ANÁLISE DE DESEMPENHO

Algumas das características básicas das redes de sensores são as altas densidades de elementos, restrições severas em relação ao consumo de energia, muitos fluxos de informações além da redundância de dados enviados e o trabalho cooperativo realizado pelos nodos.

A partir destas informações, e segundo (TAN, 2006), pode-se definir algumas das métricas chaves para medição de desempenho de uma rede de sensores móveis. Basicamente estas métricas visam garantir uma qualidade de serviço da rede que maximize a vida útil dos sensores permitindo uma troca de informações entre eles que seja suficiente. Dentre as métricas propostas por (TAN, 2006) este trabalho irá utilizar a medição do número de mensagens transmitidas e o consumo de energia economizado pelo método proposto.

Para a simulação e validação da topologia hibrida que será mostrada a seguir, comparou-se a ela um algoritmo padrão, ou seja, que possua tanto o modelo MCL de localização quanto o detector de defeitos atuando separadamente, sem que um tome conhecimento do outro. É importante notar que mesmo sem a integração proposta na topologia hibrida, o algoritmo padrão possui tanto o MCL como o detector de defeitos Friedman atuando paralelamente. Um ponto importante a ser frisado antes da análise dos dados de simulação é a definição do eixo x - tempo, como o simulador utilizado trabalha com tempo discreto, todas as análises serão feitas levando-se em consideração os chamados *rounds* ou iterações realizadas, que representam o tempo de cada nodo, ou seja, a cada interação todos os nodos terão sua vez de atuar.

Neste capítulo se quer validar a topologia hibrida como sendo uma integração de algoritmos de localização e detectores de defeitos, que através de otimização do conteúdo (o conteúdo das mensagens de um algoritmo foi somado ao de outro algoritmo) das mensagens enviadas permita que um número menor de pacotes seja transmitido pelos nodos, e que consequentemente essa redução acarrete em uma economia de energia na transmissão de mensagens, ou seja, ao enviar um número menor de pacotes diminui-se o gasto de energia nestas transmissões.

Com base nestes objetivos, a validação deste trabalho ocorre através da análise do número de mensagens enviadas em redes que possuam uma aplicação padrão contendo um detector de defeitos Friedman e um algoritmo de localização MCL não integrados. E a Topologia Híbrida que possui para os mesmos nodos da rede, ambos os algoritmos integrados. Desta forma é possível se avaliar a redução do número de mensagens enviadas por um nodo e em toda a rede durante a execução de ambas as metodologias.

Também foi analisado o consumo de energia dos nodos em ambas as aplicações durante a etapa de transmissão e recepção de mensagens, que são responsáveis por grande parte do consumo de energia da rede. O tamanho médio dos pacotes pode variar de acordo com cada nodo e aplicação, porém para esta simulação se utilizou o tamanho máximo do pacote dos nodos sensores comerciais MICAz, que é de 55 bytes, para ambas as topologias

Foram feitos testes para 120, 550 e 1100 nodos de maneira a avaliar o comportamento do algoritmo hibrido em redes de diferentes tamanhos. As informações de simulação podem ser observadas na Tabela 7.

Tabela 7 - Configurações de Simulação

Configurações de Simulação	
Configurações Gerais	
Tamanho da Rede	600 m x 600 m
Alcance dos Nodos	50 m
Velocidade Mínima	0 m/s
Velocidade Máxima	10 m/s
Tamanho do Pacote	55 Bytes
Algoritmos Simulados	
Aplicação Padrão	Sem Integração
Aplicação Hibrida	Com Integração
Número de Nodos Simulados	
120 nodos	
20 âncoras	100 desconhecidos
550 nodos	
50 âncoras	500 desconhecidos
1100 nodos	
100 âncoras	1000 desconhecidos
Consumo de Energia	
Potência de transmissão	60 mW (ou mJ/s)
Potência de recepção	40 mW (ou mJ/s)

Simulação 120 nodos

Em um primeiro cenário é analisado uma rede que possui 100 nodos desconhecidos e 20 âncoras, totalizando 120 nodos. As Figuras 27, 28 e 29 ilustram o número de mensagens transmitidas pelo algoritmo padrão, híbrido e a diferença entre ambos, respectivamente, após um período de 10 iterações.

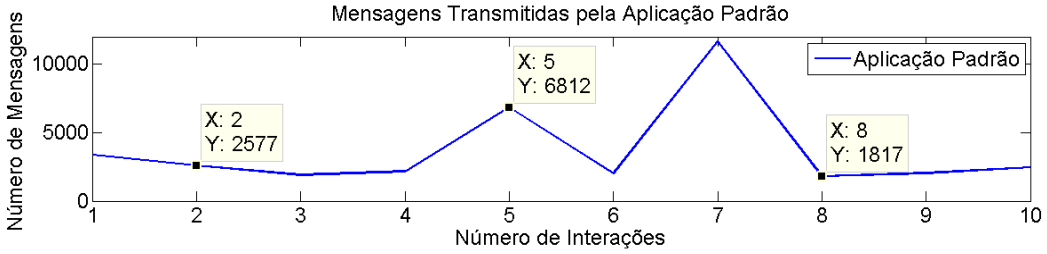

Figura 27 - Número de mensagens transmitidas pela aplicação padrão

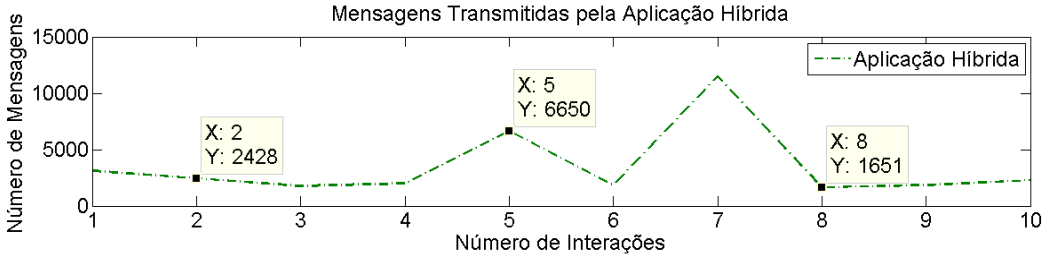

Figura 28 - Número de mensagens transmitidas pela aplicação híbrida

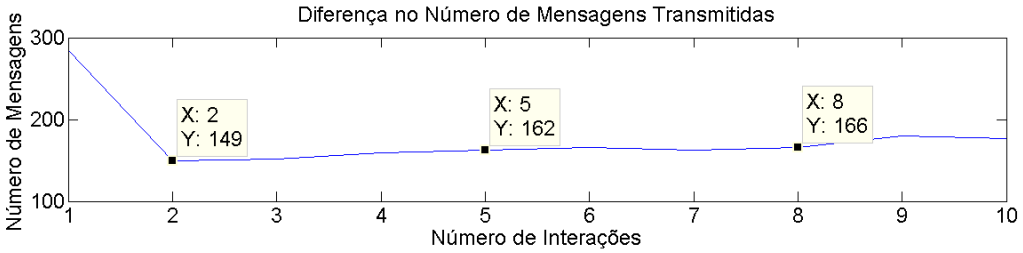

Figura 29 - Economia de mensagens enviadas – 120 nodos

Conforme mencionado anteriormente cada iteração representa um intervalo de tempo aonde os nodos trocam informações, realizam a detecção de defeitos e estimam suas coordenadas.

As Figuras 27, 28 e 29 mostram claramente uma redução no número de mensagens enviadas quando a rede esta implementada sob a topologia hibrida. Observe que durante a interação 5, por exemplo, da Aplicação Padrão a rede envia um total de 6812 mensagens enquanto a aplicação híbrida enviou 6650 para a mesma rede, ou seja, uma redução de 162 mensagens. Já durante a interação 8 a aplicação híbrida conseguiu uma economia de 1817 para 1651 mensagens, uma redução de quase 10% do número total de mensagens enviadas.

Como se pode observar na Figura 29 a diferença do número de mensagens enviadas pelos algoritmos padrão e híbrido possui uma queda considerável entre as interações 1 e 2. Sabemos que esta diferença se deve ao fato do detector de defeitos se aproveitar das mensagens enviadas pelo algoritmo de localização para realizar sua comunicação. Logo, esta diferença inicial se deve ao fato de no primeiro momento de simulação (interação 1) os nodos necessitarem enviar um número maior de mensagens pois não possuem nenhuma informação sobre a rede em que estão inseridos. Estima-se que neste primeiro momento os nodos irão enviar quase o dobro de mensagens que serão enviados na interação 2, por exemplo.

A Tabela 8 ilustra o número de mensagens enviadas pelos nodos quando executada a aplicação padrão, a aplicação hibrida e a diferença entre ambas.

Tabela 8 - Número de mensagens transmitidas para 120 nodos

Interação	Ap. Híbrida	Ap. Padrão	Economia	Economia %
1	3100	3385	285	8,41
2	2428	2577	149	5,78
3	1752	1904	152	7,98
4	2000	2159	159	7,36
5	6650	6812	162	2,37
6	1850	2016	166	8,23
7	11500	11663	163	1,39
8	1651	1817	166	9,13
9	1850	2030	180	8,86
10	2269	2446	177	7,23
Total	35050	36809	1759	6,67

Como se observa na Tabela 8, a aplicação híbrida conseguiu uma taxa de redução de mensagens que variou de 1,3% (interação 7) até 9,1% (interação 8) em comparação com a

aplicação padrão, o que ao final de 10 interações significou uma economia de 1759 mensagens, ou ainda 6,67% do total enviado pela topologia padrão. Se realizarmos uma média do número total de mensagens transmitidas na rede pelo número de nodos ao final de 10 interações se obterá uma economia de aproximadamente 15 mensagens por nodo ao final desta etapa. É importante ressaltar que cada interação representa a execução das etapas de detecção de defeitos e localização de ambas às aplicações.

É importante ressaltar que esta diminuição ocorre porque os nodos não estão mais enviando mensagens somente para detectar defeitos, e sim aproveitando as mensagens enviadas pelo algoritmo de localização para realizar sua tarefa.

A Tabela 8 também nos mostra que o número de mensagens transmitidas pode aumentar ou diminuir a cada interação, isto ocorre devido a configuração da rede em cada iteração. Por exemplo, quando um maior número de nodos estiver próximo, fazendo parte de suas respectivas regiões de abrangência, a troca de mensagens na rede será elevada, caso estes nodos se dispersem o envio de informações tende a ser menor.

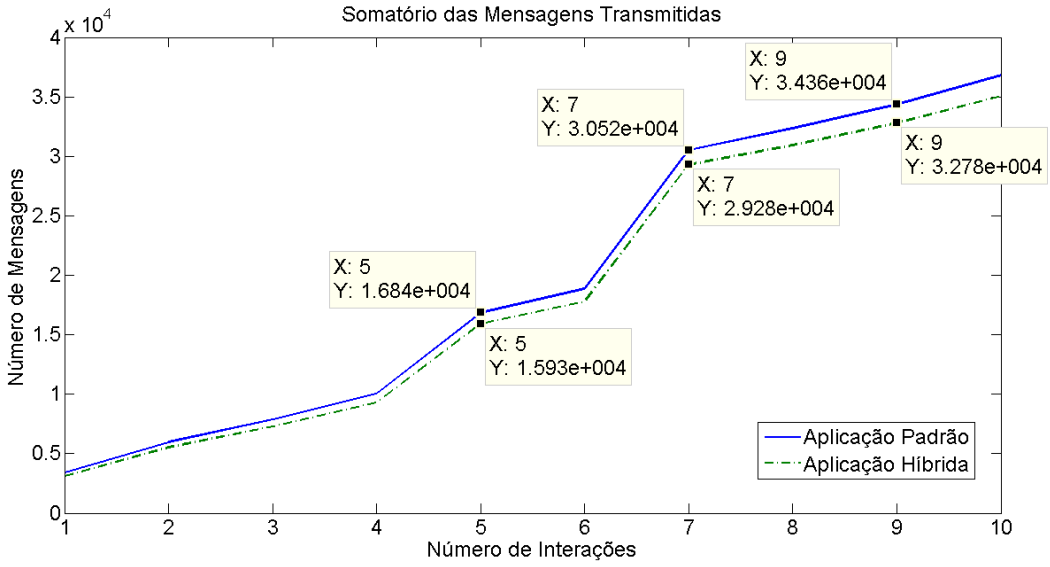

Figura 30 - Somatório de mensagens enviadas – 120 nodos

A Figura 30 ilustra o somatório das mensagens transmitidas pelos nodos após cada interação em ambas as topologias simuladas. Neste caso, a diferença do número de mensagens transmitidas aumenta com o passar do tempo, ou seja, a topologia híbrida consegue transmitir um número menor de mensagens em todas as interações simuladas, o que proporciona uma redução final do número de mensagens elevada e tem impacto direto na redução do consumo de energia dos nodos.

Conforme descrito anteriormente, o maior gasto de energia de um nodo esta no seu módulo de rádio, logo uma redução no número de mensagens transmitidas impacta diretamente na autonomia da bateria do nodo.

Se tomarmos como exemplo um nodo sensor comercial padrão MICAz CC2420 que capacidade de bateria de 2000mAh e 2,7V de tensão mínima, é possível se calcular o ganho de vida útil que os nodos sensores terão com a economia de mensagens proposta na topologia híbrida para fins de comparação.

A Figura 31 ilustra a diferença quanto ao consumo de energia gasto com a transmissão de mensagens nas aplicações padrão e híbrida.

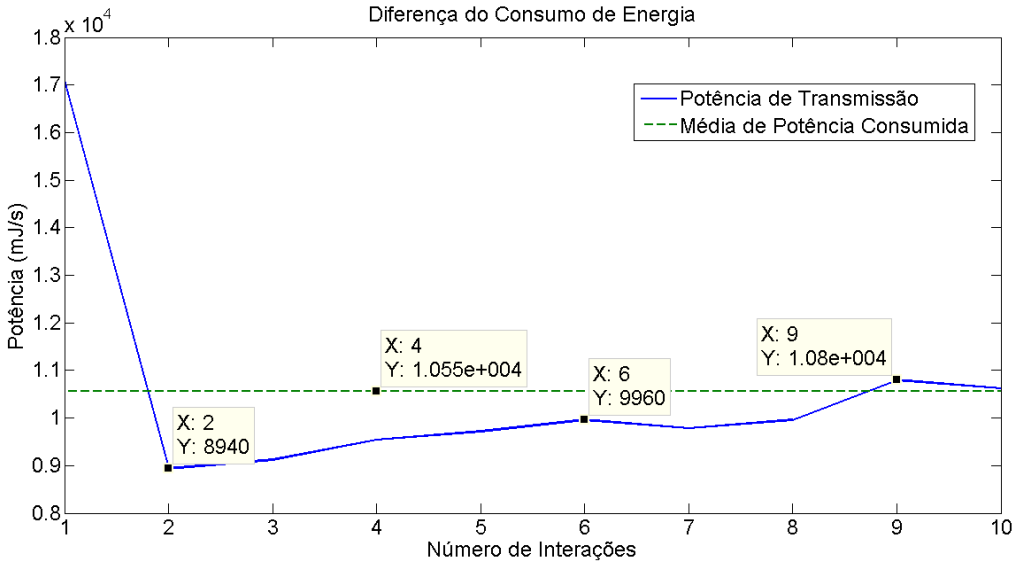

Figura 31 - Economia de Energia entre as Aplicações – 120 nodos

Como nos mostra o gráfico da Figura 31, a redução de mensagens ocasionada pela topologia híbrida tem impacto direto na redução do consumo de energia dos nodos. A diferença do consumo de energia utilizado na transmissão de mensagens esta condicionada ao número de transmissões da mesma e varia a cada interação. Na simulação realizada para 120 nodos, podemos observar que ao final de 10 interações a topologia híbrida tem uma média de potência economizada de 10550 mJ/s quando comparada a topologia padrão, o que representa uma economia de 87,9 mJ/s por nodo. O consumo médio de energia por nodo na topologia padrão foi de 1840,2mW, enquanto a média da topologia híbrida para o mesmo número de iterações (10), foi de 1752,5mW.

Se considerarmos este valor médio para fins de comparação, é possível se calcular o ganho de vida útil que o nodo sensor terá com a economia de mensagens proposta na

topologia hibrida. Para isto divide-se o valor da potência economizada pela tensão da bateria (2,7V) a fim de se obter a corrente praticada, após divide-se a capacidade da bateria de 2000mAh pela corrente praticada e se tem a vida útil média ganha através da economia de energia proporcionada pela topologia hibrida.

Consumo médio de um nodo na Topologia Padrão:

$$\frac{1840,2mW}{2,7V} = 681,55mA$$
$$\frac{2000mAh}{681,55mA} = 2,93h$$

Consumo médio de um nodo na Topologia Híbrida:

$$\frac{1752,5mW}{2,7V} = 649,07mA$$
$$\frac{2000mAh}{649,07mA} = 3,1h$$

Vida útil ganha:

$$3,1h - 2,93h = 0,17h$$

Como se observa na demonstração acima, a diferença média de consumo de energia da rede de 10550 mJ/s, entre as aplicações padrão e hibrida, proporcionaria a cada nodo um aumento de 0,17h na sua vida útil para esta bateria, o que corresponde em um aumento de 5,8% na vida útil da bateria.

Nesta análise para redes pequenas torna-se claro que um algoritmo integrado que otimize o conteúdo das mensagens transmitidas tem impacto considerável no desempenho e vida útil da rede, proporcionando não somente uma diminuição do número de mensagens trocadas na rede, como aumento de vida útil dos nodos. Mesmo assim para uma análise detalhada torna-se necessário a comparação desta rede de 120 nodos com redes de maior tamanho.

Simulação 550 nodos

Na simulação com 550 nodos busca-se avaliar o desempenho da topologia hibrida em redes que possuam maior conectividade. Neste ambiente tem-se 500 nodos desconhecidos e

50 nodos âncoras, totalizando 550 nodos, utilizando-se o mesmo ambiente e propriedades de simulação.

Conforme descrito inicialmente nesta seção, a primeira análise a ser realizada pelos algoritmos híbrido e padrão é quanto ao número de mensagens transmitidas por ambos os nodos. As Figura 32 e 33 ilustram esta comparação.

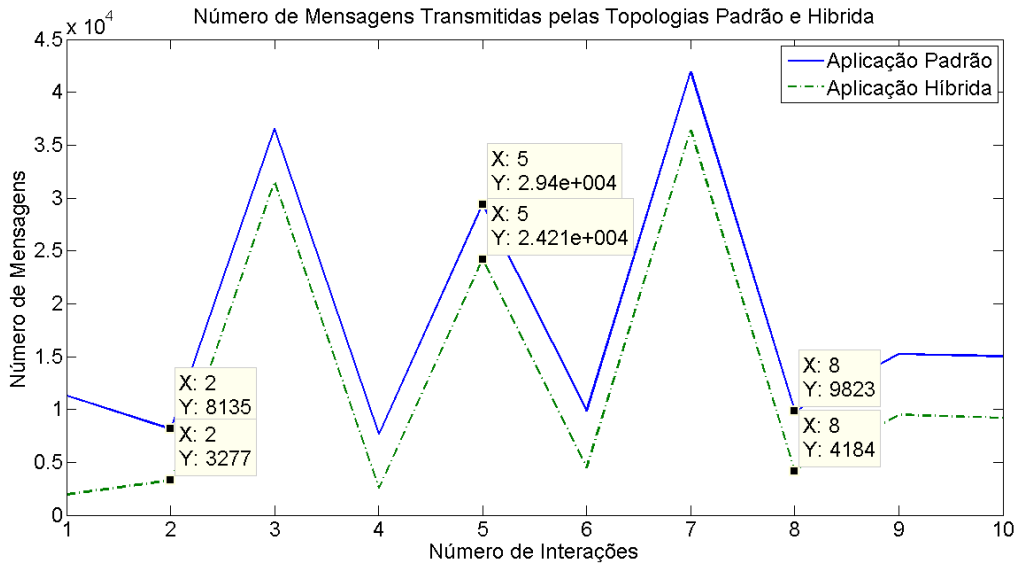

Figura 32 - Número de mensagens transmitidas em ambas as topologias

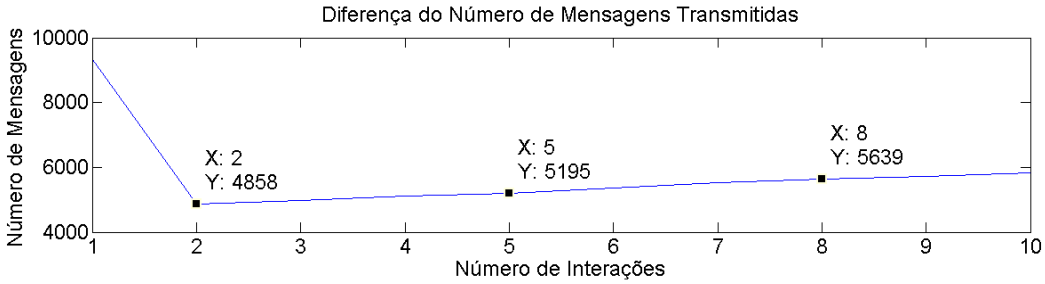

Figura 33 - Economia de mensagens transmitidas

Como se observa na Figura 33, as redes que possuem a topologia híbrida implementada ao invés do método tradicional padrão, conseguem uma redução considerável no número de mensagens enviadas. Na interação 2, por exemplo, a topologia híbrida obteve uma redução de 4858 mensagens se comparada a aplicação padrão, já na interação 8 essa diferença chegou a 5639 mensagens, aproximadamente 49% das mensagens transmitidas.

Isto nos mostra que em redes maiores e que possuam um grande número de troca de informações teremos uma redução média do número de mensagens bem mais significativa. A Tabela 9 ilustra os número de mensagens transmitidas dos algoritmos padrão e híbrido.

Tabela 9 - Número de mensagens transmitidas para 550 nodos

Interação	Ap. Híbrida	Ap. Padrão	Economia	Economia %
1	1960	11297	9337	82,65
2	3277	8135	4858	59,71
3	31566	36549	4983	13,63
4	2548	7661	5113	66,74
5	24208	29403	5195	17,66
6	4506	9861	5355	54,30
7	36450	41970	5520	13,15
8	4184	9823	5639	57,40
9	9493	15213	5720	37,59
10	9216	15033	5817	38,69
Total	127408	184945	57537	44,15

Como descrito na simulação anterior, a primeira interação possui um maior número de mensagens transmitidas, pois os nodos não tem conhecimento de seu ambiente de simulação e enviam um número maior de mensagens. A partir da segunda interação a rede é normalizada quanto ao envio de pacotes.

Como se observa na Tabela 9, em uma rede com mais nodos temos um número maior de mensagens transmitidas e consequentemente uma economia maior no envio de mensagens por parte da topologia hibrida. Percentualmente esta economia chegou a 44% da mensagens ao longo de 10 interações, mesmo se desconsiderarmos a primeira interação teremos um média 39% de redução no número de mensagens enviadas.

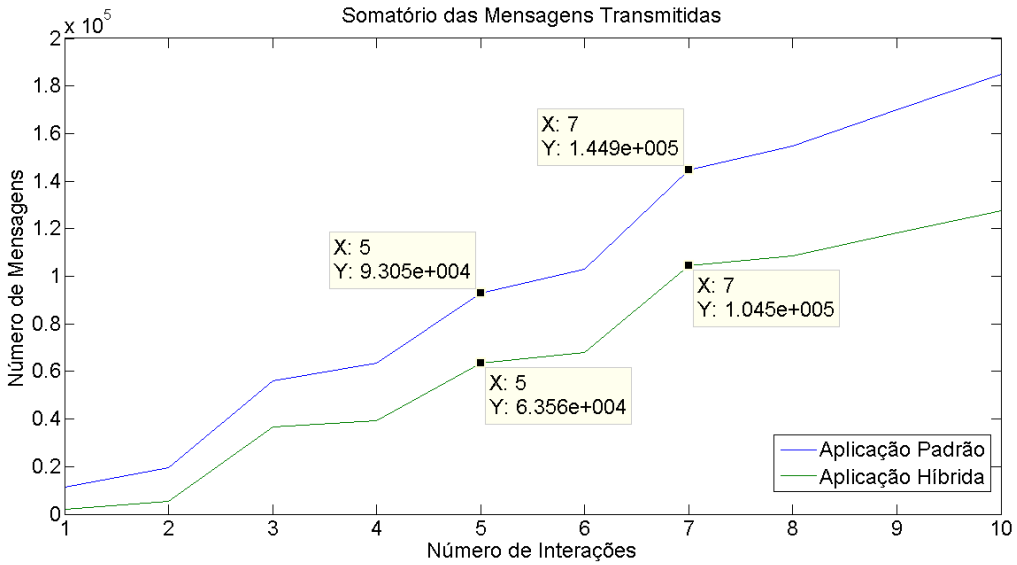

Figura 34 - Somatório de mensagens enviadas – 550 nodos

É importante ressaltar que o número de mensagens transmitidas varia a cada interação dependendo do número de vizinhos de cada nodo. A Figura 34 ilustra o somatório das mensagens de ambas as topologias. Assim como na simulação com 120 nodos a diferença do somatório das mensagens enviadas pelos algoritmos padrão e híbrido aumenta com o tempo, ou seja, é possível se afirmar que a topologia híbrida sempre irá enviar um número menor de mensagens que a topologia padrão.

Avaliando o impacto desta economia de mensagens transmitidas na vida útil dos nodos, obteve-se uma economia média de 345222 mJ/s ao longo de 10 interações, ou seja, 627,67 mJ/s conforme ilustra a Figura 35.

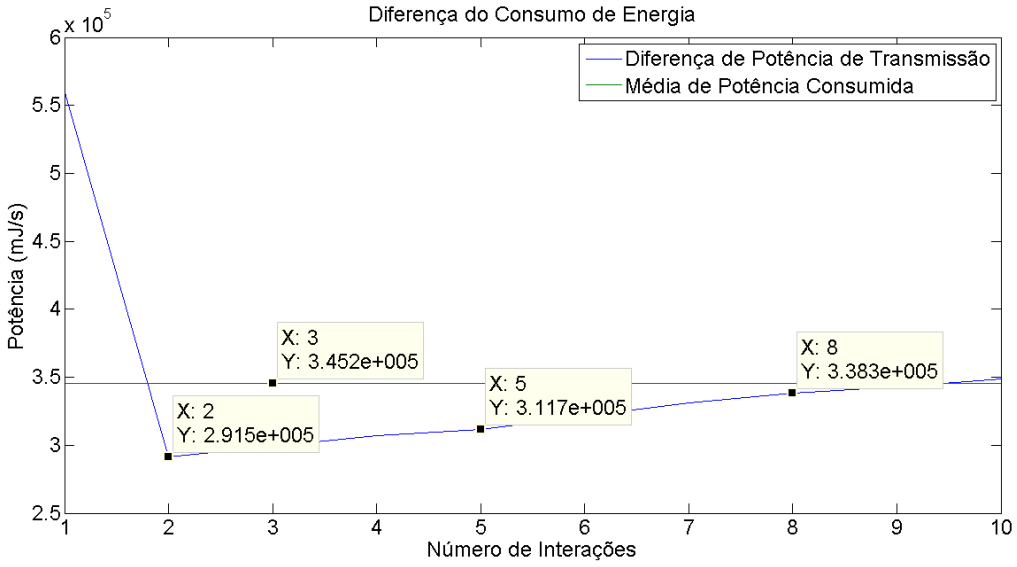

Figura 35 - Economia de Energia entre as Aplicações - 550 nodos

Assim com ilustra a Figura 35, a economia de energia é diretamente proporcional a diminuição do número de mensagens transmitidas, ou seja, quanto menor o número de mensagens transmitidas, menor será a utilização do rádio dos nodos e consequentemente menor será o seu consumo de energia, resultando num aumento de vida útil.

O consumo médio de energia por nodo utilizado na aplicação padrão durante as 10 iterações foi de 2007,67mW, enquanto na topologia híbrida este mesmo consumo foi de 1380mW por nodo.

Utilizando-se o valor médio de economia de potência de 345222 mJ/s calculado durante as 10 interações, afirma-se que cada nodo terá em média uma economia de 627,67 mJ/s o que representa um aumento médio de 1,23 horas na vida útil de cada nodo da rede para a bateria do nodo MICAz de 2000mAh e 2,7V, o que representa uma economia média de 45,89% na vida útil da bateria.

Consumo médio de um nodo na Topologia Padrão:

$$\frac{2007,67mW}{2,7V} = 743,58mA$$

$$\frac{2000mAh}{743,58mA} = 2,68h$$

Consumo médio de um nodo na Topologia Híbrida:

$$\frac{1380mW}{2,7V} = 511,11mA$$

$$\frac{2000mAh}{511,11mA} = 3,91h$$

Vida útil ganha:

$$3,91h - 2,68h = 1,23h$$

Nesta análise para 550 nodos fica evidente que a topologia híbrida tem um impacto ainda maior em redes com maior conectividade. Observa-se que para um mesmo ambiente de simulação foram aumentados 430 nodos em comparação com a simulação anterior, resultando numa maior conectividade e interação entre os nodos.

Este maior número de nodos em um mesmo espaço, faz com que existam mais nodos dentro de regiões de abrangência e consequentemente mais mensagens são enviadas. Com este escopo os benefícios da utilização da topologia híbrida ficam ainda mais evidentes ao modo que diminuem consideravelmente as mensagens transmitidas.

É possível observar que o impacto na vida útil dos nodos foi ainda maior nesta simulação do que na anterior. Isto deve-se ao fato de haver uma maior diferença entre o número médio de mensagens transmitidas na rede pelas topologias padrão e híbrida, impactando assim, em uma maior economia de bateria. No caso individual desta simulação este ganho médio na vida útil da bateria dos nodos chegou a 45,89%.

Simulação 1100 nodos

A terceira simulação analisa o desempenho do algoritmo hibrido em redes com um grande número de nodos. Para isto são simulados 1000 nodos desconhecidos e 100 nodos âncoras, um total de 1100 nodos, utilizando-se o mesmo ambiente e propriedades de simulação.

As Figuras 36 e 37 ilustram o número de mensagens trocadas pela aplicação padrão e pela topologia hibrida em um ambiente com 1100 nodos e a diferença entre ambas, respectivamente.

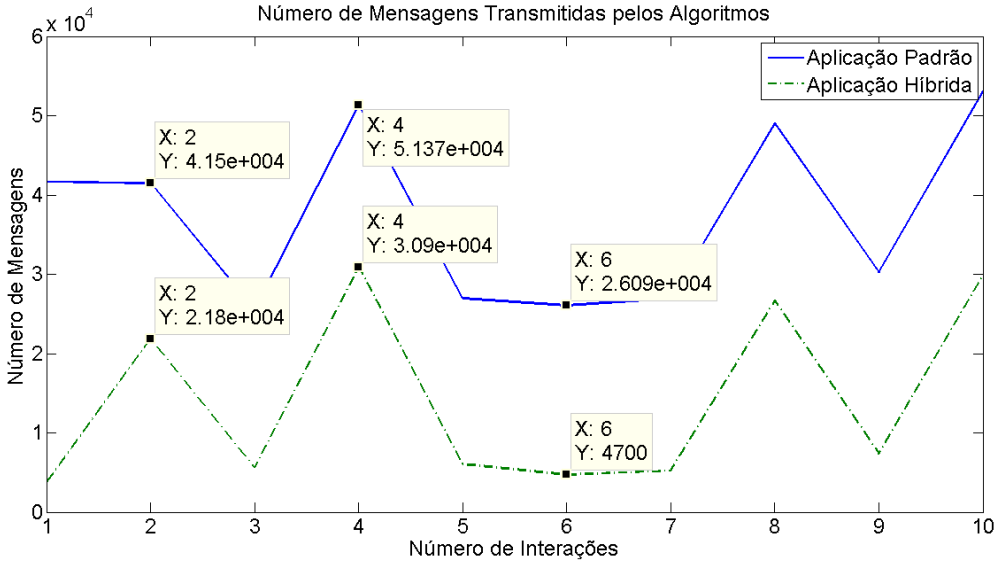

Figura 36 - Número de mensagens transmitidas - 1100 nodos

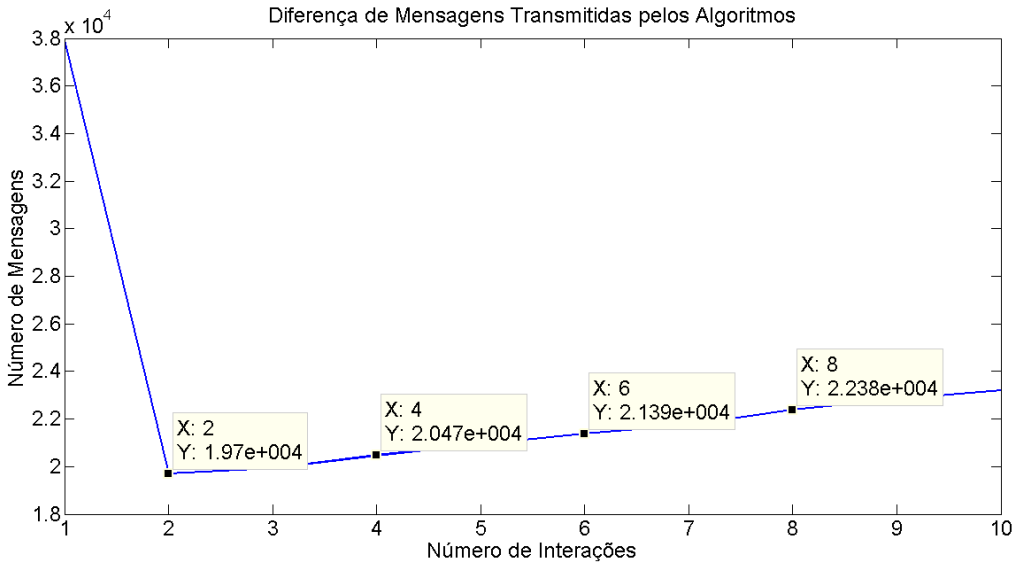

Figura 37 - Diferença de mensagens transmitidas pelos algoritmos Padrão e Híbrido

As figuras 36 e 37 ilustram o mesmo comportamento da topologia híbrida em relação a topologia padrão simulados nas análises anteriores. Observe que para uma rede com 1100 nodos a economia de mensagens é ainda maior, como na interação 6 aonde a diferença entre os algoritmos chegou a aproximadamente 21390 mensagens, ou 81% em relação a aplicação padrão.

Obviamente nesta iteração temos uma redução considerável que provavelmente não se repetirá em outras iterações. É importante observar que quanto maior a mobilidade da rede, maior a precisão do algoritmo de localização e com isto, menor o número de mensagens que os nodos necessitarão enviar. Aliado a isto, em interações aonde há grande diferença percentual entre as aplicações padrão e híbrida ocorre uma grande troca de mensagens na aplicação padrão referente a detecção de defeitos, ou seja, os nodos possuem um número elevado de vizinhos e trocam um grande número de informações referente a detecção de defeitos. Na aplicação híbrida estas mensagens são economizadas, pois o detector utiliza o algoritmo de localização para envio de suas mensagens.

Assim como nas outras simulações realizadas, o número de mensagens transmitidas irá variar de acordo com o número de vizinhos que cada nodo possui. A Tabela 10 ilustra os valores referentes ao número de mensagens transmitidas por ambos os algoritmos.

Tabela 10 - Número de mensagens transmitidas para 1100 nodos

Interação	Ap. Híbrida	Ap. Padrão	Economia	Economia %
1	3685	41629	37944	91,14
2	21804	41502	19698	47,46
3	5617	25545	19928	78,01
4	30900	51371	20471	39,84
5	6044	26976	20932	77,59
6	4700	26087	21387	81,98
7	5219	26972	21753	80,65
8	26652	49036	22384	45,64
9	7350	30176	22826	75,64
10	29845	53041	23196	43,73
Total	141816	372335	230519	61,79

É possível observar na Tabela 10 que a aplicação híbrida conseguiu, em média, uma redução de 61,79% das mensagens, se desconsiderarmos a interação 1, aonde os nodos enviam mais informações por desconhecerem o ambiente de simulação, ainda teremos 58,56% de economia de mensagens enviadas.

O impacto desta economia de transmissão de mensagens na vida útil dos nodos pode ser observada na Figura 38, que ilustra a economia de energia obtida pela utilização da aplicação híbrida.

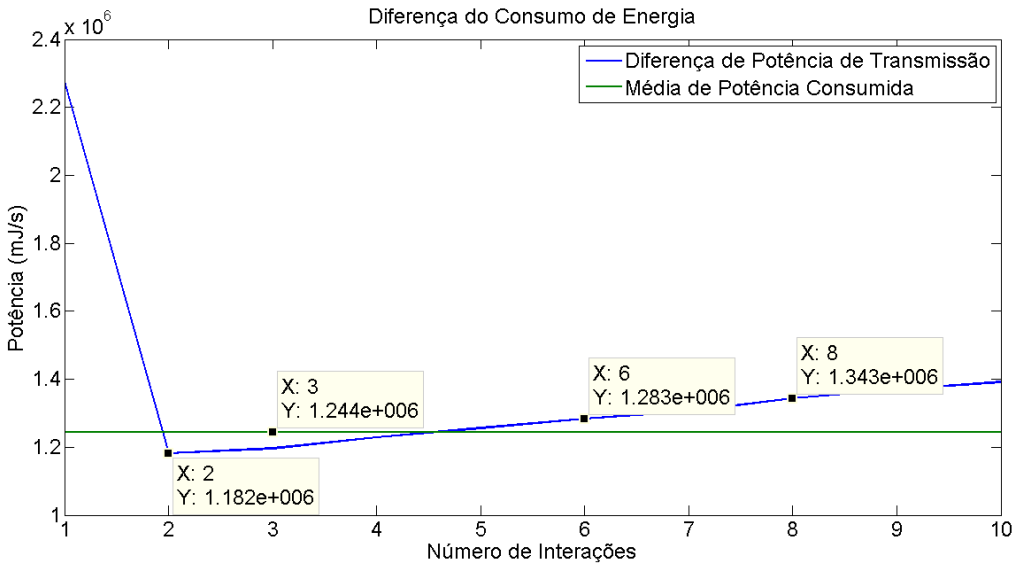

Figura 38 - Economia de Energia entre as Aplicações – 1100 nodos

A Figura 38 segue o mesmo padrão da economia de energia analisado nas simulações anteriores, note que esta diferença aumenta a medida que a diferença de redução de mensagens entre as aplicações também aumenta. Obviamente se a rede envia um número menor de pacotes, possui um gasto de energia menor. Para esta simulação a topologia padrão obteve um gasto de energia médio por nodo de 2030,91mW, enquanto a topologia padrão conseguiu para o mesmo ambiente um consumo médio por nodo de 720mW.

Utilizando-se a média de economia de potência obtida durante as 10 interações simuladas, pode-se calcular o impacto desta redução no aumento de vida útil dos nodos. Se o valor de potência médio economizado pela topologia híbrida foi de 1244000 mJ/s, ou 1130,9 mJ/s por nodo, em uma bateria de 2000 mAh e 2,7V, teremos um aumento médio de 4,84h por nodo da rede em apenas 10 interações, ou seja, um aumento na vida útil de aproximadamente 180%.

Consumo médio de um nodo na Topologia Padrão:

$$\frac{2030{,}91mW}{2{,}7V} = 752{,}18mA$$

$$\frac{2000mAh}{752{,}18mA} = 2{,}65h$$

Consumo médio de um nodo na Topologia Híbrida:

$$\frac{720mW}{2{,}7V} = 266{,}6mA$$

$$\frac{2000mAh}{266{,}6mA} = 7{,}5h$$

Vida útil ganha:

$$7{,}5h - 2{,}65h = 4{,}84h$$

Em comparação com as simulações anteriores, nota-se que a economia no número de mensagens transmitidas pela topologia híbrida é cada vez maior para redes com um maior número de nodos, o que representa um gasto de energia média por nodo cada vez menor e consequentemente uma vida útil ganha cada vez maior. Para esta simulação com 1100 nodos,

a vida útil ganha por nodo em caso de utilização da topologia híbrida chegou a 180% em relação a topologia padrão.

O aumento no número de mensagens transmitidas na rede, mesmo com a implementação da topologia híbrida, ocasiona um gasto de energia maior da rede se comparado a redes pequenas que trocam poucas informações. Porém a média de mensagens transmitidas por nodo é consideravelmente menor, o que ocasiona uma maior economia média de bateria.

De modo geral, as análises comprovam a melhora de desempenho proporcionada pelo algoritmo hibrido. A integração do detector de defeitos com o algoritmo de localização possibilita a otimização do conteúdo das mensagens transmitidas pelos nodos, diminuindo o número de mensagens transmitidas na rede e consequentemente o consumo de energia, aumentando a vida útil dos nodos. Além disto, a diminuição no número de mensagens trocadas evita possíveis congestionamentos e perdas de mensagens na rede.

6. CONCLUSÕES

Buscando uma nova metodologia para redes de sensores móveis, que aperfeiçoasse a comunicação entre os nodos, esta dissertação explorou uma possível integração entre o algoritmo de localização MCL (HU; EVANS, 2004) e o detector de defeitos Friedman (FRIEDMAN, 2005).

Este trabalho também vai ao encontro a esta linha de pesquisa, tentando atuar diretamente aonde a rede é mais sensível, na redução do consumo de energia através da diminuição das mensagens transmitidas pelos nodos. Esta diminuição nas trocas de mensagens ocorre devido a integração de detectores de defeitos e algoritmos de localização.

A topologia hibrida proposta nesta dissertação buscou realizar esta integração de arquiteturas, aonde o detector de defeitos foi implementado dentro do algoritmo de localização, tomando vantagem das mensagens já transmitidas por este algoritmo para enviar e receber informações de suspeitas de falhas. Apesar das limitações que algoritmos distintos possuem ao trabalhar em conjunto, conseguiu-se validar o modelo em um ambiente totalmente móvel e aleatório.

Após os testes realizados e descritos no capítulo anterior, se tornam claro os benefícios desta integração. Os cenários simulados comprovam que a diferença no número de mensagens enviadas em uma rede convencional que possua os modelos MCL e Friedman não integrados em comparação com a metodologia hibrida com os mesmos algoritmos integrados, pode passar de 30% do total de mensagens, variando de acordo com o tamanho da rede. Esta diferença percentual aumenta de acordo com o número de nodos na rede.

Além de evitar possíveis congestionamentos e perdas de mensagens na rede, o que acaba prejudicando a qualidade de serviço da mesma, a topologia hibrida também retira o detector de defeitos da camada de aplicação, integrando-o dentro do algoritmo de localização.

Por fim, a topologia também proporciona uma economia considerável no consumo de energia de cada nodo. A maior fonte de consumo da rede é o rádio, que é responsável pelo envio e recebimento de mensagens, a transmissão é a função que mais consome energia na rede e propostas que busquem a redução do número de mensagens na rede otimizam diretamente a autonomia dos nodos. Durante as simulações realizadas esta economia de energia proporcionou um aumento de vida útil dos nodos que variou de 5,8% até 180%, de acordo com o tamanho da rede e a bateria utilizada (2000mAh, 2,7V).

Finalmente pode-se afirmar que o presente trabalho apresentou uma solução inovadora para a redução do consumo e do número de mensagens em uma rede sem fio.

6.1. Trabalhos Futuros

Os benefícios da otimização do conteúdo das mensagens trocadas entre os nodos são diversos e uma integração entre diferentes algoritmos possibilita também que novos trabalhos neste sentido sejam desenvolvidos, como por exemplo, um detector de defeitos que se utiliza das coordenadas dos nodos para traçar possíveis trajetórias. O desenvolvimento desta integração em hardware também pode ser uma grande vantagem competitiva para a rede, aproveitando-se de que atualmente alguns algoritmos de localização já estão sendo implementados nesta camada.

7. REFERÊNCIAS

AGUILERA, M.K., CHEN, W., and TOUEG, S., **Heartbeat: A timeout-free failure detector for quiescent reliable communication**. In Workshop on Distributed Algorithms, pages 126-140, 1997.

AKYILDIZ, I., SU, W., SANKARASUBRAMANIAM, Y., and CAYIRCI, E. (2002). **A Survey on Sensor Networks,** IEEE Communications Magazine, Agosto, 2002.

BAGGIO, A.; LANGENDOEN, K. **Monte-carlo localization for mobile wireless sensor networks**. MSN 2006, p. 317–328, 2006.

BAGGIO, M. A. **Atento: Um detector de defeitos para MANETS baseado na potência do sinal.** Dissertação (Mestrado em Ciência da Computação). Programa de Pós Graduação em Informática, Universidade Federal de Santa Maria, Santa Maria. 2010.

BULUSU, N.; HEIDEMANN, J.; ESTRIN, D. **Gps-less low cost outdoor localization for very small devices**. IEEE Personal Communications Magazine, 2000.

CAMP, T.; BOLENG, J.; DAVIES, V. A **survey of mobility models for ad hoc network research**. *Wireless Communications & Mobile Computing*, v. 2, n. 5, 2002.

CHANDRA, T.D.; TOUEG, S. **Unreliable failure detectors for reliable distributed systems.** Journal of the ACM, [S.l.], v.43, n.2, p.225-267, Jan 1996.

DELLAERT, F.; FOX, D.; BURGARD, W.; THRUN, S. **Monte Carlo Localization for Mobile Robots**. IEEE International Conference on Robotics and Automation (ICRA). May 1999.

FELBER, P.; DÉFAGO, X.; GUERRAOUI, R.; OSER,P. **Failure Detectors as First Class Objects.** In: INTL. SYMP. ON DISTRIBUTED OBJECTS AND APPLICATIONS, DOA, 1999. Proceedings [S.l.]: IEEE Computer Society, 1999.

FISCHER, M. J.; LYNCH, N. A.; and PATERSON, M. S. **Impossibility of distributed consensus with one faulty process**. Journal of ACM, 32:374–382, 1985.

FRIEDMAN, R.; TCHARNY, G. **Evaluating failure detection in mobile ad-hoc networks.** Int. Journal of Wireless and Mobile Computing, 2005.

GARTNER, F.C. **Fundamentals of Fault-Tolerant Distributed Computing in Asynchronous Environments.** ACM Computing Surveys, [S.l.], v.31, n.1, Mar. 1999.

GETTING, I. **Perspective/navigation-the global positioning system**. *IEEE Spectrum*, v. 30, n. 12, p. 36–38,43–47, 1993.

GILAT, A. "**Matlab Com Aplicações em Engenharia**". Editora Bookman, 2°Edição, 2006.

GRACIOLI, G.; Nunes, R.C., **Detectores de Defeitos para Redes Wireless Ad Hoc**, IX Escola Regional de Redes de Computadores, Passo Fundo/RS, 2006.

HU, L.; EVANS, D. **Localization for mobile sensor networks**. In Tenth Annual International Conference on Mobile Computing and Networking, 2004.

HUTLE, M. **An efficient failure detector for sparsely connected networks**. Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN 2004), Innsbruck, Austria, 2004.

JALOTE, P. **Fault tolerance in distributed systems.** Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1994.

KAPLAN, E. D. ***Understanding GPS: principles and applications***. [S.l.]: Boston, Artech House, 1996.

KRIZMAN, K.; BIEDKA, T.; RAPPAPORT, T. **Wireless position location: Fundamentals, implementation strategies, and sources of error**. IEEE 47th Vehicular Technology Conference, v. 2, p. 919–923, 1997.

KONG, A.; LIU, J. S.; WONG, W. H. **Sequential Imputations and Bayesian Missing Data Problems**. *Journal of the American Statistical Association.* Vol. 89, pp 278-288. 1994.

LAMPORT, L. **Distribution** (sobre sistemas distribuídos) [Mensagem enviada ao DEC SCR bulletin board em 12:23:29 PDT on 28 May 1987] disponível em http://research.microsoft.com/users/lamport/pubs/pubs.html, acesso em janeiro de 2011.

MONDINELLI, F.; KOVACS-VAJNA, Z. **Self-localizing sensor network architectures.** *IEEE Transactions on Instrumentation and Measurement*, v. 53,n. 2, p. 277–283, 2004.

NICULESCU, D.; NATH, B. **Ad hoc positioning system (aps) using aoa**. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies - INFOCOM, v. 3, p. 1734–1743, 2003.

NICULESCU, D.; NATH, B. **Ad hoc positioning system (aps).** IEEE Global Telecommunications Conference - GLOBECOM, v. 5, p. 2926–2931, 2001.

NUNES, R. C. **Adaptação Dinâmica do Timeout de Detectores de Defeitos Através do Uso de séries Temporais**. Tese (Doutorado em Ciência da Computação). Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre. 2003.

OH-HEUM, K.; HA-JOO, S. **Localization through map stitching in wireless sensor networks.** *IEEE Transactions on Parallel and Distributed Systems: Accepted for future publication*, 2007.

OLIVEIRA, L. L. **Algoritmo de Localização de Nodos para Redes de Sensores Móveis.** Tese (Doutorado em Engenharia Elétrica). Programa de Pós Graduação em Engenharia Elétrica, Universidade Federal de Santa Maria, Santa Maria. 2009.

OLIVEIRA, L. L.; DESSBESELL, G. F.; MARTINS, J. B. S.; MONTEIRO, J. C. **Hardware Implementation of a Centroid-based Localization Algorithm for Mobile Sensor**

Networks. In: IEEE International Symposium on Circuits and Systems (ISCAS), 2011, Rio de Janeiro. IEEE International Symposium on Circuits and Systems (ISCAS), 2011.

PANTAZIS, N.; VERGADOS, D**. A survey on power control issues in wireless sensor networks**. *IEEE Communications Surveys and Tutorials*, v. 9, n. 4, p. 86–107, 2007. PARKINSON, B.; GILBERT, S. Navstar: Global positioning system - ten years later. *Proceedings of the IEEE*, v. 71, n. 10, p. 1177–1186, 1983.

PATWARI, N.; III, A. O. H. **Using proximity and quantized rss for sensor localization in wireless networks**. IEEE ACM 2nd Workshop on Wireless Sensor Networks and Applications, 2003

PEREIRA, M. R., AMORIM, C. L. e CASTRO, M. C. S. **Tutorial sobre redes de sensores**. 2003

POWELL, D. **Distributed Fault-Tolerance: A short Tutorial**. In: INTERNATIONAL WORKSHOP ON DEPENDABLE COMPUTING AND ITS APPLICATIONS, 1998, Johannesburg – South Africa. Proceedings…[S.l.: s.n.], 1998.

RENESSE, R. V., MINSKY, Y., and HAYDEN, M. **A gossip-style failure detection service**. Technical Report TR98-1687, 1998.

SILVA, F. L.; FERREIRA, L. F.; RIZZETTI, T. A.; et all. **Análise Comparativa de Detectores de Falhas para Redes Móveis.** VI Simpósio de Informática de Região Centro do RS – SIRC/RS 2007. (ISBN 978-85-88667-76-1, Santa Maria, RS).

SILVA, I. M. D. **Análise de Desempenho de Sistemas de Comunicação Sem-Fio para Monitoramento de Unidade de Produção de Poços Petrolíferos Terrestres.** Dissertação (Mestrado em Engenharia da Computação). Programa de Pós Graduação em Engenharia Elétrica e de Computação, Universidade Federal do Rio Grande do Norte, Natal. 2008.

SOUSA, M. P. **Diversidade Cooperativa Adaptativa Aplicada a Redes de Sensores sem Fio.** Dissertação (Mestrado em Engenharia da Computação). Programa de Pós Graduação em Engenharia Elétrica, Universidade Federal de Campina Grande, Campina Grande. 2009.

SPENCER, S. J. **The two-dimensional source location problem for time differences of arrival at minimal element monitoring arrays**. The Journal of the Acoustical Society of America, v. 121, n. 6, p. 3579–3594, 2007.

SRIDHAR, N. **Decentralized Local Failure Detection in Dynamic Distributed Systems**. Proceedings of the 25th IEEE Symposium on Reliable Distributed Systems (SRDS'06). (pp. 143 - 154). Washington: IEEE Computer Society, 2006.

TAI, A. T.; TSO, K. S. **Failure detection service for ad hoc wireless networks applications: a cluster-based approach.** [S.l.: s.n.], 2004. (IAT-302184, IA Tech, Inc., Los Angeles, CA).

TAI, A. T.; TSO, K. S.; SANDERS, W. H. **Cluster-Based Failure Detection Service for Large-Scale Ad Hoc Wireless Network Applications**. In: DSN ’04: PROCEEDINGS OF

THE 2004 INTERNATIONAL CONFERENCE ON DEPENDABLE SYSTEMS AND NETWORKS (DSN'04), 2004, Washington, DC, USA. **Anais. . .** IEEE Computer Society, 2004. p.805.

TAN, H. X. **Quality of Service in Wireless Sensor Networks**, Final Report submitted in CS6204 – AY 2005/06 Semester II

TANENBAUM, A S. **Distributed operating systems**. New Jersey: Prentice Hall, 1995. 614 p. ISBN 0132199084

THRUN, S.; FOX, D.; BURGARD, W.; DELLAERT, F. **Robust Monte Carlo Localization for Mobile Robots**. Artificial Intelligence Journal. 2001.

TILAK, S.; ABU-HHAZELEH, N. B.; and HEINZELMAN, W. **A taxonomy of wireless micro-sensor network models**. ACM SIGMOBILE Mobile Computing and Communications Review, 6(2):28–36, 2002.

TURCHETTI, R. C. **Uma nova abordagem para redução de mensagens de controle em detectores de defeitos.** Dissertação (Mestrado em Engenharia de Produção). Programa de Pós Graduação em Engenharia de Produção, Universidade Federal de Santa Maria, Santa Maria, 2006.

VIVEKANANDAN, V.; WONG, V. **Concentric anchor-beacons (cab) localization for wireless sensor networks**. IEEE International Conference on Communications, v. 9, p. 3972–3977, 2006.

WEBER, T.S. **Tolerância a Falhas: Conceitos e Exemplos.** http://www.inf.ufrgs.br/~taisy/disciplinas/textos/Dependabilidade.pdf - último acesso em Janeiro de 2012.