

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

ALGORITMO DE LOCALIZAÇÃO DE NODOS
PARA REDES DE SENSORES MÓVEIS

TESE DE DOUTORADO

Leonardo Londero de Oliveira

Santa Maria, RS, Brasil

2009

ALGORITMO DE LOCALIZAÇÃO DE NODOS PARA REDES DE SENSORES MÓVEIS

por

Leonardo Londero de Oliveira

Tese apresentada ao Curso de Doutorado do Programa de Pós-Graduação em Engenharia Elétrica, Área de Concentração em Processamento de Energia, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de
Doutor em Engenharia Elétrica.

Orientador: Prof. Dr. João Baptista dos Santos Martins
Co-orientador: Prof. Dr. José Carlos Alves Pereira Monteiro

Santa Maria, RS, Brasil

2009

Dados Internacionais de Catalogação na Publicação (CIP)

O48a Oliveira, Leonardo Londero de
Algoritmo de localização de nodos para redes de sensores móveis . /
Leonardo Londero de Oliveira – Santa Maria , 2009.
136 f. ; 30 cm.

Tese (Doutorado) – Programa de Pós Graduação em Engenharia
Elétrica – Universidade Federal de Santa Maria, 2009.
“Orientação: Prof. Dr. João Baptista dos Santos Martins”.
“Co-orientação: Prof. Dr. José Carlos Alves Pereira Monteiro”.

1. Redes de Sensores Móveis. 2. Algoritmo de Localização.
3. ASIC. I. Martins, João Baptista dos Santos. II. Monteiro, José Carlos
Alves Pereira. III. Título.

CDU: 621.38(043)

Bibliotecária Responsável: Lizandra Veleda Arabidian CRB/10-1492

©2009

Todos os direitos autorais reservados a Leonardo Londero de Oliveira. A reprodução de partes ou do todo deste trabalho só poderá ser feita com autorização por escrito do autor.

Endereço: Avenida Roraima, 1000, Centro de Tecnologia, Grupo de Microeletrônica,
Anexo B, Bairro Camobi, Santa Maria, RS, 97105-900

Fone: +55 55 32209423;

Endereço Eletrônico: leonardo@mail.ufsm.br, www.ufsm.br/gmicro

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

A Comissão Examinadora, abaixo assinada,
aprova a Tese de Doutorado

**ALGORITMO DE LOCALIZAÇÃO DE NODOS PARA REDES
DE SENSORES MÓVEIS**

elaborada por
Leonardo Londero de Oliveira

como requisito parcial para obtenção do grau de
Doutor em Engenharia Elétrica

COMISSÃO EXAMINADORA:

João Baptista dos Santos Martins, Dr.
(Presidente/Orientador)

José Carlos Alves Pereira Monteiro, Dr.
(Co-orientador)

Ricardo Augusto da Luz Reis, Dr. (UFRGS)

Eduardo Antonio César da Costa, Dr. (UCPel)

Alexandre Campos, Ph.D. (UFMS)

Raul Ceretta Nunes, Dr. (UFMS)

Santa Maria, 21 de Dezembro de 2009

À minha esposa Olga e ao meu filho Mateus

AGRADECIMENTOS

Ao fim desta importante etapa da minha vida construí uma valiosa carga que levarei para o resto dos meus dias. Esta carga não é formada apenas pelo conhecimento técnico-científico acumulado em todos estes anos.

O Grupo de Microeletrônica da UFSM proporcionou o convívio com diversas pessoas, sejam elas profissionais da indústria, estudantes ou os doutores do seu quadro docente. Este Grupo extrapola o caráter exclusivamente teórico e, produz, além do conhecimento técnico-científico, a cooperação com o meio privado, essencial em um país que precisa aplicar na prática a tecnologia aprendida na teoria.

Quero agradecer em especial aos meus orientadores João Baptista dos Santos Martins e José Monteiro por acreditarem no meu trabalho e trilharem este caminho junto comigo. Aos colegas Fernando Herrmann, Gustavo Dessbesell e Guilherme Perin pelas cooperação técnica e agradável convivência. Aos amigos Paulo, Leonardo e Diego Marquezini, fundamentais na minha estada em Porto Alegre.

Agradeço ao CNPq, a CAPES e ao Grupo Algos (INESC-ID, Lisboa) por financiarem este trabalho e pelo valioso suporte recebido durante a minha estada em Portugal.

Por fim, agradeço à minha Família, em especial aos meus Pais, Dioní Jorge Costa de Oliveira e Maria Oneida Londero de Oliveira por me darem todas as condições de que precisei para chegar onde estou. Agradeço a Deus por me presentear com a minha esposa Olga e o meu filho Mateus, com os quais multiplico a minha felicidade a cada dia da minha vida.

“ The most exciting phrase to hear in science, the one that heralds the most discoveries, is not "Eureka!"(I found it!) but "That's funny".”

Isaac Asimov ★ 1920 † 1992

RESUMO

Tese de Doutorado

Programa de Pós-Graduação em Engenharia Elétrica

Universidade Federal de Santa Maria, RS, Brasil

ALGORITMO DE LOCALIZAÇÃO DE NODOS PARA REDES DE SENSORES MÓVEIS

AUTOR: LEONARDO LONDERO DE OLIVEIRA

ORIENTADOR: JOÃO BAPTISTA DOS SANTOS MARTINS

CO-ORIENTADOR: JOSÉ CARLOS ALVES PEREIRA MONTEIRO

Local da Defesa e Data: Santa Maria, 21 de Dezembro de 2009.

Esta Tese apresenta contribuições ao processo de localização de nodos em redes de sensores móveis. Considerando a importância de algoritmos de localização em uma rede para identificar o local do evento sob estudo e determinar a ação a ser tomada pelo usuário, é desenvolvido um novo algoritmo de localização para operar em redes de sensores móveis, focando o aspecto da mobilidade não controlada dos nodos. Este algoritmo, ao qual denominamos CentroidM, teve como base de desenvolvimento o método Centroid. Foram mantidas as características positivas do Centroid e exploradas as limitações do método para a sua execução em uma rede que considera o movimento. Além do objetivo em reduzir o erro das estimativas de posicionamento calculadas pelo algoritmo, o consumo de potência do método apresentado neste trabalho é enfatizado. O projeto de um circuito integrado dedicado que desempenhe o processo de localização em uma rede móvel é justificado pela intensa utilização de tal subsistema neste tipo de rede, além dos benefícios que um *hardware* dedicado traz face à redução da energia gasta nesta operação. Desta forma, o foco deste trabalho recai sobre um algoritmo de localização para redes móveis e as características desta abordagem que permitiram aumentar a precisão das estimativas de posicionamento tanto em um cenário com ausência de movimento dos nodos, quanto em outro onde o aspecto da mobilidade sem atuadores foi considerada. Os resultados obtidos nesta Tese demonstraram que o CentroidM é 30% mais preciso que o Centroid para um cenário estático e 40% considerando a mobilidade da rede. O ganho alcançado em precisão teve um custo computacional que elevou em 2,76 vezes o tempo gasto pelo CentroidM para realizar um procedimento de localização. Contudo, os resultados de simulação mostraram que é possível eliminar a sobrecarga computacional e ainda assim atingir ganhos em precisão próximos a 10%. O desenvolvimento do método de localização é complementado pela sua implementação em um circuito integrado dedicado, reduzindo a energia gasta no processo de estimativa da posição em uma ordem da magnitude face à execução do algoritmo em um microcontrolador comercial de baixo consumo.

Palavras-chave: redes de sensores móveis; algoritmo de localização; ASIC

ABSTRACT

Doctoral Thesis

Post-Graduate Program in Electrical Engineering

Federal University of Santa Maria, RS, Brazil

NODE LOCALIZATION ALGORITHM TO MOBILE SENSOR NETWORKS

AUTHOR: LEONARDO LONDERO DE OLIVEIRA

ADVISOR: JOÃO BAPTISTA DOS SANTOS MARTINS

CO-ADVISOR: JOSÉ CARLOS ALVES PEREIRA MONTEIRO

Place and Date: Santa Maria, December 21st, 2009.

This thesis presents contributions to node localization in mobile sensor networks. Considering the importance of localization algorithms in identifying the location of an event in order to better determine the action to be taken by the user, a new localization algorithm to operate in mobile networks was designed, assuming an uncontrollable movement of the nodes. This algorithm, which we call CentroidM, has the Centroid method as a stand. Positive features of the Centroid algorithm were kept while their limitations due to the dynamic characteristics of the network movement were mitigated. Besides the concern regarding the accuracy of the method, the power consumption of the algorithm was addressed too. The design of an dedicated integrated circuit to deal with the localization task in a mobile network is justified given the high activity of such a subsystem in the given scenario. This way, the focus of this Thesis relies on a low power localization algorithm for mobile networks, with characteristics to allow raising calculated coordinates accuracy in both static and mobile scenarios. The results show that the CentroidM is 30% more accurate than the Centroid on a static scenario, and 40% when we consider a node movement without actuators. These benefits have some computational overhead, increasing 2.76 times the time spent by the CentroidM to run a localization process. However, simulation results showed it is possible to remove such overhead and still keep the achieved estimation gains near 10%. The implementation of the localization method was accomplished through an integrated circuit, which reduced the energy expended during the computation of the coordinates by an order of magnitude when comparing with the execution of the algorithm in a low power commercial microcontroller.

Keywords: mobile sensor networks; localization algorithm; ASIC

SUMÁRIO

Agradecimentos

Resumo

Abstract

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

1	Introdução	p. 19
1.1	Localização	p. 19
1.2	Motivação	p. 21
1.3	Contribuições	p. 23
1.4	Organização da Tese	p. 23
2	Classificação dos Métodos de Localização em Redes de Sensores	p. 25
2.1	Rede de Sensores	p. 25
2.2	Classificação	p. 28
2.2.1	Sistema de Posicionamento Global	p. 29
2.2.2	Métodos Baseados na Medição da Distância - <i>Range-based</i> . . .	p. 32
2.2.2.1	Tempo de Chegada - <i>Time of Arrival</i>	p. 33
2.2.2.2	Ângulo de Chegada - <i>Angle of Arrival</i>	p. 34

2.2.2.3	Tempo de Diferença de Chegada - <i>Time Difference of Arrival</i>	p. 34
2.2.2.4	Indicação de Força do Sinal Recebido - <i>Received Signal Strength Indicator</i>	p. 36
2.2.2.5	Discussão Acerca dos Métodos	p. 37
2.2.3	Métodos Baseados na Conectividade - <i>Range-free</i>	p. 38
2.2.4	Mobilidade	p. 41
2.3	Resumo	p. 44
3	Revisão de Algoritmos de Localização	p. 45
3.1	Coordenadas Absolutas e Relativas	p. 45
3.2	Redes Móveis com Atuadores para o Movimento	p. 47
3.2.1	Eleição de um Líder	p. 50
3.3	Método Monte Carlo	p. 51
3.4	Mobilidade Aleatória	p. 55
3.5	Resumo	p. 58
4	Algoritmo Proposto	p. 59
4.1	Algoritmo Centroid	p. 59
4.2	CentroidM	p. 63
4.2.1	Limitações do Centroid	p. 63
4.2.2	Alterações Propostas ao Algoritmo Centroid	p. 66
4.3	Resumo	p. 73
5	Ambiente de Desenvolvimento e Metodologia	p. 74
5.1	<i>Software</i>	p. 74
5.2	TinyOS e nesC	p. 75
5.3	TOSSIM	p. 77
5.4	Mobilidade	p. 79

5.4.1	Mobilidade de Percurso Aleatório	p. 82
5.4.2	Mobilidade de Percurso Aleatório com Pausas	p. 82
5.5	Avrora	p. 84
5.6	Resumo	p. 85
6	Circuito Dedicado do Algoritmo Proposto	p. 86
6.1	Fluxo Digital do Circuito Integrado	p. 87
6.1.1	Síntese Lógica do CentroidM	p. 90
6.1.2	Síntese Física do CentroidM	p. 91
6.2	Descrição das Etapas de Verificação e Avaliação dos Algoritmos Centroid e CentroidM	p. 92
6.3	Arquitetura	p. 95
6.4	Resumo	p. 99
7	Resultados Obtidos e Análise de Desempenho	p. 100
7.1	Topologia Estática	p. 101
7.2	Estudo da Mobilidade	p. 109
7.2.1	Mobilidade Aleatória	p. 110
7.2.2	Mobilidade Aleatória com Inserção de Pausas	p. 112
7.3	Equilíbrio do Consumo de Energia: CentroidM versus Centroid	p. 117
7.4	Versão em <i>Hardware</i> do CentroidM	p. 118
7.4.1	Implementação em Alto Nível	p. 118
7.4.2	Implementação Física do CentroidM	p. 120
7.5	Resumo	p. 123
8	Conclusões	p. 125
	Referências	p. 128

LISTA DE FIGURAS

Figura 1	Rede de sensores sem fio evidenciando os nodos de referência. . . .	p. 26
Figura 2	Componentes de um nó sensor.	p. 27
Figura 3	Fluxo simplificado para a localização utilizando o sistema GPS. . .	p. 30
Figura 4	Interpretação geométrica da trilateração.	p. 33
Figura 5	Interpretação geométrica do ângulo de chegada.	p. 34
Figura 6	Interpretação geométrica do tempo de diferença de chegada. . . .	p. 35
Figura 7	Algoritmo CAB.	p. 39
Figura 8	Composição das distâncias euclidianas no método DV-HOP (NICULESCU; NATH, 2001). Figura adaptada de (NICULESCU; NATH, 2001).	p. 40
Figura 9	Fonte de erro do algoritmo DV-Hop (adaptado de (TIAN et al., 2007)).	p. 40
Figura 10	Exemplo abstrato de um movimento aleatório de nodos relativo a animais em uma fazenda.	p. 42
Figura 11	Um cenário onde os nodos âncora são robôs móveis e controlam seus movimentos para percorrer um determinado caminho e localizar nodos estáticos (adaptado de (SHENOY; TAN, 2005)).	p. 43
Figura 12	Exemplo abstrato da escolha de um líder, baseada na programação do algoritmo em reconhecer nodos com maior quantidade de informação. .	p. 51
Figura 13	Diferentes regiões de localização baseadas na granularidade. Adaptado de (BULUSU; HEIDEMANN; ESTRIN, 2000).	p. 62
Figura 14	Cenário de topologia de rede indicando a mobilidade de um nó em diferentes momentos.	p. 65
Figura 15	Exemplo do impacto das métricas de conectividade ponderadas no cálculo da estimativa.	p. 70

Figura 16	Fluxo utilizado na etapa de validação funcional.	p. 75
Figura 17	Esquema simplificado da arquitetura TOSSIM (adaptado de (LEVIS et al., 2003)).	p. 78
Figura 18	Biblioteca Meyer na Universidade de Stanford, representando um ambiente com muitas fontes de ruído.	p. 81
Figura 19	Lago artificial Lagunita na Universidade de Stanford, representando um ambiente com pouco ruído.	p. 82
Figura 20	Exemplo de movimento de nodos utilizando o <i>Random Walk Mobility Model</i>	p. 83
Figura 21	Exemplo de movimento de nodos utilizando o <i>Random Waypoint Mobility Model</i>	p. 84
Figura 22	Nó sensor evidenciando o subsistema de localização.	p. 87
Figura 23	Fluxo digital para o projeto do CentroidM: etapa pré-leiaute.	p. 88
Figura 24	Fluxo de síntese física empregado para o CentroidM.	p. 89
Figura 25	Esquema de verificação do circuito dedicado.	p. 93
Figura 26	Diagrama de topo simplificado para o CentroidM.	p. 96
Figura 27	Diagrama de blocos do sistema de localização.	p. 96
Figura 28	Diagrama do processo de execução das máquinas de estados do circuito dedicado.	p. 98
Figura 29	Representação de uma distribuição aleatória de nodos estáticos a serem localizados. Os pontos envoltos por hexágonos representam os nodos âncora da rede.	p. 102
Figura 30	Simulação da rede estática descrita na Figura 29.	p. 103
Figura 31	Erro para o nó envolto por uma circunferência na Figura 29 em um cenário com ausência de movimento.	p. 104
Figura 32	Erro para uma simulação envolvendo uma área de $50m \times 50m$	p. 106
Figura 33	Erro para um único nó da simulação mostrada na Figura 32 ao longo de 500 segundos.	p. 107

Figura 34	Erro de localização versus a posição para o algoritmo Centroid (BULUSU; HEIDEMANN; ESTRIN, 2000): mapeamento topológico do erro.	p. 108
Figura 35	Erro de localização versus a posição para o trabalho proposto nesta Tese: mapeamento topológico do erro.	p. 108
Figura 36	Impacto da mobilidade sobre o erro calculado para o <i>Random Walk Mobility Model</i>	p. 111
Figura 37	Impacto do ruído - <i>Random Walk Mobility Model</i>	p. 112
Figura 38	Impacto da mobilidade para o <i>Random Waypoint Mobility Model</i>	p. 113
Figura 39	Impacto do ruído - <i>Random Waypoint Mobility Model</i>	p. 113
Figura 40	Erro de localização referente ao algoritmo CentroidM para o movimento de caminhada aleatória do nó número 11 descrito na topologia da Figura 21.	p. 116
Figura 41	Erro de localização referente ao algoritmo Centroid para o movimento de caminhada aleatória do nó número 11 descrito na topologia da Figura 21.	p. 116
Figura 42	Impacto da redução de mensagens recebidas pelos nodos e aumento do erro para o CentroidM.	p. 118
Figura 43	Dados visuais de entrada e saída da implementação em <i>hardware</i> do CentroidM para uma simulação de 120 segundos.	p. 120
Figura 44	Leiaute do circuito dedicado em tecnologia $0,13\mu$ com uma área estimada em $0,32mm^2$	p. 122

LISTA DE TABELAS

Tabela 1	Cenários de mobilidade de acordo com o tipo de movimento de nodos âncora e nodos comuns.	p. 42
Tabela 2	Algoritmos de localização de nodos em redes de sensores.	p. 46
Tabela 3	Exemplo do impacto de CM_i e μCMp_i , relativo ao cenário apresentado nas Figuras 15(a) e 15(b), para a seleção dos nodos âncora utilizados no cálculo das coordenadas.	p. 70
Tabela 4	Primeiros pontos do gráfico mostrado na Figura 31.	p. 105
Tabela 5	Erro médio de todos os nodos envolvidos no cenário - impacto do aumento de área	p. 107
Tabela 6	Impacto da mobilidade sobre o erro calculado considerando o <i>Random Walk Mobility Model</i>	p. 110
Tabela 7	Impacto sobre o erro calculado considerando um ambiente ruidoso.	p. 111
Tabela 8	Valores de erro para o nó número 11 da Figura 21.	p. 114
Tabela 9	Centroid versus CentroidM - implementação em <i>software</i>	p. 117
Tabela 10	Dados para o CentroidM - primeira abordagem.	p. 121
Tabela 11	Dados para o CentroidM - segunda abordagem.	p. 121
Tabela 12	Sumário do sistema de localização integrado para redes de sensores móveis.	p. 123

LISTA DE ABREVIATURAS

ADC	<i>Analog-to-Digital Converter</i>
AOA	<i>Angle of Arrival</i>
APIT	<i>Approximate Point in Triangle</i>
ASIC	<i>Application Specific Integrated Circuit</i>
CAB	<i>Concentric Anchor Beacon</i>
CMOS	<i>Complementary Metal Oxide Semiconductor</i>
EL	<i>Erro de Localização</i>
FPGA	<i>Field Programmable Gate Array</i>
FSM	<i>Finite State Machines</i>
GDSII	<i>Graphic Data System II</i>
GPS	<i>Global Positioning System</i>
HDL	<i>Hardware Description Language</i>
IEEE	<i>Institute of Electrical and Electronic Engineers</i>
IP core	<i>Intellectual Property core</i>
MCL	<i>Monte Carlo Localization</i>
NAVSTAR	<i>Navigation Satellite Timing and Ranging</i>
NLOS	<i>Non-Line of Sight</i>
NMOS	<i>Negative Channel Metal Oxide Semiconductor</i>
PMOS	<i>Positive Channel Metal Oxide Semiconductor</i>
PRR	<i>Packet Reception Rate</i>

Lista de Abreviaturas

RF	<i>Radio Frequency</i>
RFID	<i>Radio Frequency Identification Device</i>
RFM	<i>Radio Frequency Module</i>
RSSI	<i>Received Signal Strength Indicator</i>
RTL	<i>Register Transfer Level</i>
RTT	<i>Round Trip Transmission</i>
SANLA	<i>Selective Anchor Localization Algorithm</i>
SDF	<i>Standard Delay Format</i>
SNR	<i>Signal to Noise Ratio</i>
TOA	<i>Time of Arrival</i>
TDOA	<i>Time Difference of Arrival</i>
VCD	<i>Value Change Dump</i>
VHDL	<i>VHSIC Hardware Description Language</i>

1 INTRODUÇÃO

Este Capítulo expõe a importância do processo de localização de nodos em uma rede de sensores. Os métodos de localização de nodos receberam a partir de 2004 especial atenção da comunidade científica para o cenário onde estes dispositivos estão movimentando-se aleatoriamente. Os desafios e possibilidades relativos a este aspecto são então discutidos. Identificam-se também as motivações e contribuições do trabalho desenvolvido nesta Tese¹, finalizando com a organização do texto.

1.1 Localização

Monitorar eventos ou processos é uma antiga preocupação do ser humano e, por vezes, um desafio imposto a cada aumento da complexidade da situação em estudo. Ao longo de sua história o homem tem empregado técnicas de monitoramento nas mais diversas áreas da ciência, desde a mais simples contagem diária do número de animais de um rebanho, até ínfimas medições sísmicas do comportamento de placas tectônicas.

As razões para este desejo de controlar também se estendem por uma quantidade de opções limitada apenas pela criatividade humana. De fato, à medida que cresce a necessidade de monitorar um número cada vez mais ampla de fenômenos, estas novas aplicações exigem pesquisas e soluções que possibilitem atender a esta demanda. Avanços científicos dos mais variados possibilitaram que as dimensões dos circuitos eletrônicos diminuíssem a níveis nanoscópicos. Este avanço na fabricação dos transistores por si só contribuiu muito para a redução do consumo de potência dos dispositivos eletrônicos de forma a possibilitar o avanço de um campo da ciência amplamente estudado nos dias de hoje: redes de sensores.

A tarefa localização de nodos em uma rede de sensores pode ser interpretada como o cálculo das coordenadas de uma posição em um dado sistema de coordenadas. Basicamente,

¹O texto descrito neste trabalho segue as orientações de configuração de documentos descritas em (UFSM, 2006), bem como a ortografia da Língua Portuguesa vigente em 2009.

mente 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.

A localização é importante para qualquer aplicação que necessite interpretação de dados em um contexto físico. Dados coletados do meio não possuem significado relevante caso seja impossível obter as coordenadas geográficas do fenômeno sob estudo (RABAEY et al., 2000). Tome-se, por exemplo, um cenário onde um grande número de sensores está distribuído na copa de árvores em uma floresta (TOLLE et al., 2005). Tais dispositivos poderiam estar encarregados de monitorar princípios de incêndio. Uma vez que o foco é detectado, alguma ação por parte da entidade responsável precisa ser tomada para lidar com esta situação. Contudo, caso não haja informação a respeito da posição do fenômeno, pouco ou nada poderá ser feito para contornar a situação e a detecção feita pela rede pode ser inútil.

Este cenário pode ser visualizado sob outro ponto de vista. Uma vez que os nodos estão dispostos em árvores, é factível pressupor que a rede é estática. Contudo, se os nodos são retirados das árvores e passam a fazer parte de um cenário onde, por exemplo, são carregados por animais selvagens perfeitamente integrados ao habitat do fenômeno sob estudo, a rede é dita móvel.

Nesta nova abordagem, os nodos da rede estão constantemente expostos a um movimento não controlado. Isto possibilita que a detecção de fumaça, por exemplo, não seja o único propósito, mas também o comportamento dos animais possa ser capturado. Desta forma, vislumbra-se diversas possibilidades relacionadas à aquisição de informação por parte de rede, introduzidas pela mobilidade de seus nodos (TOLLE et al., 2005).

Uma vez que a rede detecta um padrão de aglomeração incomum de animais, isto pode indicar que os mesmos estão fugindo de uma situação de perigo iminente. Além disso, caso os nodos diminuam a sua velocidade, pode-se inferir que estes estejam se alimentando ou, em caso da falta de movimento por um longo período de tempo, é factível pressupor um impedimento físico causado por um trauma diverso ou mesmo a morte. Em outras palavras, estes padrões de comportamento podem apontar para a existência de focos de incêndio, mas não apenas isto. Se a situação anterior for transportada de uma floresta para uma fazenda e considerar que os nodos da rede estão dispostos em um rebanho de gado, a rede pode detectar, pelo padrão de movimentos, por exemplo, um animal doente ou mesmo indicar abigeato.

Todos os nodos mencionados nos exemplos anteriores compartilham uma mesma restrição de projeto. Pelo fato de serem dispositivos autônomos, o quesito energia é um fator a ser considerado. Equipamentos portáteis operados por baterias, ou mesmo com a obtenção de energia do meio ambiente, sofrem uma pressão natural face à redução do consumo de potência (CHANDRAKASAN; BRODERSEN, 1995). Isto impacta diretamente no tempo de operação útil destes equipamentos (POLASTRE; SZEWCZYK; CULLER, 2005). O requerimento para uma rede de sensores é que seus elementos possam processar informações por um período de tempo aceitável sem que haja intervenção humana. Tal período pode variar desde dias, meses ou anos, dependendo do objetivo da rede. É desejável, portanto, que uma aplicação tenha seu consumo de potência reduzido sem que para tanto o seu desempenho seja comprometido.

Especialmente uma rede de sensores com muitos dispositivos impede economicamente a manutenção focada na troca da fonte de alimentação. Mesmo se forem considerados poucos nodos, a localização geográfica destes pode ser um obstáculo para esta tarefa (WERNER-ALLEN et al., 2005).

1.2 Motivação

Cenários como os mencionados na Seção 1.1 demandam diferentes abordagens para estimar a posição dos elementos da rede e, o processo de localização precisa ser ativado diversas vezes para acomodar o movimento dos nodos. Desta forma, frequentemente o pesquisador depara-se com situações onde os nodos se movem arbitrariamente de tal maneira que as interconexões entre os mesmos mudam continuamente e as estações base estão apenas ocasionalmente disponíveis.

Especialmente em rede de sensores móveis, a localização do dispositivo é um tópico interessante devido à sua relação com a distribuição da informação e o consumo de energia, uma vez que a mobilidade introduz uma componente em tempo real ao algoritmo de localização. Por outro lado, a descoberta da posição do nó em uma rede de sensores estática é realizada apenas uma vez, depois da alocação dos nodos em suas posições, onde o fato de despender alguns minutos calculando posições é perfeitamente aceitável. Mesmo considerando o caso onde durante a deposição dos nodos sensores em uma rede estática, os mesmos sejam alimentados com a informação de suas posições, para uma rede de sensores com muitos nodos esta abordagem pode ser economicamente inviável. Além disso, o procedimento de localização, seja ele automático ou manual, precisa ser repetido para acomodar situações onde ocorre destruição dos nodos ou reposição manual,

por exemplo, o que por sua vez aumenta a importância de um sistema de localização.

Grande parte dos algoritmos de localização que lidam com a questão da mobilidade foram desenvolvidos sob o ponto de vista da utilização de robôs, com toda infra-estrutura requerida para implementar a aplicação em questão. Contudo, existem diferenças substanciais (WU; SHENG; SONG, 2006) entre a localização com robôs e a localização sem atuadores envolvidos no movimento. A localização assistida por robôs usualmente assume que o integrante da rede possui um conhecimento prévio do mapa do terreno ou o constrói à medida que percorre a área sob estudo (LADD et al., 2005). Além disto, o processo de estimar as coordenadas do robô é frequentemente auxiliado pela informação de sensores indicando velocidade, direção e/ou sentido do movimento (SHENOY; TAN, 2005). Cada uma destas abordagens, ou seja, movimento com e sem atuadores, é apropriada para diferentes aplicações como poderá ser visto ao longo do Capítulo 3. Além disso, a ênfase da pesquisa em localização de nodos está quase sempre ligada exclusivamente à precisão do método, minimizando a preocupação com o aspecto do consumo de potência (AKYILDIZ et al., 2002).

A última década tem recebido especial atenção da comunidade científica acerca do processo de localização de nodos em redes de sensores, especialmente quanto ao caso onde os nodos não possuem atuadores para o movimento (HU; EVANS, 2004; BAGGIO; LANGENDOEN, 2006; HSIEH; WANG, 2006; STEVENS-NAVARRO; VIVEKANANDAN; WONG, 2007). Com a redução de custos e tamanho dos nodos sensores, é possível introduzir um número cada vez mais elevado destes dispositivos em uma rede. Este ponto em particular também fez com que novos problemas surgissem. Uma rede de sensores que outrora poderia ter a posição de seus dispositivos configurada manualmente passou a sofrer com a maior quantidade de elementos. Com isto, o processo de localização dos nodos tende a eliminar o aspecto da configuração manual, ao mesmo tempo em que faz emergir a importância de um bloco dedicado à tarefa de auto-localização como um elemento adicional ao nó sensor (AKYILDIZ et al., 2002).

O sistema de localização de um nó é amplamente utilizado em aplicações que empreguem a mobilidade da rede. Com base no descrito acima, o processo de localização é repetidamente ativado durante o ciclo de vida do nó sensor. Desta forma, é perfeitamente factível a escolha de um circuito dedicado com o intuito de redução da potência consumida. Um circuito dedicado que desempenhe esta função pode auxiliar na redução do consumo de potência envolvido com o processamento interno da informação, visto que o mesmo é otimizado para tal propósito. Além disto, a concepção de uma rede de sensores é extremamente orientada (NICULESCU, 2005) à aplicação na qual ela está inserida

e, portanto, o planejamento detalhado de seus componentes é um fator preponderante para o sucesso do monitoramento.

1.3 Contribuições

A partir da revisão de diversas alternativas em algoritmos de localização presentes na literatura e das modificações propostas sobre o método selecionado, pode-se elencar como principais contribuições desta Tese:

- a inclusão do aspecto da mobilidade de maneira não controlada a um algoritmo de localização² para redes estáticas³, com redução do erro de posicionamento;
- concepção de um circuito integrado dedicado para desempenhar a função de um subsistema de localização e reduzir o consumo;
- redução da energia despendida durante o procedimento de localização de nodos face a execução do algoritmo em um microcontrolador comercial de baixo consumo⁴;
- manutenção da flexibilidade do circuito dedicado pela disponibilização de parâmetros configuráveis.

1.4 Organização da Tese

O restante desta Tese será organizado como segue. O Capítulo 2 apresenta uma revisão sobre os conceitos básicos de redes de sensores. Este Capítulo também apresenta um estudo sobre os parâmetros a serem utilizados para classificar algoritmos de localização, com o foco na mobilidade dos nodos da rede.

No Capítulo 3 é feita uma revisão qualitativa das características abordadas no Capítulo 2, de forma que se possa obter uma percepção clara dos avanços produzidos nesta área. É dada ênfase aos algoritmos de localização que exploram o aspecto da mobilidade de maneira controlada e não controlada, bem como a maneira pela qual as distâncias são mensuradas: diretamente através de variáveis físicas ou considerando a conectividade da rede.

²O referido algoritmo é o Centroid (BULUSU; HEIDEMANN; ESTRIN, 2000) e será detalhado no Capítulo 4.

³Os elementos que constituem este tipo de rede são desprovidos de movimento.

⁴Microcontrolador fabricado pela empresa Atmel (ATMEL, 2009).

O Capítulo 4 apresenta o método Centroid original e suas características considerando um cenário de redes estáticas e móveis. Além disto, são expostas as razões para a escolha deste algoritmo como ponto de partida para a proposição de melhorias que englobem o aspecto da mobilidade. Este Capítulo também apresenta um novo algoritmo de localização de nodos em redes de sensores móveis para estimar as coordenadas de um nó desconhecido movimentando-se dentro de uma área e recebe mensagens de referência a partir de nodos âncora estáticos, dentro do alcance de rádio.

O Capítulo 5 apresenta a metodologia envolvida na validação funcional da proposta. Neste Capítulo são descritas as ferramentas utilizadas para a fase de reprodução do algoritmo Centroid e desenvolvimento do CentroidM em *software*.

O Capítulo 6 trata do fluxo utilizado para o projeto do circuito integrado e avaliação do consumo. Também neste Capítulo é apresentada a arquitetura que envolveu a construção do circuito dedicado que implementou o CentroidM descrito no Capítulo 4.

O Capítulo 7 reúne os resultados obtidos para em diversos cenários, ponderando acerca do desempenho do método face a ausência de movimento e mobilidade. Finalmente, as conclusões desta tese, bem como propostas para direções futuras e uma projeção dos próximos passos a serem tomados relacionadas com este trabalho são descritas no Capítulo 8.

2 CLASSIFICAÇÃO DOS MÉTODOS DE LOCALIZAÇÃO EM REDES DE SENSORES

A idéia de um algoritmo de localização em redes de sensores sem fio é situar no espaço as coordenadas, sejam relativas ou absolutas, dos nodos de uma rede, evitando portanto a necessidade de cada nó receber previamente a sua localização.

Neste capítulo será introduzida a teoria relacionada a redes de sensores. Logo após, uma classificação dos algoritmos de localização encontrados no meio científico de maneira a enfocar a mobilidade dos nodos. Em especial, dar-se-á importância ao tipo de mobilidade onde não há atuadores influenciando no movimento.

2.1 Rede de Sensores

Uma rede de sensores como mostrada na Figura 1 consiste de vários nodos, alguns deles especializados ou não, mas todos com severas restrições de potência e processamento de dados (PANTAZIS; VERGADOS, 2007). Os nodos especializados, no entanto, são aqueles que podem possuir capacidades adicionais frente aos demais, 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*¹ é pequeno em relação aos demais nodos. Isto é devido a questões de custo de implementação, manutenção ou fator de forma final do *hardware* (AKYILDIZ et al., 2002).

Basicamente um nó sensor é composto pelas unidades mostradas na Figura 2. Os sinais analógicos produzidos pelos sensores, baseados no fenômeno observado, são convertidos por um conversor analógico-digital e então disponibilizados à unidade de processa-

¹O número de dispositivos em uma rede de sensores pode ser algumas ordens de magnitude maior que em uma rede dita *ad hoc*, onde normalmente cada elemento possui uma identificação única.

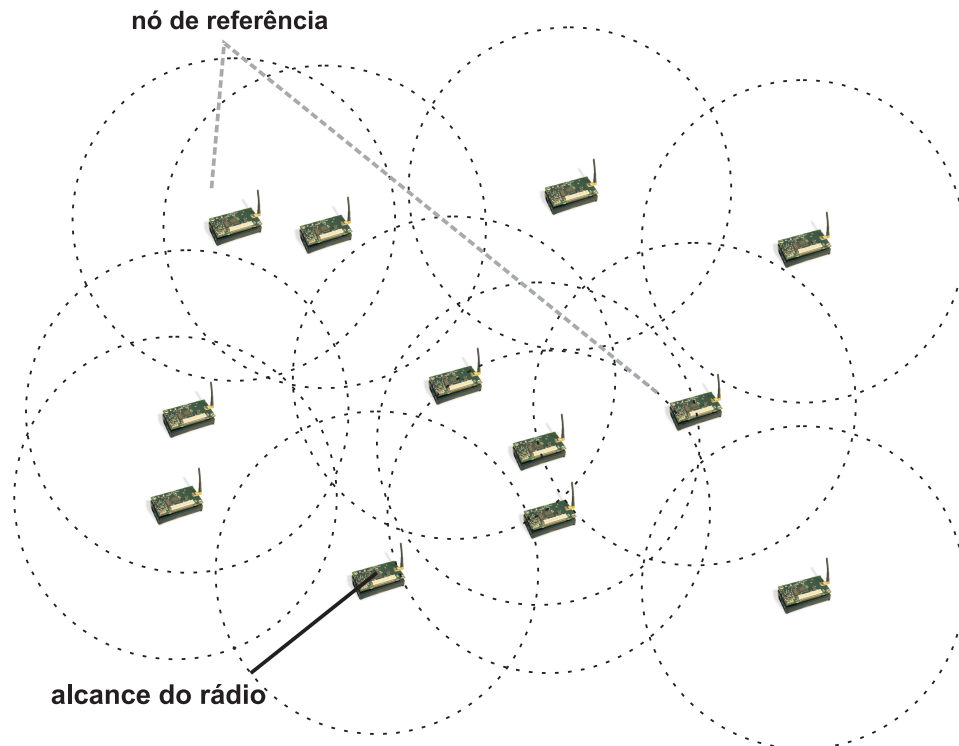


Figura 1: Rede de sensores sem fio evidenciando os nodos de referência.

mento, formada por um componente de armazenamento e por um microprocessador. Esta unidade por sua vez, interpreta o sinal obtido, realiza as tarefas pertinentes e armazena os resultados. O componente de processamento também é responsável por manter o sistema operacional, caso seja necessário, algoritmos e protocolos que sustentam a aplicação. A unidade de comunicação conecta o nó à rede, enviando a leitura dos sensores ou o resultado da manipulação desta informação e recebendo dados de outros nodos para utilizá-los em sua computação local, ou apenas reenviá-los ao seu destinatário alvo.

Um dos componentes mais importantes do nó sensor mostrado na Figura 2 é a unidade de alimentação. A alimentação de um nó sensor pode ser obtida por baterias ou pelo meio ambiente onde o nó se encontra. Particularmente, a energia extraída do meio ambiente pode ter várias origens distintas, das quais pode-se salientar:

- energia solar (BHUVANESWARI et al., 2009);
- indução eletromagnética a partir de um sinal de rádio-frequência (FRIEDMAN; HEINRICH; DUAN, 1997; ABBAK; TEKIN, 2009) (RF) propositalmente emitido próximo ao nó². É preciso portanto que um circuito de recuperação aja sobre o campo RF

²Esta proximidade, para que se entenda, equivale a valores da ordem de centímetros, partindo de menos de um centímetro (SHAMELI et al., 2008), passando por algumas dezenas desta unidade (WANG et al., 2009; HLA; AUNG, 2009), podendo alcançar uma distância de 1,5 metro (BOONTRAI; JINGWANGSA; CHERNTANOMWONG, 2009).

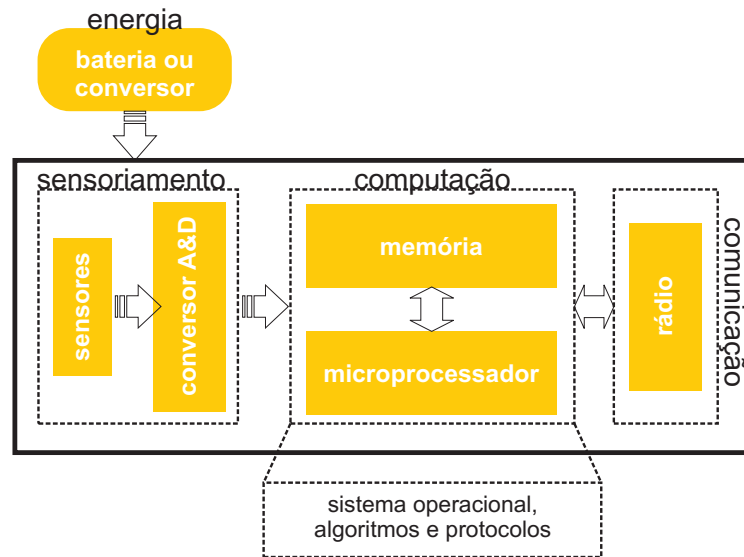


Figura 2: Componentes de um nó sensor.

para gerar e manter uma tensão adequada ao funcionamento do *hardware*. Este processo é comumente utilizando na identificação por rádio-frequência, ou RFID, de objetos. A tecnologia RFID pode ser considerada como uma variação da rede de sensores (PEREIRA et al., 2008), no entanto, estes RFID's passivos possuem uma capacidade extremamente limitada acerca de sua comunicação e seu poder de processamento;

- entidades corpóreas³ (STARNER, 1996), tais como o calor do corpo, respiração, pressão sanguínea e caminhada humana;
- vibrações mecânicas (WILLIAMS; YATES, 1995).

Essencialmente uma rede de sensores é responsável por monitorar fenômenos desenvolvendo três tarefas básicas: detecção, processamento e transmissão. A detecção é realizada dentro do fenômeno ou muito próxima a ele. Esta tarefa é desempenhada pelos sensores que fazem parte do nó: umidade, pressão, temperatura ou luminosidade, por exemplo. A transmissão depende do meio utilizado, cabos físicos ou ar. O desempenho do processo de transmissão sem fio, isto é, a taxa de pacotes recebida⁴ pelo nó alvo, em uma transmissão com endereço especificado, dependerá do meio onde a rede se encontra (LEE; CERPA; LEVIS, 2007; MAHESHWARI; JAIN; DAS, 2008). A detecção do evento sob estudo implica então que existam nodos na região do fenômeno e, no caso de tais nodos serem aleatoriamente dispostos sob uma determinada área, existirá a necessidade da rede ser auto-organizável (MILLS, 2007; GAURA; NEWMAN, 2006; RABAEY et al., 2000) e de possuir

³É possível transpor estas suposições ao organismo de animais também.

⁴PRR - *Packet Reception Rate*.

um sistema de localização. A necessidade de auto-organização da rede pode ser motivada por uma disposição aleatória dos nodos ou também pelo fato de a rede ser móvel, com os seus elementos executando movimentos de maneira controlada ou não.

Uma vez que o campo de estudo de redes de sensores é uma área em contínua expansão e de grande investigação no meio científico, também existe uma diversa gama de aplicações possíveis de serem empregadas neste campo. Algumas destas aplicações incluem, mas não estão limitadas a, monitoramento de rios, rastreamento de animais selvagens ou domésticos, localização de pessoas a serem resgatadas em um desastre, detecção de intrusos em uma residência, monitoramento de campos de batalha, tratamentos médicos, brinquedos interativos, entre outros. Em poucas palavras, existem diversos cenários nos quais sistemas de localização podem ser estudados, ou seja, as necessidades de uma rede de sensores formam um conjunto de fatores muito ligados ao meio onde ela está inserida (NICULESCU, 2005). O produto esperado da pesquisa nesta área é comum, ou seja, descobrir a localização. No entanto, é possível que um determinado caso exija que os nodos da rede possuam atuadores para se movimentarem e, em outro caso, tais atuadores sejam impossíveis de serem empregados.

2.2 Classificação

Existem diferentes métodos utilizados para desempenhar o processo de localizar nodos em redes de sensores. Tais métodos ou algoritmos implicam muitas vezes em uma grande dificuldade ao procurar estabelecer comparações de desempenho justas entre algoritmos de localização. Isto ocorre principalmente porque as descrições dos métodos são realizadas em linguagens de programação distintas e a simulação destas propostas realizadas em programas que muitas vezes não disponibilizam os mesmos parâmetros físicos, por exemplo. Neste trabalho o interesse é ressaltar qualitativamente as características que envolvam o aspecto da mobilidade da rede, enfatizando o processo de localização.

Existem fatores que precisam ser levados em consideração ao analisar algoritmos de localização para redes de sensores, principalmente quando incluem em sua topologia dispositivos móveis. Estes parâmetros englobam, por exemplo:

- como estes dispositivos ou nodos adquirem movimento;
- se algum destes nodos requer equipamento mecânico adicional;
- que tipo de modelo de mobilidade foi utilizado durante a simulação;

- em que plataforma foram validados os resultados.

Todos estes pontos possuem impacto acerca de diversos aspectos relacionados com o desempenho do algoritmo, como o consumo de potência, a escalabilidade da rede de sensores, a medida da convergência dos resultados e o compromisso com a aplicação alvo. Estas características irão trilhar a especificidade e, portanto, o projeto do algoritmo de localização para operar em um determinado cenário.

Devido à quantidade elevada de tópicos relacionados com a análise de métodos do cálculo da posição de nodos, a realização de uma análise justa (LANGENDOEN, 2006) de diferentes algoritmos de localização pode ser comprometida quando é considerada apenas a coleta de informações sobre os mesmos em artigos científicos. Algoritmos de localização podem ser avaliados através de:

- experimentos físicos;
- métodos analíticos;
- simuladores voltados a redes de sensores como o TOSSIM (LEVIS et al., 2003);
- simuladores construídos para um propósito específico como em (HU; EVANS, 2004);
- ou mesmo uma ferramenta amplamente empregada no meio acadêmico e profissional como o MatLab (MOZUMDAR et al., 2008).

Por outro lado, reproduções de resultados sem a disponibilização dos códigos fontes pelos autores podem inviabilizar tal tarefa. Dadas estas disparidades, não se tem por objetivo coletar dados numéricos de artigos mas, em contrapartida, realizar uma revisão qualitativa de importantes métodos de localização enfatizando o aspecto da mobilidade.

Optou-se neste trabalho por classificar os algoritmos de localização pelo tipo de comunicação empregada no método, além de como o aspecto da mobilidade é tratado e se a abordagem prevê ou não a utilização de um sistema de auto-localização. A seguir, portanto, serão descritos todos estes fatores de maneira a embasar a construção de uma tabela que reúna as características de algoritmos presentes na literatura.

2.2.1 Sistema de Posicionamento Global

Uma solução para obter a localização dos nodos em uma rede móvel é através do sistema NAVSTAR. Também referido como Sistema de Posicionamento Global ou

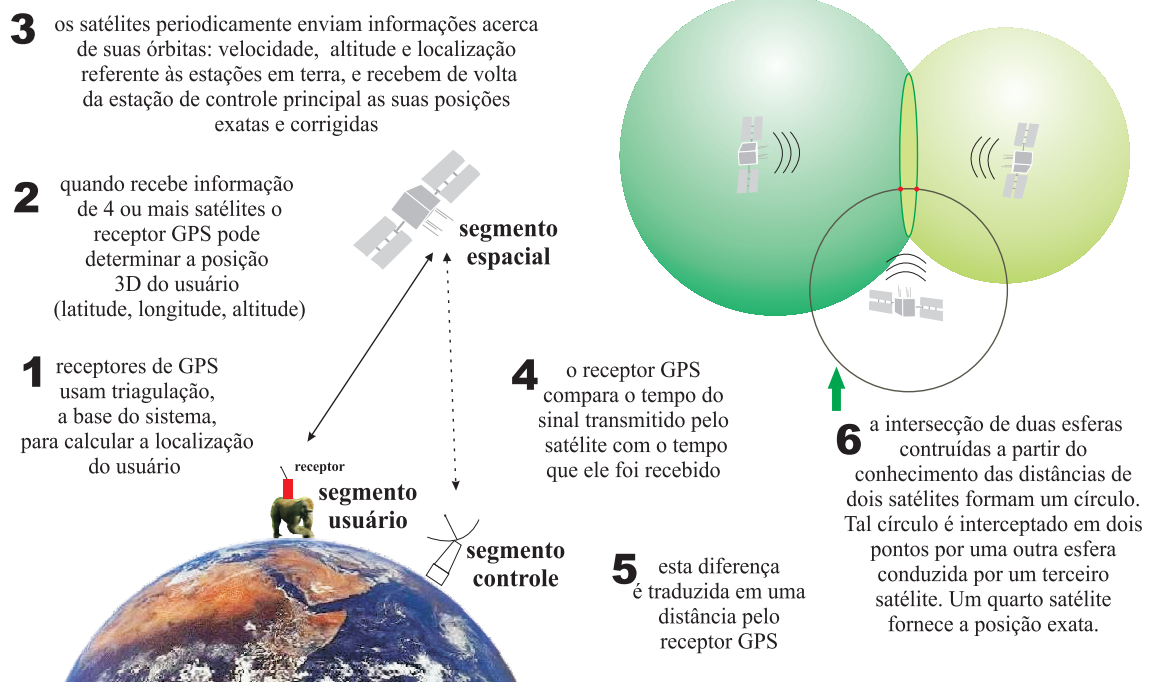


Figura 3: Fluxo simplificado para a localização utilizando o sistema GPS.

GPS (PARKINSON; GILBERT, 1983; GETTING, 1993), este sistema de navegação é baseado em satélites e utiliza a triangulação para estabelecer posições, como mostra o esquema simplificado na Figura 3. Os receptores GPS captam a informação do sinal de pelo menos 4 satélites e utilizam a triangulação para calcular a posição exata do nó em um sistema de coordenadas 3D, comparando o tempo entre a transmissão do sinal pelo satélite com o tempo no qual ele é recebido e, desta forma, calculando a diferença entre estes dois períodos. Os receptores GPS recebem as posições aproximadas e corrigidas (exatas) dos satélites para então calcular o quão longe eles estão de cada satélite, usando o princípio do tempo de chegada: $velocidade \times tempo\ de\ viagem = distância$. Desta maneira é imperativo o uso de relógios de alta exatidão, caso contrário os erros de cálculo podem ser inaceitáveis.

A velocidade do sinal que chega ao dispositivo GPS é a de uma onda de rádio, 300.000 Km/s menos qualquer atraso devido à viagem pela atmosfera terrestre. O tempo é conseguido pela troca de informação entre o receptor e o satélite, onde o receptor procura combinar um código gerado de maneira aleatória com outro previamente gerado pelo satélite, desta forma, determinando o quanto é necessário atrasar (permutar) seu código para combinar com o código do satélite. O sistema GPS foi primeiramente desenvolvido com propósitos militares, mas está disponível para civis desde os anos 80. Atualmente existem 24 satélites⁵ circulando o planeta Terra duas vezes por dia em uma órbita muito

⁵Totalmente operacionais desde 1999.

precisa e transmitem sinais de rádio de potência da ordem de 20W a 50W. A informação contida neste sinal não irá ultrapassar a maioria dos objetos sólidos como prédios ou montanhas, pois é necessária a existência de uma linha de visão direta, ou *line of sight*, com o receptor.

O sistema GPS consiste basicamente em três segmentos: espaço, controle e usuário. O segmento de controle é responsável por monitorar as estações em terra e em especial uma estação mestre de controle em volta do mundo conjuntamente com duas outras antenas, provendo informação corrigida a respeito da órbita e do tempo para os satélites. O receptor GPS e quem o utiliza compõem o segmento usuário, enquanto que o segmento espacial aglomera 21 satélites mais 3 reservas a cerca de 19.000 Km acima da superfície da Terra. Isto permite ao receptor GPS captar sinais de pelo menos 4 satélites a qualquer tempo.

Considerando o acima descrito e o fato de que tal sistema está disponível 24 horas por dia, provendo posições absolutas sem custos financeiros de contratação do serviço, pode-se intuir que o problema da localização está resolvido. As características mostradas fazem do GPS, num primeiro momento, uma abordagem atrativa para descobrir a localização de todos os nodos em uma rede de sensores. Embora seja verdade dizer que a localização possa ser realizada apenas utilizando dispositivos GPS, eles são caros em termos de consumo de potência e custo financeiro do dispositivo (LADD et al., 2005; OH-HEUM; HA-JOO, 2007; SAVVIDES; HAN; STRIVASTAVA, 2001). Desta forma, o seu emprego maciço como solução para a questão da localização em uma rede com dezenas ou centenas de nodos pode tornar-se economicamente inviável. Além disto, a recepção do sinal em interiores não é garantida. De acordo com (OH-HEUM; HA-JOO, 2007), pelo fato do dispositivo GPS requerer uma linha de visão direta com o satélite, este método não é indicado para locais subterrâneos ou com obstruções.

As limitações supramencionadas motivaram a pesquisa por métodos alternativos baseados na medição da distância entre dois nodos. Algumas aplicações, no entanto, realmente precisam saber informação de posicionamento acerca da posição global e, para limitar a quantidade de dispositivos GPS utilizados, os algoritmos, sejam eles para redes estáticas ou móveis, frequentemente lançam mão do emprego de alguns nodos chamados de âncoras (*anchors*), também referenciados por *seeds*, *beacons* e *landmarks*. Tais nodos desempenham um papel diferenciado na rede e possuem um prévio conhecimento de suas localizações considerando um sistema de coordenadas global ou relativo (HU; EVANS, 2004; BAGGIO; LANGENDOEN, 2006; DIL; DULMAN; HAVINGA, 2006; WU; SHENG; SONG, 2006; CARUSO et al., 2005; OH-HEUM; HA-JOO, 2007).

Os nodos âncora enviam suas posições para todos os nodos em seu raio de alcance para que estes possam localmente calcular as suas próprias posições de forma distribuída. Um método de cálculo das coordenadas que considera o aspecto da mobilidade da rede é o método adaptado Monte Carlo⁶ (DELLAERT et al., 1999) para rede de sensores móveis. Dispositivos GPS foram utilizados pelo algoritmo de Localização Monte Carlo para lidar com o cenário onde o movimento aleatório, sem a presença de atuadores, de nodos comuns e âncoras era considerado (HU; EVANS, 2004).

Como mencionado anteriormente, a abordagem de localização por GPS utiliza essencialmente a triangulação do sinal e a diferença entre tempos de viagem para calcular a posição. Considerando uma localização em um sistema de coordenadas que opere em duas dimensões (2D), pelo menos três pontos de referência, cada qual com um conhecimento prévio de sua própria posição, são necessários para localizar completamente um nó usando a triangulação, como pode ser observado na Figura 4. Uma vez que curvas quadráticas podem ter múltiplas intersecções no plano, um terceiro elemento de referência é necessário. As distâncias entre estes nodos âncora e o ponto desconhecido são também necessárias para realizar a triangulação e, portanto, os nodos do tipo âncora precisam também estar no raio de alcance do nó desconhecido, como representado pela circunferência pontilhada da Figura 4. O sistema de equações descrito pela Equação 2.1 pode ser resolvido, após linearização, pelo emprego do algoritmo dos mínimos quadrados, por exemplo.

$$\begin{aligned}
 (x_1 - u_x)^2 + (y_1 - u_y)^2 &= R_1^2 \\
 (x_2 - u_x)^2 + (y_2 - u_y)^2 &= R_2^2 \\
 (x_3 - u_x)^2 + (y_3 - u_y)^2 &= R_3^2
 \end{aligned}
 \tag{2.1}$$

onde R_i é a distância conhecida entre o ponto desconhecido e cada nó âncora i , (x_i, y_i) são as coordenadas bi-dimensionais do i^{th} ponto de referência e (u_x, u_y) são as coordenadas do nó cuja posição se deseja obter.

2.2.2 Métodos Baseados na Medição da Distância - *Range-based*

As distâncias entre o nó desconhecido e os nodos âncora são obtidas por medidas de alcance, ou *range*, do sinal de rádio. Abordagens *range-based* requerem um *hardware* especial, como no caso da medição da distância pelo ângulo de chegada e são classificadas em:

⁶Este método será explicado na Seção 3.3.

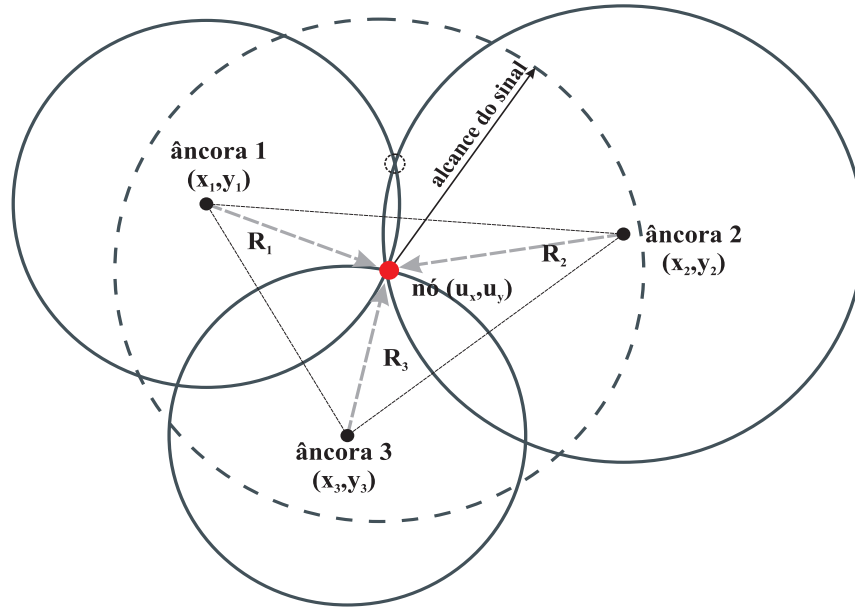


Figura 4: Interpretação geométrica da trilateração. Dois nodos âncora indicarão a posição para dois pontos. Um terceiro nó âncora é necessário para resolver a posição desconhecida de forma precisa. A localização está na intersecção de no mínimo três circunferências, compostas pelos três nodos âncora. Esta figura apresenta o caso ideal onde não existem erros de medida. Na prática, as três circunferências não irão interceptar em um ponto.

- tempo de chegada (TOA - *Time of Arrival*);
- diferença no tempo de chegada (TDOA - *Time Difference of Arrival*);
- ângulo de chegada (AOA - *Angle of Arrival*);
- intensidade do sinal recebido (RSSI - *Received Signal Strength Indicator*).

2.2.2.1 Tempo de Chegada - *Time of Arrival*

Time of Arrival ou diferença no tempo de chegada (TOA) (KAPLAN, 1996; CAFFERY; STUBER, 1998) utiliza o tempo de chegada absoluto, *time of flight*, do sinal viajando de um nó qualquer, a ser descoberta a localização, até um dado nó âncora, ou vice-versa. Esta técnica trata da medição do tempo que o sinal leva para ser transmitido, levando consigo informações identificadas do nó emissor a uma localização conhecida, até alcançar um nó receptor. Geometricamente como mostrado na Figura 4, tal distância representa os raios R_1 , R_2 e R_3 das circunferências, centradas nos nodos âncora, dentro dos quais o nó desconhecido deve estar contido. Novamente, como este método é baseado em informações de tempo, a sincronização entre os nodos âncora e o nó desconhecido deve ser muito precisa, de outra forma os cálculos podem retornar localizações com grandes erros.

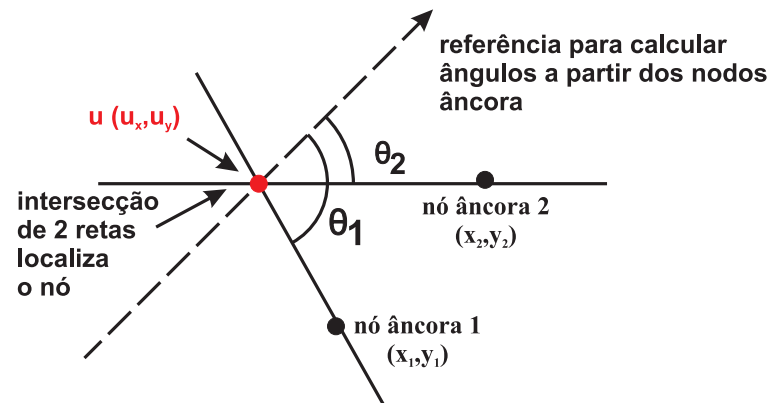


Figura 5: A posição do nó desconhecido é estimada a partir dos ângulos medidos para os sinais que estão chegando, detectados nos nodos âncora fixos (triangulação). Quaisquer dos dois ângulos podem determinar a localização a partir do triângulo formado. Como pode ser visto, a intersecção de duas linhas originadas pela medida do ângulo a partir do eixo do nó desconhecido com respeito a outros nodos vizinhos, cada qual formado por uma radial (medição reversa do ângulo) a partir do âncora até o nó desconhecido, é utilizada para a localização em duas dimensões.

2.2.2.2 Ângulo de Chegada - *Angle of Arrival*

Angle of Arrival ou ângulo de chegada (AOA) (RONG; SICHITIU, 2006; NICULESCU; NATH, 2003), por outro lado, implica na utilização de um equipamento especial: um conjunto de antenas direcionais em cada nó sensor, para habilitar a medição do ângulo de incidência dos sinais recebidos relativo ao seu próprio eixo, uma direção fixa chamada orientação. A Figura 5 apresenta uma ilustração gráfica deste método. A seta pontilhada representa a orientação do nó, a partir da qual todos os outros ângulos são reportados. A orientação, medida no sentido horário ou trigonométrico, pode ser absoluta, se é fixada para o Norte (ângulo igual a zero), ou relativa em caso contrário. A localização do nó a ser determinado é restrita ao longo de uma linha iniciando a partir do nó âncora. Desta forma, com um mínimo de dois nodos âncora não colineares, a localização do nó pode ser calculada como a intersecção de duas linhas, por trigonometria simples ou triangulação. Problemas como reflexão, difração e dispersão dificultam a precisão da medida dos ângulos.

2.2.2.3 Tempo de Diferença de Chegada - *Time Difference of Arrival*

Time Difference of Arrival (GUSTAFSSON; GUNNARSSON, 2003; BUCHER; MISRA, 2002; SPENCER, 2007; KRIZMAN; BIEDKA; RAPPAPORT, 1997), também conhecida como técnica hiperbólica de localização da posição, utiliza a co-relação cruzada de um sinal chegando a dois nodos âncora, precisamente sincronizados no tempo, como um método indireto para calcular a distância e triangular a posição de um nó desconhecido. Uma hipérbole é uma curva a partir da qual a diferença nas distâncias a dois pontos fixos (os focos) é constante. Desta forma, esta correlação cruzada, ou atraso, define uma hipérbole

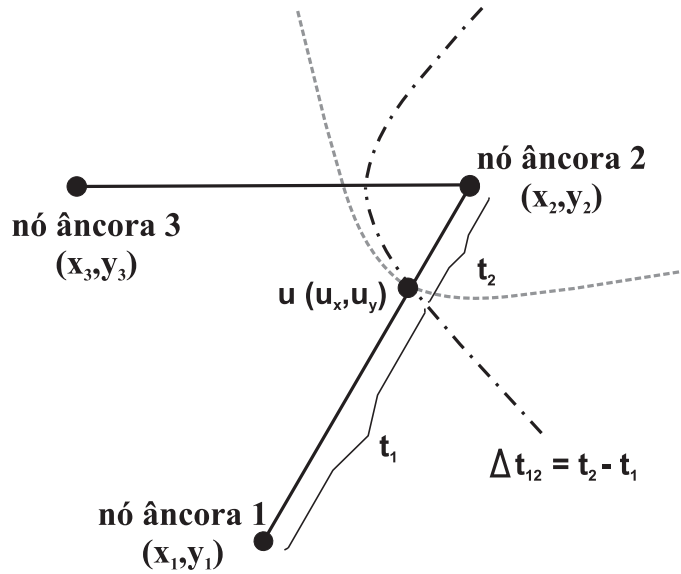


Figura 6: Neste exemplo de multilateração, dois pares de nodos âncora (a_1, a_2) e (a_2, a_3) são os focos das hipérbolas. O nó u está localizado na intersecção de tais hipérbolas e t_1, t_2 são os tempos medidos nos quais os sinais levam para alcançar os nodos âncora a_1 e a_2 , respectivamente, a partir de u .

com uma diferença constante dos nodos âncora, que estão localizados nos focos (CAFERY; STUBER, 1998). Com isto, o algoritmo converte uma diferença de tempo em uma distância constante a partir do par correspondente de nodos âncora, os quais descrevem uma curva hiperbólica única. A diferença em distâncias dos dois nodos âncora a partir do nó desconhecido é estimada baseada na diferença no tempo de chegada do sinal que parte do nó desconhecido e chega aos dois nodos âncora. A Equação 2.2 resume tais relações e pode ser manipulada até atingir a forma padrão da equação de uma hipérbole (mais detalhes sobre esta formulação pode ser encontrado em (SPENCER, 2007; SCHAU; ROBINSON, 1987)).

$$c\Delta t_{12} = \sqrt{(x_1 - u_x)^2 + (y_1 - u_y)^2} - \sqrt{(x_2 - u_x)^2 + (y_2 - u_y)^2} \quad (2.2)$$

onde c é a velocidade do sinal (velocidade de uma onda eletromagnética, por exemplo), Δt_{12} é a diferença no tempo de chegada ao nodo âncora em relação ao nó desconhecido, as coordenadas $(x_1, y_1), (x_2, y_2)$ representam as posições dos âncoras e (u_x, u_y) são as coordenadas do nó cuja posição deve ser determinada.

A Figura 6 apresenta uma ilustração geométrica da Equação 2.2. Cada medição TDOA remonta a uma curva hiperbólica ao longo da qual o nó desconhecido pode estar posicionado e, através da intersecção de duas hipérbolas 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 (SPENCER, 2007)

com apenas duas hipérbolas, 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.

Quando múltiplos nodos âncora estão disponíveis, múltiplas hipérbolas são formadas, e a interseção de um grupo de hipérbolas geometricamente provê a estimativa da localização do nodo desconhecido. Crítico para uma operação apropriada, os âncoras 1 e 2 precisam estar sincronizados e espalhar seus sinais ao mesmo tempo para que qualquer nó que esteja ao alcance do rádio possa receber informação, caso contrário, os erros podem se tornar inaceitáveis. Assumindo uma onda de rádio, com a velocidade próxima aos $c = 300.000$ Km/s, ela leva aproximadamente 3,3ns para viajar 1m. Para uma boa estimativa na localização pela técnica de TDOA, Savarese (SAVARESE; RABAEY; BEUTEL, 2001) reportou que é necessária uma precisão mínima de 0,3ns por metro entre dois nodos âncora, o que corresponderia a um erro de 10%.

2.2.2.4 Indicação de Força do Sinal Recebido - *Received Signal Strength Indicator*

O método intitulado Indicação de Força do Sinal Recebido ou em inglês *Received Signal Strength Indicator* RSSI (MONDINELLI; KOVACS-VAJNA, 2004; PATWARI; III, 2003) é baseado no nível de potência do sinal na recepção. Tal sinal é enviado pelo nó desconhecido e, no receptor, sua força ou intensidade pode ser calculada utilizando a Equação 2.3:

$$P_t = P_r \cdot K \cdot r^{-2} \quad (2.3)$$

onde P_r é a potência recebida [W], P_t é a potência transmitida [W], r é a distância entre as antenas de transmissão e recepção [m] e K é um fator relacionado com características das antenas, comprimento de onda e meio de transmissão.

O método RSSI é baseado na precisão da informação de potência que é transmitida, P_t , a qual pode ser efetivamente controlada, provendo boa exatidão para distâncias de cerca de 3 metros, em boas condições de transmissão. Entretanto sua exatidão pode sofrer com cenários onde fontes de interferência são relevantes, resultando em um caminho onde não há linha de visada ou *Non-Line of Sight* (NLOS) (CAPKUN; HAMDI; HUBAUX, 2001). Por causa disto, não é possível unir transmissor e receptor por uma linha imaginária reta. Apesar desta hipótese, uma grande vantagem do método RSSI é que ele não precisa sincronização de tempo, tornando-o preferível a TOA ou TDOA neste aspecto. Adicionalmente, nenhum *hardware* especial é requerido, o que reduz o custo de

implementação relativo ao método AOA, por exemplo.

2.2.2.5 Discussão Acerca dos Métodos

Como discutido anteriormente, a técnica de TOA requer boa sincronização entre os nodos, emissor e receptor, e três nodos âncora para calcular a trilateração. Uma maneira para reduzir a dependência da sincronização é utilizar a *Round Trip Transmission* (RTT). Em (KANG et al., 2006) foi apresentado um método baseado neste conceito como um melhoramento sobre TOA. Ele consiste nos nodos âncora transmitirem um sinal ao nó desconhecido e recebê-lo de volta. Conhecendo-se o tempo de atraso de retorno e a velocidade de propagação do sinal é possível estimar a distância entre o nó desconhecido e o âncora sem sincronização de tempo utilizando Equação 2.4:

$$d = \frac{c \cdot (t_2 - t_1)}{2} - c \cdot t_p \quad (2.4)$$

onde c é a velocidade do sinal, d é a distância entre o nó âncora e o nó desconhecido, t_1 e t_2 são os tempos nos quais o sinal é transmitido e recebido, respectivamente no nó âncora e t_p é o tempo de processamento para nó desconhecido.

Em (XIAO; REN; TAN, 2006) os nodos âncora também não precisam estar sincronizados com os nodos desconhecidos. Em contrapartida, os nodos âncora precisam estabelecer relações precisas de tempo com eles mesmos para cancelar o tempo inicial desconhecido das transmissões e os nodos desconhecidos também precisam sincronização de tempo entre si, possivelmente para trocar dados com outros nodos desconhecidos. Em (PARK; AHN; YU, 2007) os autores apresentaram um método alternativo chamado *inter-node time difference of arrival* (ITDOA) que não requer nenhum tipo de sincronização. Todos os nodos de referência são assumidos serem assíncronos com respeito a seus próprios relógios locais. No entanto, estes mesmos nodos de referência e os nodos desconhecidos são supostos possuírem seus sistemas sendo operados a mesmas taxas de frequência. A principal idéia é estabelecer um número de pulsos de *clock* entre o nó desconhecido e os nodos de referência para extrair informação de tempos de chegada através de formulação matemática e então computar tais distâncias.

Como descrito em TOA, TDOA também necessita de pelo menos três âncoras para localizar o nó desconhecido utilizando um método geométrico chamado de multilateração: uma maneira de calcular a posição que usa TDOA de sinais emitidos por pelo menos três ou mais transmissores sincronizados com informação de localização. Os dispositivos que

operam sob AOA para determinar a posição são os mais caros⁷ dentre os apresentados até agora, porém utilizam o processo de triangulação e apenas dois nodos âncora são necessários para se computar a posição do nó desconhecido, pela intersecção de duas linhas partindo dos âncoras até o nó. RSSI é o método mais barato⁸ entre TOA, TDOA e AOA, mas possui menor exatidão também. Ele é mais dependente da linha de visão direta e mais susceptível a variações de amplitude do sinal recebido, sendo então indicado para localidades livres de obstruções.

2.2.3 Métodos Baseados na Conectividade - *Range-free*

Enquanto métodos *range-based* utilizam medidas da distância, métodos *range-free* apenas baseiam-se no conteúdo das mensagens, isto é, a informação de conectividade para derivar a localização dos nodos na rede. Os algoritmos de localização *range-free* podem ser divididos em:

- técnicas locais, que recaem em uma alta densidade de nodos âncora na rede, de tal maneira que cada nó da rede pode ouvir diversos nodos de referência como os algoritmos CAB (VIVEKANANDAN; WONG, 2006) e Centroid (BULUSU; HEIDEMANN; ESTRIN, 2000; DENG et al., 2008; CHEN et al., 2008a; FORGHANI, 2008; LIU; ZHANG; HU, 2008);
- técnicas de contagem de saltos (NICULESCU; NATH, 2001; JI; LIU, 2006; CHUAN, 2008; CHEN et al., 2008b; XU et al., 2009; ZHAO; SUN; ZHANG, 2009) ou *hop counting*, onde as mensagens iniciadas pelos nodos de referência são propagadas através da rede com acréscimo na contagem de saltos ou *hops*. Este processo é denominado por *flooding of the network*).

O algoritmo de localização CAB (*Concentric Anchors-Beacons*), proposto em (VIVEKANANDAN; WONG, 2006), é direcionado a redes de sensores estacionárias e apresenta uma técnica com computação local que utiliza o método *range-free*. Como mostrado na Figura 7, três nodos de referência, que são equipados cada um com um receptor GPS, transmitem sinais especiais chamados de *beacons* em dois diferentes níveis de potência, representados pelas circunferências concêntricas ao redor dos nodos âncora.

Estes dados são coletados pelo nó sensor cercado por três nodos âncora, mas somente dois deles são selecionados. Eles são usados para calcular os pontos de intersecção válidos

⁷É necessário um conjunto de antenas externas acopladas ao nó sensor.

⁸Detectores de potência (LEE; SONG; NAM, 2008) são circuitos integrados com os demais blocos de um nó. Além disto, a medição da distância não exige sincronização entre nodos.

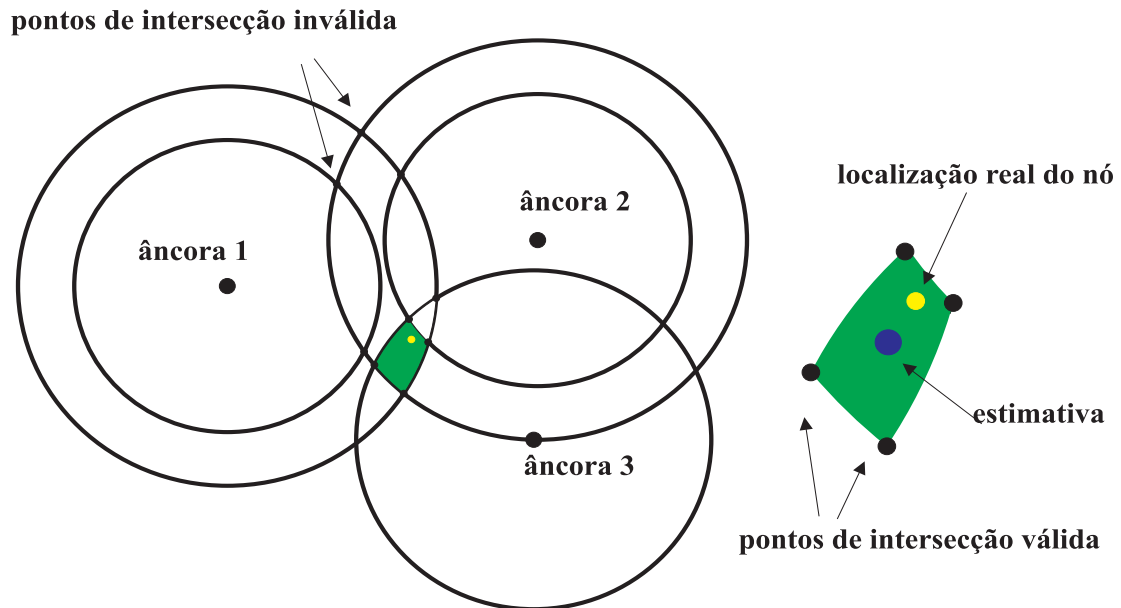


Figura 7: Algoritmo CAB: dois diferentes níveis de potência são usados para determinar os pontos de intersecção e estimar a posição em uma abordagem *range-free* (adaptado de (VIVEKANANDAN; WONG, 2006)).

que satisfaçam todas as regiões delineadas pelos três nodos de referência. A exatidão deste método é altamente dependente da percentagem de nodos depositados que possuem a característica de serem âncoras. Apesar de métodos *range-free* serem menos exatos que os métodos *range-based*, devido a limitações de *hardware* dos dispositivos sensores, algoritmos de localização *range-free* são uma alternativa viável em termos de custos financeiros (HE et al., 2003). As abordagens puramente *range-free* utilizam apenas o circuito de comunicação, presente em qualquer nó de uma rede sem fio. Desta forma, estes métodos dispensam: a utilização de um circuito adicional de medição da intensidade do sinal⁹; a sincronização entre nodos¹⁰; a utilização de um conjunto de antenas¹¹.

Uma técnica de contagem dos saltos de comunicação bem conhecida é o método DV-Hop (NICULESCU; NATH, 2001). Neste método as distâncias entre vizinhos são medidas utilizando a contagem de saltos na transmissão da informação necessária para atingir o nó de destino. Cada nó âncora sabe sua própria posição e obtém as distâncias para outros âncoras em termos de saltos ou *hops*, como mostrado na Figura 8. O número de saltos é traduzido para uma medida em metros, por exemplo. Esta informação é utilizada para estimar a distância média de um único *hop* pela Equação 2.5. Tian et.al (TIAN et al., 2007) propuseram um melhoramento sobre (NICULESCU; NATH, 2001), dando o nome à sua proposta de SANLA ou *Selective Anchor Localization Algorithm*. Foi considerada

⁹Utilizados em métodos RSSI.

¹⁰Empregada nos métodos TDOA e TOA.

¹¹AOA.

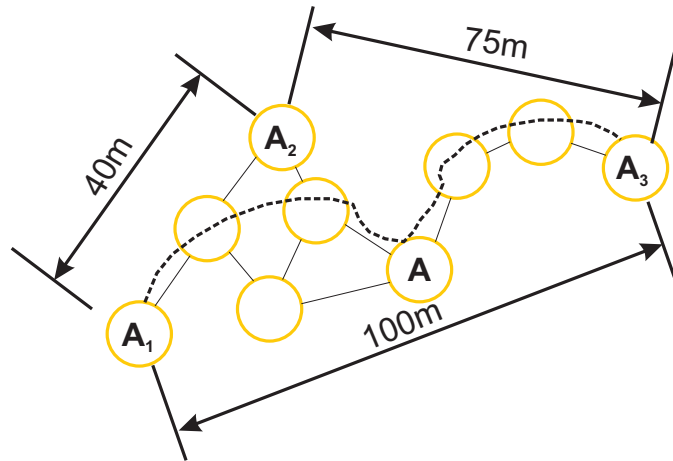


Figura 8: Composição das distâncias euclidianas no método DV-HOP (NICULESCU; NATH, 2001). Figura adaptada de (NICULESCU; NATH, 2001).

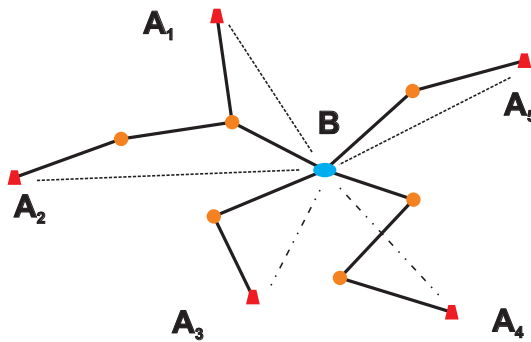


Figura 9: Fonte de erro do algoritmo DV-Hop (adaptado de (TIAN et al., 2007)).

para o aumento da precisão dos resultados uma etapa de seleção dos nodos âncoras. Os autores baseiam-se no fato de que há casos para os quais as estimativas de distância dos nodos desconhecidos aos nodos âncoras são muito precárias, influenciando no resultado final. Pela análise da Figura 9 pode-se verificar que os caminhos entre o nó B e os âncoras A_1 , A_2 e A_5 são próximos a uma linha reta, ou pelo menos não se distanciam muito. Por outro lado, os caminhos entre B e A_3 e A_4 são muito mais tortuosos que os anteriores, no entanto, o nó B , no momento de computar a sua posição levará em conta todos os nodos próximos à sua posição.

$$HopEstimation_i = \frac{\sum \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}}{\sum h_i}, i \neq j \quad (2.5)$$

Uma vez que todos os nodos na rede conseguem adquirir o valor mínimo de número de hops para todos os nodos âncora, tais valores podem então ser convertidos em distâncias físicas. Se o nó desconhecido possui estimativas de distância a pelo menos três nodos âncora, as localizações podem ser estimadas pelo método da multilateração.

Bulusu et.al (BULUSU; HEIDEMANN; ESTRIN, 2000) escolheram o modelo de rádio baseado na conectividade para demonstrar o algoritmo de localização. Os nodos de referência são supostos serem estáticos e o alcance de seus sinais de rádio cobrem toda a região sob estudo, transmitindo periodicamente sinais de referência com suas respectivas posições. Enquanto os nodos movem-se através da região, a idéia principal do método é inferir uma relação de proximidade a um conjunto de pontos de referência e depois calcular a localização com a ajuda da intersecção de regiões de conectividade cobertas por tais pontos ou nodos de referência. Isto é definido como o Centro ou **Centroid** de tais âncoras. Estes nodos de referência são escolhidos quando o resultado de uma métrica de conectividade¹² (*Connectivity Metric*) definida pelos autores em (BULUSU; HEIDEMANN; ESTRIN, 2000) na Equação 2.6 excede, por exemplo, 90%:

$$CM_i = \frac{Nrecv(i, t)}{Nsent(i, t)} \times 100 \quad (2.6)$$

onde $Nrecv(i, t)$ é o número de informações de referência que foram enviadas pelo nó âncora i no período de tempo t e $Nsent(i, t)$ é o número de informações de referência enviadas pelo nó âncora i e recebidas no período de tempo t .

2.2.4 Mobilidade

Os cenários que podem ser imaginados para redes de sensores sem fio incluem aplicações com nodos âncora e/ou nodos comuns que estão em um estado de não-movimento ou que são dispositivos móveis. Considerando redes de sensores, é possível prever situações onde os nodos âncora possuem controle sobre seus movimentos ou não. No primeiro caso, os algoritmos de localização procuram determinar a posição de um robô em uma região com um mapa predefinido. Nesta abordagem, os robôs são frequentemente assistidos por muitos sensores com informação precisa da distância disponibilizada por pontos de referência ou marcos.

Por outro lado, a localização em redes de sensores sem atuadores para o movimento opera em um terreno não mapeado sem controle algum sobre sua mobilidade e sem conhecimento sobre velocidade e direção do movimento. Este cenário relatado é na verdade o caso mais crítico considerando a característica da mobilidade. A maioria dos artigos que enfocam algoritmos de localização para redes de sensores sem fio consideram que o nó possui controle sobre o seu movimento em uma região com um mapa predefinido (REN; MENG;

¹²A métrica de conectividade é localmente calculada pelo nó desconhecido referente a cada nó de âncora.

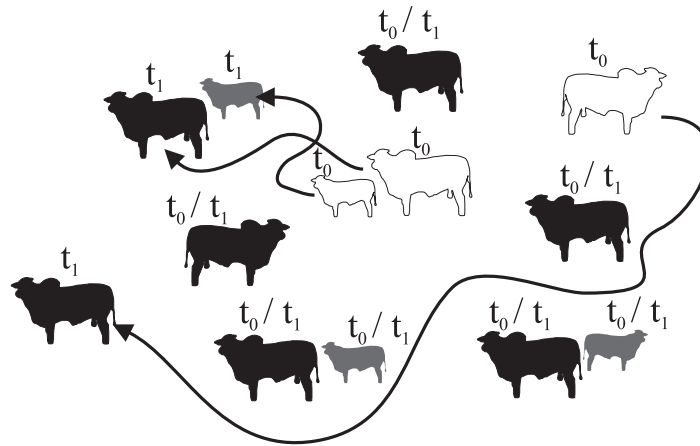


Figura 10: Exemplo abstrato de um movimento aleatório de nodos relativo a animais em uma fazenda. Nesta Figura, t_0 e t_1 representam dois tempos distintos que indicam a primeira e a última posição assumida por alguns animais, respectivamente. Eles podem estar parados ou em movimento.

CHEN, 2006; YUAN; CHEN; XI, 2006; THRUN et al., 2001; LUO; HUBAUX, 2005; SHENOY; TAN, 2005; BERGAMO; MAZZINI, 2002). A Tabela 1 apresenta na linha 1 o cenário mais simples para um algoritmo de localização, isto é, quando todos os elementos da rede são estáticos. Em contraste, a linha número 5 da Tabela 1 mostra a situação mais crítica¹³, onde não há controle sobre o movimento de quaisquer dos elementos da rede.

Tabela 1: Cenários de mobilidade de acordo com o tipo de movimento de nodos âncora e nodos comuns.

cenários de movimento	<i>Nodos de referência</i>		<i>Nodos sensores</i>	
	1	estático		estático
2	controlado/estático			
3	controlado		controlado	
4	aleatório/estático			
5	aleatório		aleatório	

A Figura 10 apresenta um cenário onde nodos comuns e nodos âncora são representados por animais em uma fazenda, um rebanho bovino por exemplo. Este exemplo ilustrativo enfrenta um grande desafio em redes de sensores sem fio, uma vez que os nodos estão em uma situação de movimento não controlado, desta forma, sem possuir conhecimento sobre direções tomadas.

Em contrapartida, a Figura 11 mostra um exemplo de localização estudado em (SHENOY; TAN, 2005) onde os nodos comuns, representados por pontos pretos, são estáticos e os nodos âncora são robôs com atuadores para o movimento. O movimento do veículo móvel mostrado na Figura 11 objetiva reduzir o retângulo tracejado e conseqüentemente a estimativa de área onde o nó poderá estar localizado. Desta forma, Shenoy et.al

¹³O algoritmo desenvolvido nesta Tese abrange o cenário número 4, com os nodos desconhecidos movimentando-se aleatoriamente e os nodos de referência são estáticos.

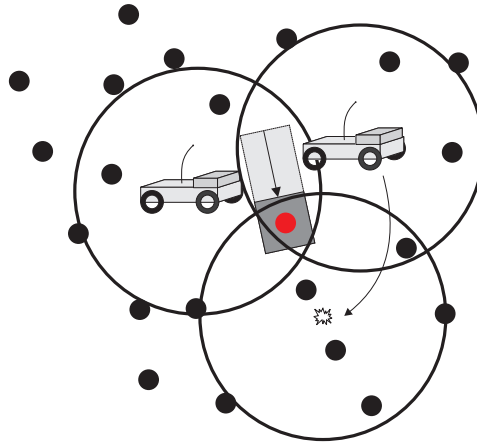


Figura 11: Um cenário onde os nodos âncora são robôs móveis e controlam seus movimentos para percorrer um determinado caminho e localizar nodos estáticos (adaptado de (SHENOY; TAN, 2005)).

lidam neste trabalho com um **algoritmo para localização** de nodos estáticos assistido por um dispositivo móvel e um **algoritmo de navegação** para conduzir a mobilidade deste último.

Os métodos de localização assistidos pelo movimento controlado de seus elementos não podem ser empregados no cenário mostrado na Figura 10, isto é, os nodos não trocam informações uns com os outros para auxiliar na decisão sobre a próxima posição do nó âncora. Estes dois métodos envolvem a independência ou dependência do nó acerca de sua mobilidade e isto irá afetar como o algoritmo de localização irá operar (HU; EVANS, 2004), ou mesmo em que situações ele é suposto de ser aplicado.

Calcular posições para nodos com controle acerca de seus movimentos possui diversas vantagens. Nodos com este tipo de controle podem aumentar a cobertura do fenômeno sob estudo, movendo-se diretamente para uma região específica ou mesmo decidir por ficar imóveis. Entretanto, como exemplificado pela aplicação ilustrada na Figura 10, isto não é sempre possível. Mesmo quando a aplicação alvo requer isto, uma rede construída sob robôs pode não ser a primeira escolha quando se leva em consideração os custos¹⁴ e o consumo de potência. Para redes de sensores densas, empregar um grande número de unidades móveis com movimento controlado pode ser extremamente caro. Uma vez que na maioria das vezes os nodos são autônomos, as restrições de potência podem impedir a utilização de atuadores nos nodos. Diferentes algoritmos são projetados para cobrir distintas situações (LANGENDOEN; REIJERS, 2003) de forma que não existe um método ou algoritmo apropriado para todas as aplicações de redes. Um cenário onde um nó sensor móvel é depositado pode ser único e requerer equipamento especial, demandar restrições

¹⁴Já no início do século estimava-se que uma rede de sensores não poderia ter dispositivos que custassem mais de um dólar (RABAEY et al., 2000).

de potência, tempo aceitável para calcular valores, precisão ou pode ter que lidar com um fenômeno físico específico.

Uma tomada a decisão de projetar um método de posicionamento face a um cenário que considere a mobilidade, cabe a preocupação com o padrão de movimento dos nodos. Trabalhar com algoritmos para redes móveis significa executar simulações ou conduzir experimentos onde os nodos envolvidos devem executar um movimento com certa direção, velocidade e seguindo determinado padrão ao longo do tempo ou mesmo sendo aleatórios. Dependendo da abordagem escolhida os nodos comportar-se-ão de tal maneira que recursivamente no tempo os mesmos tomarão decisões a respeito de que direção tomar, velocidade do movimento ou tempo de parada. Para reproduzir tal comportamento precisa-se optar entre diferentes tipos de modelos de mobilidade. A maneira como os dados envolvidos na análise de redes de sensores sem fio móveis são manipulados para se obter os resultados de desempenho possui uma grande dependência sobre o uso de diferentes modelos de mobilidade como parâmetros de entrada (CAMP; BOLENG; DAVIES, 2002).

2.3 Resumo

Este Capítulo apresentou conceitos acerca de redes de sensores, bem como abordou tipos distintos de algoritmos de localização de nodos presentes na literatura para este tipo de rede. Também foram diferenciados tais algoritmos de acordo com a maneira pela qual os dados para a estimativa da localização eram obtidos: *range-free* ou *range-based*, além de evidenciar se o cenário alvo do algoritmo em questão considerava a mobilidade dos nodos ou era constituído por uma rede estática.

3 REVISÃO DE ALGORITMOS DE LOCALIZAÇÃO

Como visto na Seção 2.2, a caracterização de um sistema de localização é influenciada por vários fatores, incluindo o tipo de comunicação, se os nodos âncora precisam da informação de dispositivos GPS, se os nodos são estacionários ou se encontram em movimento. A Tabela 2 reúne algoritmos de localização relevantes presentes na literatura, classificados de acordo com as características apresentadas na Seção 2.2. Como pode ser observado na Tabela 2, diversos métodos possuem propriedades semelhantes e apenas diferem em um ou dois aspectos. Muitos destes algoritmos são baseados em suas versões anteriores com melhoramentos em propriedades como cobertura do fenômeno sob estudo, exatidão das medidas de distância, cálculo da posição estimada ou tempo de computação. Zhang et.al (ZHANG et al., 2006) por exemplo, focou o desenvolvimento no aumento da segurança de métodos *range-based* já existentes, sem prover um novo algoritmo de localização, focando em um cenário onde tais técnicas de localização estão susceptíveis a sofrer ataques e ter a sua informação degradada.

Este Capítulo apresenta uma revisão de diversos algoritmos de localização de nodos presentes na literatura. É dado especial enfoque aos algoritmos de localização que exploram o aspecto da mobilidade de maneira controlada e não controlada, bem como a maneira pela qual as distâncias são mensuradas: diretamente através de variáveis físicas ou considerando a conectividade da rede.

3.1 Coordenadas Absolutas e Relativas

Diversos métodos apresentados na Tabela 2 não são baseados sobre receptores GPS e empregam uma abordagem *range-based* de medição da distância. As posições calculadas em uma rede de sensores podem assumir dois estados distintos: absolutas ou relativas

¹Informação não disponível.

²Mobilidade encarada como um evento ocasional.

Tabela 2: Algoritmos de localização de nodos em redes de sensores.

	técnica	tipo de comunicação	baseado em GPS	mobilidade
1	(CAPKUN; HAMDI; HUBAUX, 2001)	<i>range-based</i> (TOA)	não	nodos imprevisíveis
2	(CHENG et al., 2004)	<i>range-based</i> (TDOA)	não	âncoras estáticos/nodos móveis
3	(LI; EKPENYONG; HUANG, 2004)	<i>range-based</i> (TDOA)	não	nodos estáticos/âncoras móveis
4	(ZHANG; YU, 2008)	<i>range-based</i> (TDOA)	sim	nodos estáticos/âncoras móveis
5	(LADD et al., 2005)	<i>range-based</i> (RSSI)	não	âncoras estáticos/robôs móveis
6	(DELLAERT et al., 1999)	<i>range-based</i> (n.d. ¹)	não	robôs
7	(SICHITIU; RAMADURAI, 2004)	<i>range-based</i> (RSSI)	sim	nodos estáticos/âncoras ou robôs móveis
8	(DIL; DULMAN; HAVINGA, 2006)	<i>range-based</i> (RSSI)	sim	nodos/âncoras imprevisíveis
9	(WU; SHENG; SONG, 2006)	<i>range-based</i> (n.d. ¹)	sim	nodos/âncoras aleatórios
10	(SHENOY; TAN, 2005; SREENATH; LEWIS; POPA, 2006; SSU; OU; JIAU, 2005; YU; YU; FENG, 2008; XIAO; JIN-KUAN; YUN, 2008)	<i>range-free</i>	sim	nodos estáticos/âncoras ou robôs móveis
11	(NICULESCU; NATH, 2001)	<i>range-free</i>	sim	nodos/âncoras móveis
12	(BAGGIO; LANGENDOEN, 2006; HU; EVANS, 2004; STEVENS-NAVARRO; VIVEKANANDAN; WONG, 2007; HSIEH; WANG, 2006)	<i>range-free</i>	sim	nodos/âncoras imprevisíveis
13	(HE et al., 2003)	<i>range-free</i>	sim	nodos/âncoras móveis
14	(BULUSU; HEIDEMANN; ESTRIN, 2000)	<i>range-free</i>	não	âncoras estáticos/ nodos móveis ²

(HIGHTOWER; BORRIELLO, 2001; CAPKUN; HAMDI; HUBAUX, 2001; BAGGIO; LANGENDOEN, 2006; SHANG; RUMML, 2004; SAVARESE; RABAHEY; BEUTEL, 2001; SAVVIDES; PARK; SRIVASTAVA, 2002). Posições absolutas são comumente capturadas por sistemas GPS, onde medidas como altitude, longitude e latitude são levadas em conta. Esta informação permite à rede ter disponível a exata posição do nó na superfície da Terra. Contudo, é economicamente dispendioso utilizar receptores GPS em cada um dos nodos que compõem uma rede de sensores. Esta é na verdade uma das razões que motivam a pesquisa científica no campo da localização de nodos, ou seja, relaxar a utilização de dispositivos de auto-localização individual e, ao mesmo tempo, basear a descoberta da localização na cooperação entre nodos.

Um algoritmo de localização baseado em um sistema de coordenadas relativo, por outro lado, constrói seu próprio sistema de coordenadas e localiza os objetos de acordo

com esta referência, focando na descoberta da topologia. Em tais sistemas os nodos da rede constroem mapas locais (OH-HEUM; HA-JOO, 2007) sendo que estes algoritmos de localização são também chamados de *anchor-free* porque os mesmos não necessitam de nodos âncora. Eles são usados onde nodos dotados de capacidade de adquirir posição absoluta não podem ser empregados. A escolha entre estes dois sistemas de posicionamento implica diretamente em custos de comunicação, aplicação alvo, possibilidade de relaxar o quesito exatidão e *hardware* necessário. Considerando ambas, posições relativas e absolutas, redes totalmente móveis, com movimento não controlado de seus elementos introduzem outro problema, uma vez que os pontos de referência da rede precisam ser frequentemente supridos com a informação acerca de sua posição atual.

Na realidade, em alguns casos a opção acerca do emprego do GPS recai sobre uma escolha do usuário final. O planejamento prévio do cenário onde deverá estar presente a rede pode prevenir a necessidade da utilização destes dispositivos. Obviamente existirá um custo relacionado à configuração manual dos nodos âncora com suas posições.

3.2 Redes Móveis com Atuadores para o Movimento

Xiao et al (XIAO; JIN-KUAN; YUN, 2008) propuseram um método *range-free* onde não há cooperação entre os nodos estáticos. Calculando o RSSI de sinais transmitidos por cinco nodos âncora móveis, os nodos estáticos iterativamente restringem possíveis regiões de localização. A área quadrada considerada pelos autores é inicialmente dividida em quatro porções equivalentes. Cada nó estático a ser localizado capta intensidades de sinal de todos os âncoras e inicia a comparação destes sinais. Desta forma, ele infere sua possível localização, auxiliado por interpretações geométricas. Este procedimento é repetido até um quinto e último passo, onde o centro da área final é utilizado para calcular as coordenadas desconhecidas.

Ssu et.al (SSU; OU; JIAU, 2005) e Yu et.al (YU; YU; FENG, 2008) também desenvolveram um método que opera com a ausência de comunicação entre nodos estáticos. Em contrapartida, simularam o movimento de um nó âncora móvel equipado com um receptor GPS. O algoritmo foi primeiramente introduzido por (SSU; OU; JIAU, 2005) e é baseado em uma hipótese geométrica a qual prevê que a bissetriz de uma corda¹ passa através do centro de um círculo. Enquanto que um nó âncora move-se pela área e envia sua informação de posição, nodos sensores a serem localizados recebem os pacotes e julgam os melhores 4

¹Para a Geometria Descritiva uma corda é equivalente a um segmento de reta que une dois pontos pertencentes à circunferência.

pontos para formar duas cordas. A localização destes nodos é obtida pela interseção das duas bissetrizes das cordas. Os pontos selecionados para a definição das cordas devem pertencer ao mesmo círculo para aumentar a precisão do método.

Yu et.al (YU; YU; FENG, 2008) estenderam o trabalho apresentado por (SSU; OU; JIAU, 2005) e consideraram um ambiente ruidoso. Os autores submeteram o algoritmo a condições não ideais de propagação do sinal. Eles utilizaram o fato de que, para distâncias pequenas entre o âncora móvel e o nó sensor, a relação sinal ruído (*signal to noise ratio* - SNR) é maior, ao mesmo tempo que a probabilidade de ocorrerem erros na recepção do sinal é menor.

A cooperação entre nodos estáticos já localizados e os demais a terem a sua posição descoberta foi considerada em (ZHANG; YU, 2008). Assim como em (SSU; OU; JIAU, 2005) e (YU; YU; FENG, 2008), Zhang (ZHANG; YU, 2008) também empregou âncoras móveis durante o processo de localização, mas mediu as distâncias entre nodos com a técnica de TDOA. Um nó sensor precisa coletar 3 mensagens de um âncora em movimento para construir um sistema de equações. Nodos sensores que não possam obter pelo menos 3 pontos não colineares para computar a posição, enviam um requerimento aos seus vizinhos, já localizados, para servirem de nodos âncora estacionários e, portanto, recebem de volta suas posições.

Em (SICHITIU; RAMADURAI, 2004) apenas um único robô é empregado. Por causa disto, os nodos podem levar um tempo adicional para adquirir posições. O movimento do robô precisa garantir que cada nó receba ao menos três mensagens garantindo, sobretudo pontos de referência não colineares. É importante dizer que esta abordagem é factível apenas se o movimento do nó âncora for controlado, uma vez que a cobertura da área sob estudo possui um enorme impacto na eficácia do algoritmo. Desta forma, um diferencial proposto por Sichitiu e Ramadurai (SICHITIU; RAMADURAI, 2004) em seu trabalho foi a diminuição dos custos financeiros da rede de sensores pelo emprego de apenas um único nó móvel e que fosse capaz de adquirir sua própria posição por intermédio de um dispositivo GPS.

A característica de mobilidade em (SHENOY; TAN, 2005) e (SSU; OU; JIAU, 2005) é introduzida pelo movimento dos âncoras. Os nodos desconhecidos estimam as suas posições usando restrições geométricas, o que por sua vez faz desta comunicação uma abordagem *range-free*. Para transpor a desvantagem do aumento da comunicação, os autores em (SHENOY; TAN, 2005) exploram o caminho feito pelo robô e elegem para uma localização precisa apenas nodos que são importantes devido à geração de um evento. Ssu et.al (SSU;

OU; JIAU, 2005) introduziu em seu algoritmo a característica de não precisar interações adicionais para calcular posições, o que é interessante em redes com mobilidade completa, uma vez que isto pode salvar tempo de processamento e comunicação.

Da mesma forma que em (SHENOY; TAN, 2005), Sreenath et.al (SREENATH; LEWIS; POPA, 2006) também se preocupou em minimizar o movimento do robô que executava a tarefa de nó âncora. O objetivo almejado era o de cobrir toda a área de deposição dos nodos, mas com o máximo de eficiência. Em (SREENATH; LEWIS; POPA, 2006) nodos comuns estáticos executam um filtro de Kalman discreto no tempo para estimar suas posições enquanto que um robô, assistido por GPS, disponibiliza para todos os nodos em seu raio de alcance a informação de posição conhecida, movendo-se através da rede.

Cada nó sensor em (SHENOY; TAN, 2005) une as novas mensagens recebidas com a informação anterior para atualizar sua atual estimativa de posição, utilizando uma abordagem *range-free*. Um ponto interessante acerca deste método (SREENATH; LEWIS; POPA, 2006) é a inclusão no estudo o cenário onde há uma cooperação entre os robôs móveis e os já localizados nodos estáticos, para manter atualizada a posição do primeiro na ausência do sinal GPS. À medida que o robô move-se através da rede sua própria posição se deteriora e, frequentemente, a aquisição da informação de um dispositivo GPS é inviável. Neste caso, o algoritmo presente no robô une ambas posições conhecidas provenientes dos nodos sensores mais fielmente localizados (devido a mais numerosa atualização de dados) e a informação GPS, previamente adquirida.

O algoritmo proposto por (LADD et al., 2005) difere das abordagens de (SICHITIU; RAMADURAI, 2004; SHENOY; TAN, 2005; SSU; OU; JIAU, 2005; XIAO; JIN-KUAN; YUN, 2008; YU; YU; FENG, 2008; ZHANG; YU, 2008) pelo fato que estes últimos, Sichitiu e Ramadurai, Shenoy e Tan, Ssu et.al, Xiao, Yu, Zhang, estudaram o cenário oposto, isto é, os nodos a serem localizados são estáticos enquanto que a tarefa de um nodo âncora é realizada por um robô móvel.

Uma rede assíncrona, isto é, onde os *clocks* nos nodos desconhecidos e âncoras são independentes, é proposto em (LI; EKPENYONG; HUANG, 2004). Neste trabalho o movimento dos nodos âncora, possivelmente assistido por atuadores, é explorado para estimar a localização dos nodos. Usando a técnica TDOA, Li et.al (LI; EKPENYONG; HUANG, 2004) se valem de tal característica do movimento para calcular diferentes tempos de chegada de sinais emitidos pelos nodos âncora móveis e recebidos por nodos estáticos. Enquanto que um âncora move-se entre dois pontos, ele dissemina sucessivos pulsos. Pelo fato de tais pulsos serem espaçados por um intervalo de tempo, o qual é um parâmetro de

projeto conhecido, e uma distância d , os nodos desconhecidos sentem uma mudança entre tempos de chegada destes pulsos e fazem cálculos com o tempo local. Depois disto, uma estação mestre recebe tal informação e então estima a posição dos nodos, o que por sua vez pode ser visto como um obstáculo em redes de sensores muito densas. Assim como em (SICHITIU; RAMADURAI, 2004), o algoritmo presente em (LI; EKPENYONG; HUANG, 2004) depende muito do movimento dos nodos âncora para cobrir de maneira apropriada a área de exploração.

3.2.1 Eleição de um Líder

O objetivo de métodos que usam uma abordagem conhecida como eleição de líder (YUAN; CHEN; XI, 2006; MALPANI; WELCH; WAIDYA, 2000; ABBAS; JAIN, 2005) é o de reduzir o consumo de potência baseado na densidade de participação de nodos. Na representação abstrata da Figura 12, os nodos comuns e âncoras são indicados por gorilas, onde o líder normalmente é um espécime dominante no grupo. Este tipo de algoritmo é suposto desempenhar três tarefas principais: eleição de um líder (*leader election*), gerenciamento da conectividade entre nodos e da comunicação. Alguns nodos recebem o título de líder e outros de não-líder. Um nó líder possui mais recursos, os quais podem ser dados sobre a rede, comparado a um nó não-líder. Este último procura formar grupos ou *clusters* com o primeiro para aumentar o aspecto de cooperação. Na realidade, os nodos podem cooperar entre eles mesmos no sentido de auto-organizar a rede, por exemplo, movendo-se para formar um *cluster*, ou ainda, optar por manter-se estacionários para não perder a conexão com o líder e, eventualmente, quebrar uma rota de comunicação com destino ao mesmo.

Este exemplo também intui o aspecto da auto-organização da rede, como já mencionado anteriormente. Esta auto-organização deve levar em consideração a relação de compromisso que envolve a disponibilidade de recursos do nó. Tais recursos não estão apenas limitados ao *hardware*, como energia residual ou sensores especializados. A conectividade da rede também deve ser um ponto de decisão.

O exposto acima indica que a eleição de um líder é apropriada para redes robotizadas. De fato, o movimento controlado dos nodos atribui muitas vantagens a este tipo de abordagem. Contudo, pode-se fazer um paralelo com a vida selvagem. Um líder pode ser interpretado como um macho ou uma fêmea alfa. A Figura 12 assume um cenário onde não há controle dos dispositivos sensores sobre o movimento. Nesta situação, a rede deve ser auto-organizável e capaz de interpretar padrões de movimento de forma a elevar um

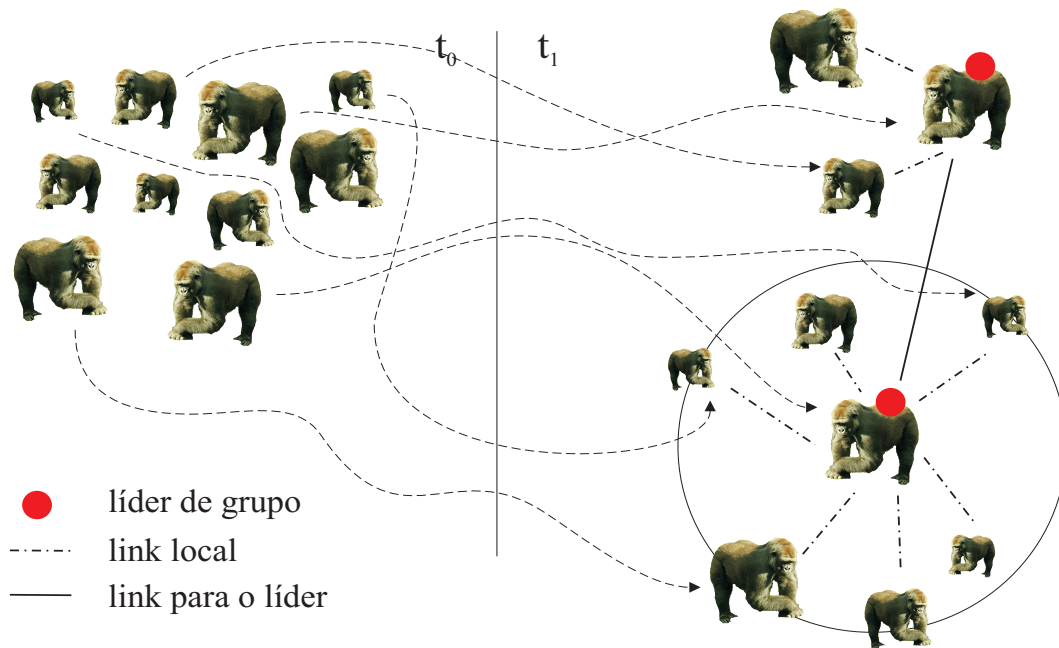


Figura 12: Exemplo abstrato da escolha de um líder, baseada na programação do algoritmo em reconhecer nodos com maior quantidade de informação, por exemplo.

nó comum a um âncora. Um nó líder, neste caso, é naturalmente seguido de perto pelos integrantes do seu grupo. Em um cenário como este, a rede poderia desativar por longos períodos a execução da localização nestes integrantes do grupo, reduzindo o consumo de potência destes nodos.

Em (JUANG; OKI; WANG, 2002), a estrutura social de zebras (RUBENSTEIN, 1994), bem como o padrão de seu movimento em vários períodos do dia e sob diferentes situações foi analisada. Este estudo serviu para os cientistas balisarem a seleção de animais que deveriam ser rastreados. Mudanças climáticas possivelmente capturadas por sensores especiais também refletem na dinâmica de todo o grupo e, portanto, auxiliam na escolha do líder. Neste caso, o aspecto de eleição de um líder foi abordado de maneira manual em uma pesquisa biológica e levou em consideração a melhor opção para monitorar o comportamento de uma população.

3.3 Método Monte Carlo

O método MCB ou *Monte Carlo Localization Boxed* (BAGGIO; LANGENDOEN, 2006) é um algoritmo proposto para manipular redes de sensores sem fio. Ele foi construído sobre o MCL (*Sequential Monte Carlo Localization*) (HU; EVANS, 2004), o qual primeiramente introduziu, pelo melhor conhecimento do autor, o problema de lidar com um cenário onde ambos nodos comuns e nodos âncora estão em movimento sem atuadores para tal ação.

O algoritmo MCL é baseado na inferência de Bayes. Thomas Bayes foi um matemático (século XVI) que apresentou a regra de Bayes, um método relacionado com valores esperados ao invés de frequências absolutas de acontecimentos para calcular a probabilidade de um evento. A inferência Bayesiana (LADD et al., 2005; HARTL; LI, 2005) mostra como relacionar eventos aleatórios, baseados na coleta de evidências ou observações que são utilizadas para verificar se a hipótese é verdadeira. Tais hipóteses recebem pesos ou graus de importância de acordo com as observações. Considerando o MCL, a inferência de Bayes, ou inferência estatística, possui dois passos principais:

- predição;
- filtragem.

Na etapa de predição algumas possibilidades são consideradas para se construir uma distribuição posterior² baseada em um conjunto de predições prévias, enquanto que na fase de filtragem, tais predições são avaliadas e aquelas que não se enquadram nas restrições escolhidas são filtradas. A posição final estimada é a média de todas as predições. Recentemente (STEVENS-NAVARRO; VIVEKANANDAN; WONG, 2007) e (HSIEH; WANG, 2006) também apresentaram uma variação do MCL. Contudo, tanto estes dois últimos trabalhos quanto aqueles desenvolvidos por (STEVENS-NAVARRO; VIVEKANANDAN; WONG, 2007) e (HSIEH; WANG, 2006) têm em comum a preocupação com a localização considerando a mobilidade da rede de sensores, além de assumir que os nodos âncora possuem conhecimento de suas posições geográficas utilizando dispositivos GPS. Este cenário com mobilidade total, isto é, onde nodos comuns e âncoras não são estacionários, mas movem-se de forma aleatória, introduz uma componente tempo real, ou seja, o momento da aquisição da informação passa a fazer parte dos algoritmos. Isto faz com que uma rede de sensores móveis não seja tolerante ao atraso e, desta forma, leva à necessidade de redução do tempo de computação de maneira conjunta com o esforço de tornar este cálculo o mais simples possível.

Particularmente a questão da redução do tempo gasto pelo algoritmo para computar as estimativas foi abordada por (BAGGIO; LANGENDOEN, 2006). Foi proposto que os nodos construíssem caixas virtuais para restringir a área de captação das amostras necessárias para executar o processo de localização. A exatidão da localização foi mantida similar a (HU; EVANS, 2004), ao passo que o tempo de computação foi reduzido. Esta preocupação não foi observada em Hu et.al (HU; EVANS, 2004) e Navarro et.al (STEVENS-NAVARRO; VIVEKANANDAN; WONG, 2007).

²Probabilidade condicional obtida através de dados empíricos.

O ponto chave do algoritmo desenvolvido em (HSIEH; WANG, 2006) é o controle da disseminação da informação iniciada pelos nodos âncora e propagada pelo restante da rede, com aumento da contagem de *hops*. Este processo é comumente chamado de *flooding*. Este controle impõe algumas restrições para o processo de localização dos nodos desconhecidos. Uma vez que o cenário proposto por Yi-Ling e Wang (HSIEH; WANG, 2006) apresenta nodos comuns e nodos âncora movendo-se de maneira arbitrária, é factível considerar diferentes números de nodos âncora disponíveis ao passar do tempo para executar a triangulação. No pior cenário, isto é, quando não existe nenhum nó de referência disponível, o algoritmo lança mão do que os autores denominaram por *backup* inteligente. A propagação das mensagens dos nodos âncora (*flood*) cobre uma área limitada e, em alguns casos, há nodos não cobertos por um nó de referência. De posse desta situação, tais nodos devem requerer de seus vizinhos diretos a informação de posição dos âncoras, e estes a outros vizinhos, até que a solicitação seja satisfeita.

Na Tabela 2 o algoritmo exposto em (DIL; DULMAN; HAVINGA, 2006) também utiliza o método de Localização Monte Carlo. Em contraste, o método proposto por Dil et.al (DIL; DULMAN; HAVINGA, 2006) empregou medidas de distância *range-based* em sua abordagem em conjunto com informação *range-free*, com o propósito de reduzir o erro no algoritmo. Isto é particularmente interessante porque os autores lidam com a relação de compromisso entre erros de medida de distância e os custos computacionais do processamento da informação. Assim como (BAGGIO; LANGENDOEN, 2006), Dil et.al tentaram antecipar alguma avaliação da informação para a função de amostragem com o intuito de reduzir custos computacionais devido à diversas interações na fase de filtragem. Os autores utilizaram o mesmo modelo de mobilidade empregado por (HU; EVANS, 2004) e consideraram medições baseadas na distância, nomeadamente RSSI³. As abordagens *range-based* podem alcançar resultados mais exatos no processo de localização quando comparadas a abordagens *range-free*. Entretanto, os custos envolvidos para cada nó também podem ser maiores para o primeiro caso. Além disto, o desempenho de abordagens *range-based* não se comportam de maneira satisfatória em um ambiente anisotrópico.

Apesar de Dil et.al (DIL; DULMAN; HAVINGA, 2006) ter alcançado um nível onde os custos computacionais são mais constantes para todas as situações e o desempenho do algoritmo é menos dependente da velocidade do nó, isto pode ser uma desvantagem quando são levados em consideração diferentes padrões de mobilidade (CAMP; BOLENG; DAVIES, 2002). Não foi observada uma preocupação acerca da exatidão das medidas de distância recebidas e da análise sobre o cenário onde os âncoras estão em movimento.

³Descrito na Seção 2.2.2.4.

Dil et.al também compararam sua proposta com o algoritmo *Ad Hoc Positioning System* ou APS (NICULESCU; NATH, 2001). A comparação levou em consideração o método de propagação *range-free DV-hop*, onde as distâncias para os nodos de referência são obtidas em termos de *hops*. Niculescu et.al considerou alterações na topologia da rede em sua avaliação, mas apenas como uma hipótese, não como uma ação. A idéia fundamental do algoritmo DV-hop é a de que os nodos adquirem as distâncias em número de *hops* para os âncoras. Uma vez que estes âncoras, que estão equipados de dispositivos GPS, possuem um conhecimento prévio sobre suas posições físicas, as distâncias podem ser convertidas de *hops* para estimativas em metros, por exemplo.

Diferentemente de (DELLAERT et al., 1999) e (LADD et al., 2005), Capkun et.al (CAPKUN; HAMDI; HUBAUX, 2001) consideraram que os nodos possuem movimento aleatório, construindo seu próprio sistema de coordenadas, apropriado para um determinado conjunto de aplicações. De fato, construir um sistema de localização pode evitar que muitos nodos da rede precisem ser equipados com dispositivos GPS, em contrapartida, há um custo computacional a ser avaliado neste caso. Ao invés de analisar movimentos aleatórios dos nodos, Dellaert et.al (DELLAERT et al., 1999), também baseado sobre o MCL, utilizaram apenas robôs na rede, contendo um mapa predefinido na memória, enquanto que em (LADD et al., 2005) robôs precisam adquirir a localização de nodos âncora estáticos e aprender o mapa de uma área interna.

Cheng et.al (CHENG et al., 2004) propuseram um algoritmo de localização baseado na TDOA, o qual utiliza as diferenças nos períodos de chegada dos sinais a partir dos nodos âncora e medidos pelo nó desconhecido. Todos os tempos de chegada no Esquema de Posicionamento Baseado no Tempo (CHENG et al., 2004) (TPS) são medidos localmente sem nenhuma sincronização global entre eles. A sincronização empregada no trabalho envolve a utilização de uma base de tempo única entre os nodos da rede. No esquema TPS, um nó âncora A_1 chamado de nó mestre é responsável por iniciar o processo de localização pelo envio de um sinal base que alcança o nó desconhecido U no tempo t_1 e outro nó âncora A_2 no tempo t_{A2} . O nó âncora A_2 por sua vez responderá no tempo local A_2 o valor t'_{A2} a A_1 com o sinal base mencionando a informação $t'_{A2} - t_{A2}$. Este sinal então alcançará U no tempo t_2 . A Equação 3.1 resume as relações mencionadas acima. Pelo menos três ou mais equações são necessárias para acomodar o terceiro nó âncora e realizar a trilateração (CHENG et al., 2004).

$$d_{A1A2} + d_{UA2} - d_{UA1} + c \cdot (t'_{A2} - t_{A2}) = c \cdot (t_2 - t_1) \quad (3.1)$$

onde $d_{A_1A_2}$ é a distância entre os nodos âncora A_1 e A_2 , d_{UA_1} e d_{UA_2} são as distâncias desconhecidas de U a A_1 e A_2 , respectivamente.

O algoritmo TPS (CHENG et al., 2004) executa a trilateração, mas diferentemente de (LI; EKPENYONG; HUANG, 2004), os nodos sensores calculam de forma independente suas posições depois de ouvirem três sinais de referência durante cada intervalo de tempo definido. Esta característica distribuída é uma vantagem em redes de sensores constituídas por muitos dispositivos⁴ desde que minimiza o impacto da comunicação no desempenho do sistema. O esquema TPS possui dois estágios: a detecção da distância e o cálculo da posição. No primeiro passo um nó âncora fixo chamado de nó mestre inicia o processo de comunicação e algumas mensagens são trocadas entre nodos âncora estacionários e o nó desconhecido. Este nó desconhecido é responsável por medir todas as diferenças em tempos de chegada localmente, sem uma sincronização global, e transformá-las em medidas de distância. No segundo passo, a trilateração é executada para estimar as coordenadas geométricas.

3.4 Mobilidade Aleatória

Nodos comuns e âncoras movem-se de maneira aleatória no algoritmo com mobilidade assistida MA-MDS-MAP(P) (WU; SHENG; SONG, 2006). Todavia os autores ainda fazem algumas ponderações acerca da efetividade do algoritmo para redes de sensores realmente móveis. Wu et.al (WU; SHENG; SONG, 2006) adaptaram o algoritmo MDS-MAP(P) (SHANG; RUMMLER, 2004) e contornaram um de seus principais fundamentos: o menor caminho entre dois nodos é aproximadamente proporcional à distância Euclidiana entre os mesmos. Isto não é válido para nodos móveis não controlados. O conceito de adicionar nodos virtuais é explorado por Wu et.al (WU; SHENG; SONG, 2006) de tal maneira que isto é incluso à mobilidade dos nodos quando eles se movem diretamente em uma direção e tentam retornar à sua posição de origem. Na realidade o modelo de mobilidade empregado em (WU; SHENG; SONG, 2006), nomeadamente modelo randômico de mobilidade de caminhada, mantém conhecimento acerca de localizações passadas e valores de velocidade. É claro que não existe comunicação entre um nó virtual e um nó real, mas a informação de posição retida provê dado adicional para executar uma localização de maneira mais precisa.

Como mencionado, algoritmos tradicionais baseados na informação de conectividade,

⁴Este número é relativo e pode variar desde algumas dezenas a milhares de nodos. Depende ainda dos recursos disponíveis para cada nó.

como o método DV-Hop (NICULESCU; NATH, 2001) e o Centroid (BULUSU; HEIDEMANN; ESTRIN, 2000), valem-se da disseminação da informação entre os nodos para estimar posições. Quando uma mensagem atingir um determinado nó que é importante no processo de estimar a distância média de um único *hop*, é muito provável que toda a configuração da rede tenha sido alterada. O nó, portanto, em uma abordagem de computação distribuída irá sempre calcular uma localização imprecisa. Este efeito não será devido à falta de informação ou imprecisões intrínsecas ao tipo de computação empregada, mas devido à maneira como este algoritmo consegue esta informação. Isto constitui uma perda da qualidade da informação com o tempo.

Uma comparação quantitativa acerca do problema de determinar a localização do nó em uma rede de sensores *ad-hoc*, focando no aspecto estático, pode ser encontrada em (LANGENDOEN; RELJERS, 2003). Enquanto que muitos pesquisadores concentram esforços apenas sobre a precisão do algoritmo de localização, Tilak et.al (TILAK et al., 2005) atentaram para padrões de movimentos. Nesta abordagem interessante, Tilak et.al foram motivados pelo projeto ZebraNet (JUANG; OKI; WANG, 2002), onde os nodos são zebras, e se concentraram em diferentes políticas para determinar a execução do algoritmo de forma a prover uma localização eficiente sob o ponto de vista da energia gasta. De posse disto, alguma reflexão precisa ser feita sobre questões envolvendo se uma **elevada** precisão ou **suficiente** precisão são requeridas para ultrapassar alguns obstáculos e tornar a aplicação factível. Esta relação de compromisso fustiga a pesquisa e desenvolvimento dos algoritmos *range-based* e *range-free*.

A maioria dos trabalhos sobre algoritmos de localização foca em idéias puramente teóricas ou avança até o ponto de simulações com muitas premissas ideais levadas em consideração (BAGGIO; LANGENDOEN, 2006; DIL; DULMAN; HAVINGA, 2006; HU; EVANS, 2004; CAPKUN; HAMDI; HUBAUX, 2001). Estas premissas incluem áreas livres de obstáculos; inexistência de imprecisão devido ao constante movimento e atrasos na comunicação; nodos com mesmos raios de alcance sem perdas de mensagens ou erros de GPS para nodos âncora. Alguns poucos fazem uso de dispositivos físicos para expor ao mundo real seus métodos. Em contraste, Tufan et.al (KARALAR et al., 2004) incluíram um parâmetro físico através da integração em Silício do método estudado. Neste trabalho os autores investigam uma solução dedicada em *hardware* de um sistema de localização para redes de sensores baseado na triangulação (SAVARESE; RABAEY; LANGENDOEN, 2002). De fato, os autores em (KARALAR et al., 2004) não focaram o aspecto da mobilidade, mas pesquisaram o algoritmo apropriado para permitir uma implementação em *hardware* e a conseqüente redução da energia empregada na operação. Por outro lado, este tipo de abordagem, aliada a um

determinado grau de flexibilidade, mostra ser uma alternativa bastante plausível (RABAEY et al., 2000), ainda mais quando se considera que tal procedimento de localização em uma rede móvel é repetido inúmeras vezes.

Mais detalhes sobre algoritmos de localização baseados em abordagens *range-free* mas que não exploram aspectos de mobilidade podem ser encontradas nos seguintes trabalhos (JI; LIU, 2006; RISTIC et al., 2006; WU; LIU; KING, 2005; TAY; CHANDRASEKHAR; SEAH, 2006; HUANG; SELVAKENNEDY, 2006; SUN; YU; MEI, 2006; LIU; WU, 2005; MENDOZA; SILVA; DIAZ, 2005). Por esta razão, estas últimas referências não estão expostas na Tabela 2. Por exemplo, em (LIU; WU, 2005), a abordagem estática está em evidência quando os autores investigam diferentes estratégias de depositar os nodos, aleatória e uniformemente, para melhor investigar o algoritmo APIT (HE et al., 2003) (*Approximate Point in Triangle*), o qual também necessita de nodos âncora equipados com GPS. Abaixo está reunido um conjunto de importantes características que um sistema móvel deve cobrir:

- computação distribuída: excluindo (SHANG; RUMMLER, 2004), todos os outros métodos presentes na Tabela 2 recaem no fato de que cada nó está executando os cálculos para determinar a sua posição. Ao contrário de redes *ad-hoc* tradicionais, em uma rede de sensores móvel, onde o número de sensores pode ser diversas ordens de magnitude maior que os nodos em uma rede *ad-hoc*, a idéia de uma computação centralizada pode elevar os custos de comunicação a tal ponto de inviabilizar a operação da rede, dado o consumo de potência elevado e mobilidade dos nodos;
- *range-free*: isto ainda é um ponto em aberto e está presente em cerca de 50% dos métodos investigados. De fato, abordagens baseadas na medição pura da distância fornecem resultados de medição relativamente mais precisos;
- capacidade de manipular redes móveis com diferentes padrões de movimentos, controlados ou não para nodos âncora e simples. Isto é ainda mais imperativo para aplicações que visem nodos sem atuadores para o movimento;
- elegibilidade de um líder: esta característica pode auxiliar a adquirir padrões de movimento;
- manter informação sobre a história de movimentação dos nodos suficiente⁵ para a localização;

⁵A suficiência referida aqui é relativa e está ligada à quantidade de memória disponível em *hardware* no nó sensor, por exemplo.

- exatidão suficiente: esta característica é basicamente relacionada com o tipo de comunicação empregada, *range-free* ou *range-based*, sendo que um bom exemplo de como lidar com isto pode ser encontrado em (HIGHTOWER; BORRIELLO, 2001);
- baixo consumo de potência: na maioria das vezes isto não está em primeiro plano ou não é bem enfatizado nos algoritmos estudados, mas é imperativo para a sobrevivência da rede.

3.5 Resumo

Este Capítulo expôs uma revisão acerca dos algoritmos de localização de nodos mais representativos presentes na literatura. Foram apresentados algoritmos que operam em cenários onde os nodos possuem atuadores para o movimento. Salientou-se a preocupação em minimizar o tempo de computação total necessário para entregar uma estimativa de localização, bem como a classificação temporal das informações recebidas e calculadas pelos nodos da rede, ou seja, a informação deteriora-se com o tempo. Ao final do Capítulo foram reunidas diversas características importantes para algoritmos de localização de nodos direcionados a redes móveis.

4 *ALGORITMO PROPOSTO*

Os algoritmos de localização referenciados no Capítulo 3 descrevem diversos métodos para a estimativa das coordenadas de um nó. Algumas destas abordagens focam na cooperação entre os nodos da rede, enfatizada pela troca de mensagens ou na mobilidade assistida de determinados nodos especializados.

Após estudar as alternativas de algoritmos para redes de sensores, o algoritmo Centroid foi escolhido como base para a implementação das modificações a serem propostas. Este algoritmo utiliza apenas o conteúdo das mensagens enviadas pelos nodos âncora para estimar localmente as coordenadas do nó. Neste Capítulo será apresentado em detalhes o algoritmo Centroid (BULUSU; HEIDEMANN; ESTRIN, 2000), bem como as razões que levaram à escolha deste método como base para o desenvolvimento do algoritmo desta Tese. Também serão apresentadas as suas limitações para um cenário de mobilidade.

O método proposto mantém a idéia original do Centroid. A estimativa final é tratada como o centro das coordenadas dos âncoras selecionados, exatamente como o Centroid original, exceto para os termos utilizados nesta operação, bem como o processo de seleção de âncoras. Será apresentada a flexibilidade do método proposto para a inserção em ambientes cuja mobilidade da rede está presente, possibilitando ainda o aumento da precisão dos resultados tanto no cenário estático quanto no móvel.

4.1 *Algoritmo Centroid*

O algoritmo Centroid (BULUSU; HEIDEMANN; ESTRIN, 2000) é um método de localização de nodos em redes de sensores utilizado para estimar as coordenadas de um nó através da conectividade deste em relação a nodos de referência. Este tipo de algoritmo assume que os nodos que podem se comunicar possuem uma determinada implicação relativa às suas posições.

Neste método, um número pré-determinado de nodos especializados, chamados de

nodos âncora (A_i), possuem conhecimento acerca de sua localização. Este conhecimento pode ser devido ao fato de eles possuírem um dispositivo GPS acoplado ou, por serem configurados previamente, quando da sua deposição na rede. Tais nodos âncora são responsáveis por enviar mensagens periodicamente em um tempo T , também chamadas de *beacons*, cujo conteúdo é composto pelas suas coordenadas (XA_i, YA_i) , além de sua identificação i na rede. Estas mensagens não possuem um destinatário específico, ou seja, são enviadas em *broadcast*, e podem ser captadas por qualquer nó que esteja no raio de alcance R de seu rádio.

Os nodos a serem localizados, por sua vez, recebem estas mensagens e iniciam o seu tratamento. Durante um período de tempo t cada mensagem recebida dispara um contador interno, o qual é incrementado de acordo com o remetente da mensagem. Ao mesmo tempo as coordenadas recebidas também são identificadas ao nó âncora que as enviou. Não é necessário manter uma tabela prévia com as coordenadas de cada âncora. Este ponto em particular aumenta a escalabilidade do método. Cada nó âncora é suposto de transmitir um número restrito de mensagens, chamado de tamanho da amostra S durante o período de amostragem t .

Ao término do tempo t de amostragem ou tempo coleta de dados, um processo de estimativa da localização é disparado e os dados armazenados durante todo este período são utilizados para estimar as coordenadas do nó. O primeiro passo é calcular as métricas de conectividade CM_i , já descritas na Equação 2.6. Todos os nodos âncora que excederem a sua CM_i de um determinado valor limiar $CMthresh$, definido pelo usuário, farão parte do cálculo final das coordenadas estimadas (X_{est}, Y_{est}) dadas pela Equação 4.1, extraída de (BULUSU; HEIDEMANN; ESTRIN, 2000). Desta forma, o nó localiza-se de acordo com a intersecção das regiões de conectividade do conjunto de pontos de referência selecionado, que é definida como sendo o **centro** destes pontos de referência. Os termos mencionados estão reunidos logo a seguir.

$$(X_{est}, Y_{est}) = \left(\frac{XA_{i_1} + \dots + XA_{i_k}}{k}, \frac{YA_{i_1} + \dots + YA_{i_k}}{k} \right) \quad (4.1)$$

onde k é o número de nodos âncora cuja CM_i ultrapassou $CMthresh$.

R	raio de alcance do rádio
d	distância de separação entre dois nodos âncora adjacentes
T	intervalo de tempo entre dois <i>beacons</i> sucessivos transmitidos pelos nodos âncora
t	tempo de amostragem dos dados ou tempo de coleta
A_i	nó âncora
(X_{A_i}, Y_{A_i})	coordenadas do nó âncora A_i
$Nsent(i, t)$	número de <i>beacons</i> que foram enviados pelo nó âncora i durante o tempo t
$Nrecv(i, t)$	número de <i>beacons</i> transmitidos pelo nó âncora i e recebidas durante o tempo t
CM_i	métrica de conectividade simples calculada para o nó âncora A_i
S	tamanho da amostra para CM_i
$CMthresh$	valor limiar (threshold) para CM_i
(X_{est}, Y_{est})	localização estimada pelo nodo
(X_a, Y_a)	localização verdadeira do nodo

O algoritmo Centroid pode ser ajustado para diversas situações. Estes ajustes levam em conta as necessidades da aplicação que está sob estudo. Os parâmetros descritos acima que podem ser configurados são T , t e S ¹. Também podem ser alterados obviamente o R , mas depende do *hardware* empregado, e o (X_{A_i}, Y_{A_i}) através da alteração proposital da topologia dos nodos de referência. A relação entre o tempo de amostragem t , o tamanho da amostra S e T é dada pela Equação 4.2².

$$t = (S + 1 - \epsilon) \times T (0 < \epsilon \ll 1) \quad (4.2)$$

Ao contrário de muitos algoritmos de localização baseados na medição da distância, *range-based*, entre o nó a ser localizado e um grupo de nodos âncora, o Centroid opera com o cálculo do centro de regiões geométricas retangulares. A vantagem deste ponto é que calcular estes centros é uma operação matematicamente mais simples que a interseção de

¹Embora sejam parâmetros configuráveis, a escolha de valores para T , t e S , não pode ser feita de maneira arbitrária, pois há uma relação de dependência entre eles.

²O termo ϵ é muito próximo de zero, no entanto, não está descrito em (BULUSU; HEIDEMANN; ESTRIN, 2000). A autora de (BULUSU; HEIDEMANN; ESTRIN, 2000) também não retornou ao questionamento acerca do significado de ϵ .

círculos. A desvantagem é que o Centroid torna-se dependente da sobreposição do alcance de rádio dos pontos de referência, isto é, a granularidade das regiões de localização como mostrado na Figura 13. Desta forma, a deposição dos nodos de referência tem um papel preponderante nas estimativas de localização entregues pelo algoritmo.

A baixa complexidade computacional do Centroid fica evidente quando se analisam dois trabalhos de referência no âmbito da localização para redes de sensores móveis: (HU; EVANS, 2004) e (BAGGIO; LANGENDOEN, 2006). Ambos métodos empregam a conectividade entre os nodos como princípio de estimativa da localização. Basta dizer que o algoritmo proposto por Hu e Evans (HU; EVANS, 2004), o MCL, gasta aproximadamente 10.000 vezes mais tempo para entregar uma estimativa de posicionamento em relação ao Centroid (BULUSU; HEIDEMANN; ESTRIN, 2000). Baggio (BAGGIO; LANGENDOEN, 2006) baseou a sua abordagem sobre (HU; EVANS, 2004) com o objetivo de reduzir substancialmente este tempo de processamento e manter as características de precisão do MCL considerando a mobilidade da rede. Ainda assim o MCB (BAGGIO; LANGENDOEN, 2006) é cerca de 6.000 vezes mais lento que o Centroid. Além disto, os autores em (HU; EVANS, 2004) e (BAGGIO; LANGENDOEN, 2006) não consideraram o tempo envolvido com a troca de mensagens entre os nodos vizinhos quando propuseram os tempos de computação de seus algoritmos.

O método Centroid (BULUSU; HEIDEMANN; ESTRIN, 2000) possui diversas vantagens que o torna até os dias de hoje um método tomado como parâmetro para comparações e/ou extensões (CHEN et al., 2008a; DENG et al., 2008; LIU; ZHANG; HU, 2008; WANG; LIU; XIAO, 2008; BAGGIO; LANGENDOEN, 2006; REICHENBACH; BLUMENTHAL; TIMMERMANN, 2006; VIVEKANANDAN; WONG, 2006; SSU; OU; JIAU, 2005; BLUMENTHAL; REICHENBACH;

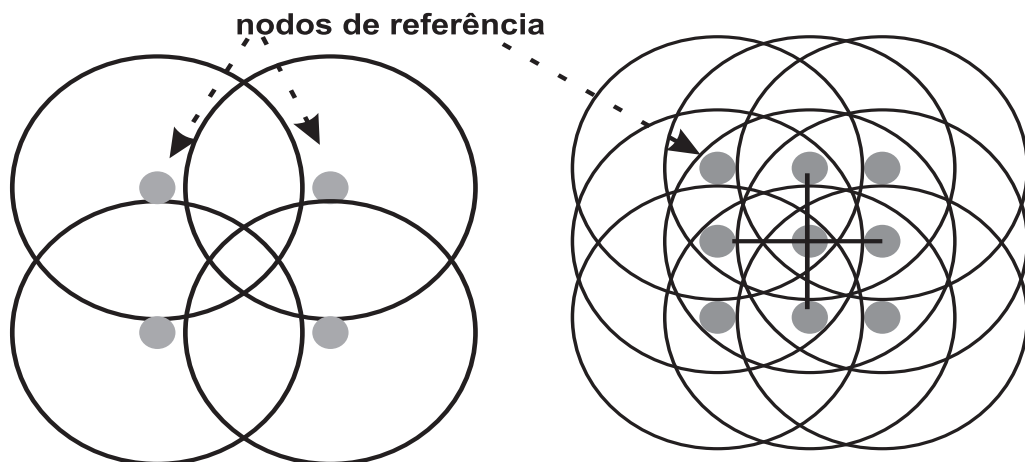


Figura 13: Diferentes regiões de localização baseadas na granularidade. Adaptado de (BULUSU; HEIDEMANN; ESTRIN, 2000).

TIMMERMANN, 2005; HU; EVANS, 2004) pela comunidade científica. Podem-se resumir tais vantagens a seguir:

- estimativas de localização são baseadas apenas no recebimento das mensagens. Isto possibilita aos nodos não precisarem trocar mensagens com nodos vizinhos, economizando energia na transmissão de dados;
- o item acima também interfere positivamente no tempo de convergência das estimativas de localização, uma vez que apenas os dados dos nodos âncora serão analisados. Desta forma, elimina-se um tempo eventualmente gasto na troca de mensagens com outros nodos;
- mensagens pequenas. O Centroid recebe apenas mensagens que contenham a identificação do nó âncora transmissor e as suas coordenadas;
- baixa complexidade computacional. O Centroid não lida com equações complexas ou problemas estatísticos. Isto por sua vez diminui de forma considerável o tempo de processamento dos dados.

De forma contrária a outros métodos na literatura, como por exemplo em (HU; EVANS, 2004), não existe a necessidade de um período de treinamento para que o algoritmo estabeleça o menor erro de localização. Desde a primeira estimativa o algoritmo já é capaz de inferir o melhor resultado. Este fato é também devido a não necessidade do Centroid realizar comunicação com seus vizinhos, sejam eles nodos comuns ou nodos âncora. Além disto, a informação recebida em um único período de amostragem é suficiente para fornecer ao algoritmo todos os dados de que ele necessita para produzir uma estimativa de localização.

4.2 CentroidM

4.2.1 Limitações do Centroid

O algoritmo Centroid não foi concebido com a mobilidade da rede em mente. O aspecto da mobilidade em redes de sensores recai diretamente sobre a mudança das interconexões entre os nodos da rede. Desta forma é provável que a topologia de uma rede móvel sofra alterações com uma frequência muito maior que em uma rede dita estática. Este tipo de comportamento faz com que a informação obtida por um nó da rede em um

determinado instante passe a não ter mais a mesma relevância pouco tempo depois, uma vez que a posição geográfica do nó se alterou no decorrer do tempo.

Existem duas características principais que tornam o Centroid incompatível para a sua utilização em redes de sensores móveis. A **primeira** diz respeito à etapa de seleção dos nodos âncora. Como descrito na Seção 4.1, apenas as coordenadas do nó âncora para o qual o seu valor computado de CM_i ultrapassar CM_{thresh} serão incluídas na etapa final de estimativa da posição. Esta seleção, apesar de objetivar eleger apenas nodos de grande relevância, pode impossibilitar a estimativa final em um ambiente ruidoso ou mesmo considerar nodos de menor importância em um cenário de movimento, o que resulta em uma estimativa de menor precisão.

A Figura 14 apresenta uma topologia de rede de sensores composta por 5 nodos âncora estáticos e um nó móvel. As circunferências representam os raios de alcance das mensagens transmitidas pelos nodos âncora e são supostos serem iguais entre si, bem como o alcance de recepção do nó móvel. As regiões descritas na Figura 14 estão configuradas da seguinte forma:

- região 1: composta pela intersecção das regiões delimitadas pelos raios de alcance dos âncoras 1 e 4;
- região 2: composta pela região delimitada pelo raios de alcance do âncora 4 excluindo as intersecções com as regiões delimitadas pelos raios de alcance dos âncoras 1 e 3;
- região 3: composta pela intersecção das regiões delimitadas pelos raios de alcance dos âncoras 2 e 4.

Uma vez que o tempo de amostragem é iniciado, o algoritmo Centroid começa a armazenar as mensagens recebidas. Isto acontece ao mesmo tempo em que o nó se movimenta segundo a trajetória descrita na Figura 14. O nó finaliza o seu movimento em t_1 , quando também é disparado um procedimento de cálculo. O que ocorre neste caso é que, dependendo da velocidade do nó móvel e de sua trajetória, o número de mensagens recebidas do âncora 1, no início do período de amostragem t , será maior que o número de mensagens recebidas do âncora 2. Ao final do movimento, e do tempo de amostragem, o nó está chegando à região 3 e, portanto, não receberá o número suficiente de mensagens que possibilite que a CM_i computada para o âncora 2 ultrapasse a CM_{thresh} estabelecida para o cenário. Isto fará com que, apesar do nó estar mais próximo do âncora 2, a posição estimada será estabelecida entre os âncoras 1 e 4.

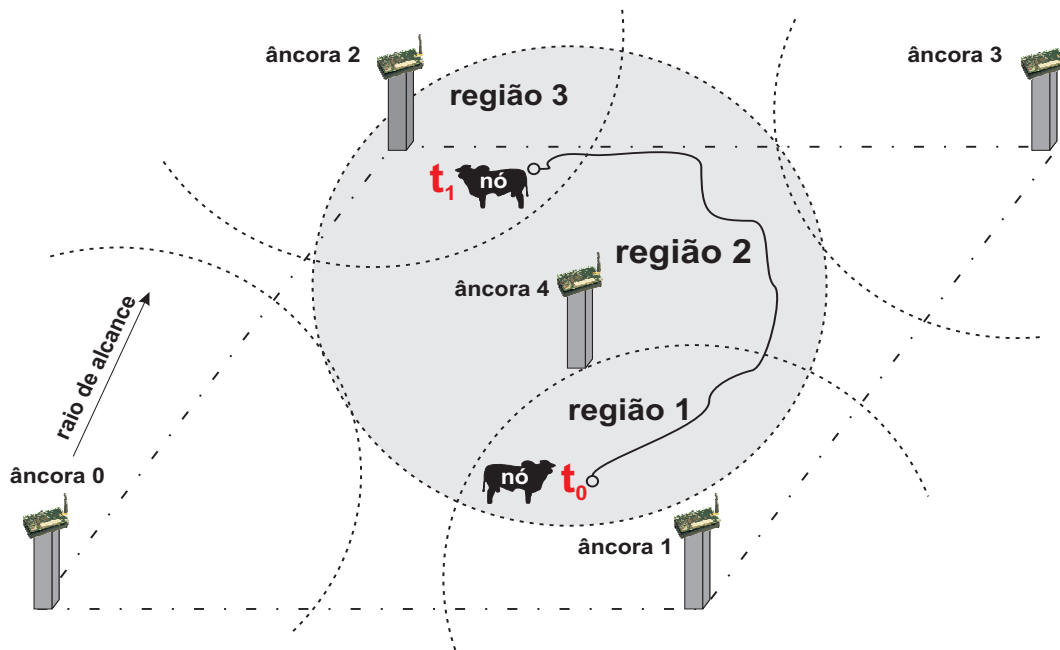


Figura 14: Representação de uma topologia de rede de sensores com um nó móvel e 5 nodos âncora. O nó móvel recebe mensagens de pontos de referência distintos no decorrer do tempo.

O **segundo** ponto negativo do Centroid é relacionado à sua precisão. Embora seja considerado o fato de o método dispor de um mecanismo de seleção de nodos âncora eficiente para redes estáticas, o espaço total de resultados possíveis para as estimativas de localização é limitado a algumas combinações de coordenadas dos nodos âncoras que compõem a rede. Este fato engessa os resultados e provoca que pontos de possíveis localizações dentro da área estudada nunca sejam atingidos.

Enquanto que os algoritmos para redes de sensores estáticas são tolerantes a atrasos na computação dos dados ou aquisição da informação, a mobilidade torna a rede de sensores mais seletiva sob este ponto de vista. A informação, ou mesmo a qualidade desta, passa a ter uma importância relativa ao tempo na qual a mesma foi adquirida e dependente da velocidade dos integrantes da rede.

O Centroid foi inicialmente concebido no final do século passado para operar em redes estáticas. Sua utilização em redes móveis não foi completamente desenvolvida. Apesar disto, devido à sua eficiência computacional, este método foi utilizado como ponto de comparação em trabalhos que objetivavam a alteração não controlada da topologia. Diante deste cenário apresentado, advém a idéia principal do estudo desta Tese, qual seja, a extensão do Centroid para redes de sensores sem fio móveis aumentando a precisão do algoritmo para ambos os casos, móvel e para a topologia estática.

Em contrapartida ao acima mencionado, duas características expostas na Seção 4.1 tornam o Centroid um método extremamente atrativo para o desenvolvimento em redes de

sensores móveis. Esta ambiguidade é devida ao fato do Centroid ser um método baseado apenas na informação recebida, sem a necessidade de trocar mensagens com os nodos vizinhos, além de ser computacionalmente leve, o que vai diretamente ao encontro das características desejadas para uma rede de sensores móvel.

4.2.2 Alterações Propostas ao Algoritmo Centroid

O algoritmo proposto neste trabalho, ao qual denominamos de **CentroidM**, visa extrapolar as características iniciais do método de localização Centroid para englobar o aspecto da mobilidade e aumentar a precisão das estimativas de localização, tanto no caso estático quanto para uma topologia móvel. Para atingir estes objetivos é necessário focar em dois³ pontos cruciais do Centroid que limitam a sua aplicabilidade em redes de sensores móveis. Além disto, prioriza-se manter a precisão original do método em redes estáticas. Nós assumimos que o nó pode apenas julgar se está dentro ou fora do alcance de rádio com raio r dos nodos âncora, mas não é capaz de obter outras informações mais precisas, como por exemplo a medição direta da distância por quaisquer dos métodos *range-based* mencionados na Seção 2.2.2.

A idéia chave por detrás do CentroidM é manipular as informações recebidas de forma a adicionar uma terceira variável, qual seja, o tempo. A informação outrora recebida pelo Centroid não será alterada, ou seja, não haverá aumento do número de *bytes* da carga útil das mensagens enviadas pelos nodos âncora. Por outro lado, os dados contidos em tais mensagens serão armazenados durante o período de amostragem de forma a classificá-los temporalmente em diferentes períodos, sendo que para cada um destes períodos ou janelas temporais haverá um determinado peso. As informações recebidas na última janela de tempo, inserida no período de amostragem, possuirão uma importância diferenciada em relação ao que foi recebido na janela temporal do início do período.

Mais tarde, com o término do período de coleta de dados, as informações de identificação do nó âncora e suas coordenadas serão manipuladas novamente, primeiro para eleger os nodos relevantes ao processo e, após para identificar o grau de importância de suas coordenadas no cálculo final da estimativa de localização. As alterações a serem propostas visam impactar significativamente nos seguintes processos:

- seleção de nodos âncora envolvidos na etapa de estimativa das coordenadas;

³Seleção dos nodos âncora e a definição de suas importâncias no processo de localização, como será visto logo a seguir.

- quantização da importância⁴ de cada nó âncora de acordo com o número de mensagens recebidas.

O CentroidM executa o processo de seleção de âncoras em dois momentos distintos como pode ser verificado pelo pseudo-código apresentado no Algoritmo 1. A primeira seleção de nodos âncora executada pelo algoritmo CentroidM segue, em parte, o procedimento realizado pelo Centroid original. A diferença é que o CentroidM não baliza a escolha dos nodos âncora exclusivamente por um limiar inferior. Este fato é preponderante, por exemplo para ambientes ruidosos. De posse deste dado, o algoritmo constrói uma matriz que contém as combinações simples 3 a 3, de 5 nodos de referência, retornando então 10 grupos.

Uma vez que estes grupos estão definidos, o CentroidM realiza a segunda seleção, escolhendo dentre os grupos disponíveis apenas um para incluir no cálculo das estimativas. A seleção do grupo da trilateração é baseada no maior somatório, de cada grupo, relativo às métricas de conectividades ponderadas de cada âncora. A maior destas somas indicará a região (triângulo) que o nó móvel ou estático se encontra.

A posição final estimada levará em conta outra ponderação. As métricas de conectividade gerais são então abandonadas e o algoritmo utiliza a tabela de mensagens ponderadas. Os âncoras selecionados para a trilateração possuem um determinado peso calculado em relação à soma das mensagens ponderadas do grupo. Cada peso será aplicado a cada coordenada resultando na estimativa de localização. Todos estes aspectos contribuem para o aumento do tempo de processamento do algoritmo, no entanto mantém-se a simplicidade original dos cálculos.

O pseudo-código apresentado no Algoritmo 1 descreve o CentroidM. O algoritmo proposto lida com a informação passada para estimar a posição do nó desconhecido. Apesar do CentroidM utilizar muitos conceitos já mencionados na Seção 4.1, outros termos são utilizados pelo CentroidM e, portanto, são necessários serem definidos, como segue.

⁴Esta quantização é aplicada às coordenadas do nó âncora.

- w : janela de tempo inserida dentro do período de amostragem t
- σ_w : peso atribuído à janela temporal w
- M_w : número de mensagens recebidas de todos os âncoras na janela w
- $m_{(i,w)}$: número de mensagens recebidas do âncora A_i dentro de w
- $CMp_{(i,w)}$: Métrica de Conectividade Parcial para o âncora A_i na janela w
- μCMp_i : média ponderada dos valores de $CMp_{(i,w)}$ no tempo de amostragem t , de acordo com σ_w
- μrec_i : média ponderada das mensagens recebidas de A_i no tempo de amostragem t , de acordo com σ_w
- σm_i : peso calculado do âncora A_i considerando μrec_i
- (XM_{est}, YM_{est}) : localização estimada

Algorithm 1 Pseudo-código de alto nível para o processo de localização proposto. O código abaixo é executado uma vez ao término de um tempo de amostragem

O tempo de amostragem é dividido em janelas de tempo w iguais

for each nó âncora ouvido, calcular: **do**

Conectividade Métrica Simples CM_i

μCMp_i de $CMp_{(i,w)}$ na janela de tempo w

end for

selecionar os 5 maiores valores de CM_i

encontrar as combinações possíveis de tamanho 3 para os nodos âncora selecionados

for each combinação encontrada **do**

Somar todas as μCMp_i de cada A_i na combinação

end for

escolher a maior soma

for each A_i da combinação selecionada **do**

computar μrec_i considerando o tempo de amostragem

computar σm_i considerando μrec_i

end for

Estimar as coordenadas de acordo com σm_i e (XA_i, YA_i)

Num primeiro momento, o tempo de amostragem é particionado em janelas de tempo w iguais. Cada um destes segmentos temporais recebe um peso σ_w definido pelo usuário. Os valores de σ_w são, portanto, constantes até o momento de uma intervenção externa do mantenedor da rede, por exemplo. O peso σ_w é utilizado no processo de seleção dos nodos âncora e no cálculo das coordenadas a serem estimadas.

De maneira interligada ao particionamento do tempo de amostragem, o algoritmo irá considerar apenas o montante das mensagens recebidas dentro de cada janela de tempo para calcular as métricas de conectividade parciais $CM_{p(i,w)}$. Agindo desta forma, o processo permite inferir de forma mais precisa⁵ a importância de um nó âncora particular. Isto é executado através da análise de suas mensagens recebidas em relação ao total recebido de todos os nodos âncora, dentro de cada janela de tempo.

O exemplo apresentado nas Figuras 15(a) e 15(b) visa esclarecer de que forma o processo de ponderação das mensagens recebidas contribui para o cálculo da estimativa de posicionamento do nó. Foi observado através de resultados obtidos com simulações da rede de sensores que o balizamento da seleção dos nodos âncora apoiado apenas nos valores máximos de CM_i não era eficiente⁶. A Tabela 3 apresenta um exemplo de como as mensagens são classificadas pelo CentroidM e os valores calculados de CM_i e μCM_{p_i} .

De acordo com a Tabela 3, é possível verificar que apesar do valor de CM_5 ser superior a CM_8 , o mesmo não acontece para μCM_{p_5} e μCM_{p_8} . Isto é explicado pelo fato de o âncora A_8 recebeu uma quantidade muito maior de mensagens em relação a A_5 , indicando que o nó desconhecido está se aproximando de A_8 . O algoritmo CentroidM é capaz de atribuir uma importância maior à informação recebida ao final do tempo de amostragem. Os pesos σ_w estabelecidos para as janelas temporais w são arbitrários e constituem alguns dos parâmetros configuráveis do CentroidM.

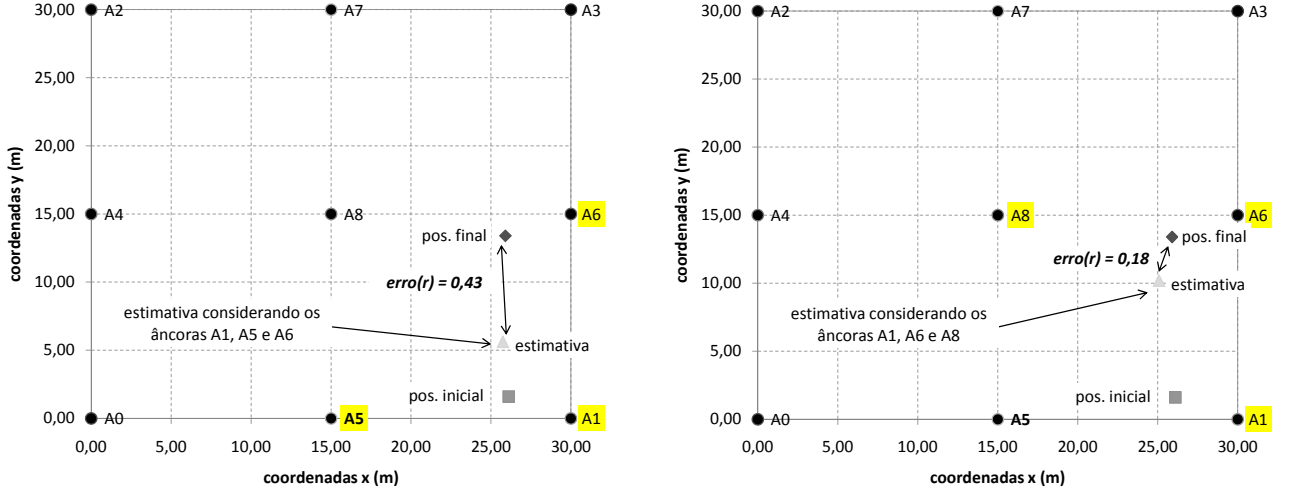
Conforme os nodos cujas posições são desconhecidas movimentam-se pela região delimitada, a informação por eles captada no início do período de amostragem t está defasada. Isto provocava a interpretação ineficaz das métricas de conectividade simples CM_i para a seleção dos nodos âncora a serem empregados no cálculo das estimativas.

Desta forma, verificou-se a importância de considerar, além da posição recebida dos nodos de referência, o tempo no qual tal informação é captada. O algoritmo CentroidM classifica e realiza uma ponderação acerca dos dados enviados pelos âncoras. Os gráficos das Figuras 15(a) e 15(b) apresentam uma região de $30m \times 30m$ onde um nó desconhecido movimenta-se. Este nó parte de uma posição inicial até alcançar uma posição final ao término do período de amostragem t , como indicado nos gráficos das Figuras 15(a) e 15(b). No primeiro caso, isto é, na simulação realizada para a Figura 15(a) apenas os valores de CM_i são considerados para a seleção dos nodos de referência, enquanto que a Figura 15(b) mostra o caso onde μCM_{p_i} define os nodos a serem utilizados no processo

⁵Ver exemplo envolvendo a Figura 15(a) e a Figura 15(b).

⁶O algoritmo Centroid utiliza apenas o valor de CM_i para a seleção dos nodos de referência, o que retorna resultados menos precisos em relação ao CentroidM, conforme poderá ser visto no Capítulo 7.

de cálculo das estimativas.



(a) seleção dos âncoras utilizando apenas CM_i .

(b) seleção dos âncoras utilizando μCMp_i .

Figura 15: Exemplo do impacto das métricas de conectividade ponderadas no cálculo da estimativa. Os gráficos das subfiguras (a) e (b) representam uma região de $900m^2$ com 9 nodos âncora, A0 a A8, e um nó cuja posição é desconhecida. Neste cenário o nó desconhecido movimenta-se de uma posição inicial até chegar a uma posição final, ao término do período de amostragem t , quando então o algoritmo de localização CentroidM é executado.

Tabela 3: Exemplo do impacto de CM_i e μCMp_i , relativo ao cenário apresentado nas Figuras 15(a) e 15(b), para a seleção dos nodos âncora utilizados no cálculo das coordenadas.

coordenadas (m)		A_i	$m_{(i,w)}$				CM_i	μCMp_i	
x	y		w_1	w_2	w_3	w_4			
0,00	0,00	0	1	0	1	1	0,09	0,37	
30,00	0,00	1	9	8	7	5	0,91	2,85	☺
0,00	30,00	2	0	1	0	0	0,03	0,06	
30,00	30,00	3	0	2	1	3	0,19	0,82	
0,00	15,00	4	0	0	1	0	0,03	0,14	
15,00	0,00	5	5	8	6	4	0,72	2,34	☹
30,00	15,00	6	5	5	7	8	0,78	3,02	☺
15,00	30,00	7	0	2	0	3	0,16	0,67	
15,00	15,00	8	1	5	5	9	0,63	2,72	☺
			\uparrow	\uparrow	\uparrow	\uparrow			
$\text{peso } \sigma_w$			1	2	4	6			

O primeiro passo para o processo de seleção dos nodos âncora recai sobre os valores calculados de métrica de conectividade simples CM_i . Sem qualquer processo de ponderação, estes valores conduzem a seleção de cinco âncoras cuja CM_i excedem CM_{thresh} ⁷. São selecionados 2 nodos âncora adicionais aos 3 necessários para o processo de trilateração. Isto é feito para englobar âncoras em potencial, não identificados na primeira

⁷Este valor é arbitrário e foi definido como 90% em (BULUSU; HEIDEMANN; ESTRIN, 2000).

etapa de seleção que utiliza apenas CM_i . O aumento do número de nodos de referência escolhidos, para além de 5, não impactou no cálculo das coordenadas finais e portanto foi descartado. Desta forma, é construída uma lista com as identificações dos nodos de referência e suas métricas de conectividade relativas. O nó desconhecido captura as informações destes âncoras e realiza uma combinação de cinco nodos em grupos de três, resultando em dez possibilidades distintas conforme a Equação 4.3.

$$C_5^3 = \frac{5!}{3!(5-3)!} \quad (4.3)$$

De posse destes dados, uma segunda lista é formada e consiste em armazenar a soma das conectividades métricas parciais ponderadas $\mu C M p_i$ para cada grupo de âncoras, de acordo com as possibilidades da combinação realizada. O processo de seleção é então finalizado e a maior soma $\mu C M p_i$ é escolhida. Um novo procedimento de ponderação toma início. Segundo a Equação 4.4, para cada nó âncora que compõe a combinação selecionada é realizado um cálculo acerca das mensagens recebidas. O número de mensagens recebidas através das janelas temporais é então analisada. Por fim, cada nó âncora recebe um peso σm_i de acordo com o total das mensagens recebidas pelo grupo selecionado.

$$\sigma m_i = \frac{\mu rec_i}{\sum (todos_selecionados) \mu rec_i} \quad (4.4)$$

A estimativa **final** da posição é calculada levando em consideração as coordenadas dos três âncoras e as suas importâncias relativas, de acordo com a Equação 4.5. Desta forma, cada nó desconhecido irá computar localmente sua localização.

$$\begin{aligned} X M_{est} &= \sum [(todos_selecionados) X A_i \times \sigma m_i] \\ Y M_{est} &= \sum [(todos_selecionados) Y A_i \times \sigma m_i] \end{aligned} \quad (4.5)$$

Distinguem-se desta forma dois momentos onde pesos são utilizados durante a execução do algoritmo. O primeiro deles diz respeito à definição por parte do usuário da importância de cada janela temporal em relação à outra. Este, na realidade, é um parâmetro configurável do algoritmo. É importante que as janelas temporais mais próximas do instante da execução do algoritmo recebam um peso maior, quando uma rede

móvel é considerada. Por outro lado, para uma rede dita estática, a importância do fator tempo no sentido de informar quando as coordenadas dos âncoras foram captadas diminui. Neste último caso não há alteração significativa da quantidade de mensagens recebidas de cada nó de referência atribuída ao movimento, pois a topologia da rede permanece constante.

Já a segunda ponderação de valores ocorre de acordo com as informações recebidas dos nodos âncora da rede. Os pesos σm_i utilizados no processo de cálculo final das coordenadas são computados de acordo com as mensagens recebidas de cada nó âncora, selecionado por um processo anterior. Desta forma, estes pesos não são diretamente definidos pelo usuário do algoritmo, mas sim um resultado da execução do algoritmo e, portanto, são modificados a cada nova rodada de estimativa.

O conceito de média móvel ponderada (FREUND; SIMON, 2000) foi aplicado ao algoritmo Centroid. Este tipo de média é utilizada com muita frequência em problemas financeiros, na análise de ações em bolsas de valores. Ela é importante quando as grandezas envolvidas não possuem a mesma importância em relação ao objetivo que está sob estudo. Desta maneira é preciso atribuir-lhes pesos e então calcular a média ponderada. A média ponderada é qualquer média que multiplica fatores para fornecer diferentes pesos para diferentes dados. De modo geral, a média ponderada \bar{x}_w de um conjunto de números x_1, x_2, \dots, x_n , cuja importância relativa é expressa numericamente por um conjunto correspondente de números w_1, w_2, \dots, w_n , é dada pela Equação 4.6.

$$\bar{x}_w = \frac{w_1x_1 + w_2x_2 + \dots + w_nx_n}{w_1 + w_2 + \dots + w_n} \quad (4.6)$$

Este conceito transportado ao CentroidM, portanto, atribui um peso mais elevado à quantidade de mensagens recebidas mais recentes e diminui a importância de valores mais antigos. Caso os pesos de maior valor sejam aplicados a dados mais recentes e menor peso aos dados mais antigos, o resultado da média móvel será mais suscetível às oscilações recentes no número de mensagens recebidas. Desta forma, são constatadas as mudanças de tendência que acompanham o desenvolvimento dos dados de uma série.

A média ponderada foi aplicada a dois momentos distintos do algoritmo. O processo de escolha dos nodos âncoras foi auxiliado pelo cálculo das conectividades métricas parciais $CMp_{(i,w)}$ e conseqüentemente pela média ponderada móvel μCMp_i destas conectividades. Já na etapa de cálculo das coordenadas, a média móvel ponderada foi empregada para ponderar o número total de mensagens recebidas e calcular o peso σm_i de cada um dos

nodos âncora envolvidos no processo de estimativa da localização do nó.

4.3 Resumo

Este capítulo apresentou em detalhes o algoritmo Centroid, analisando seus pontos positivos e negativos. Também foram apresentadas as razões para a escolha deste método de localização como base para a proposição de melhorias que incluíssem o aspecto da mobilidade no algoritmo. Foram enfatizadas modificações no método nas etapas de tratamento inicial dos dados recebidos, escolha dos nodos âncoras envolvidos no processo de localização e ponderação das coordenadas destes âncoras selecionados. Estas 3 modificações abordaram os pontos identificados como deficientes do algoritmo, principalmente no tocante à mobilidade.

5 *AMBIENTE DE DESENVOLVIMENTO E METODOLOGIA*

Este capítulo apresenta as ferramentas utilizadas durante o processo de implementação do algoritmo proposto. São introduzidos os aspectos principais do TinyOS (HILL et al., 2000), nesC (GAY et al., 2003), TOSSIM (LEVIS et al., 2003) e Avrora (TITZER; LEE; PALSBERG, 2005). Detalhes específicos do funcionamento interno dos simuladores estão além do escopo deste trabalho e podem ser verificados diretamente na literatura indicada. A primeira parte deste Capítulo versa acerca do ambiente utilizado para a validação do algoritmo CentroidM em um *software* para simulação de redes de sensores. Neste ambiente também que o algoritmo Centroid (BULUSU; HEIDEMANN; ESTRIN, 2000) original é reproduzido.

5.1 *Software*

O desenvolvimento de aplicações para redes de sensores pode ser realizado através do emprego de dispositivos físicos reais, tais como o nó sensor MicaZ (CROSSBOW, 2009b), diretamente sobre o cenário alvo. Este procedimento, apesar de eficiente sob o ponto de vista do comportamento real do sistema, muitas vezes requer tempo e equipamentos em uma quantidade que pode não estar disponível. Um simulador é capaz de transpor estas dificuldades, permitindo ao usuário o controle sobre o meio e a fácil reprodutibilidade dos testes, além de disponibilizar meios de monitorar de maneira mais detalhada as interações entre os nodos da rede.

O principal objetivo de uma simulação é disponibilizar ao pesquisador um acesso rápido ao comportamento, neste caso, de uma rede de sensores face ao cenário visionado e ao método proposto. Nesta etapa é avaliado o comportamento funcional da rede em um ambiente controlado, permitindo facilmente alterar as configurações e inspecionar

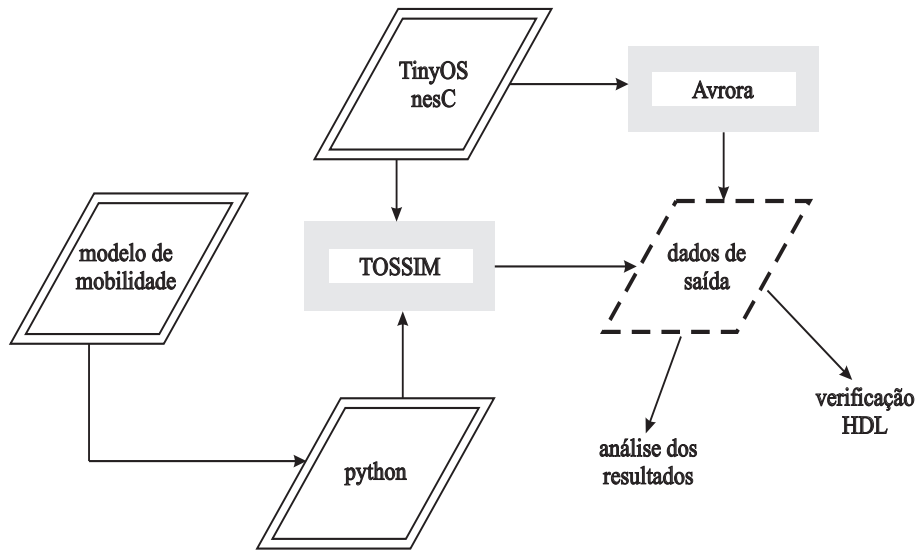


Figura 16: Fluxo utilizado na etapa de validação funcional.

interações da rede que possivelmente seriam muito rápidas, infrequentes ou demasiado complexas de serem observadas em um sistema real.

A Figura 16 apresenta o fluxo de trabalho empregado na primeira metade desta Tese. Destacam-se os programas TOSSIM (LEVIS et al., 2003) e Avrora (TITZER; LEE; PALSBERG, 2005). Cada qual desempenhou um papel específico na obtenção dos resultados. O TOSSIM foi o simulador utilizado na etapa de validação funcional tanto do Centroid original quanto do CentroidM. Os modelos de mobilidade são interpretados por um programa em Python, o qual também é responsável por configurar a simulação da rede a ser executada pelo TOSSIM. O TOSSIM por sua vez interpreta o código escrito em nesC e entrega os dados de saída para serem analisados. A estimativa precisa do tempo de processamento de cada uma das aplicações foi obtida com o Avrora. Convém salientar ainda a inserção do aspecto da mobilidade na simulação com o TOSSIM. Originalmente o TOSSIM não possui um mecanismo que permite simular os movimentos dos nodos e portanto esta etapa precisou ser adicionada ao simulador através de um mecanismo explicado na Seção 5.4.

5.2 TinyOS e nesC

TinyOS (HILL et al., 2000) é um sistema operacional em código aberto projetado para redes de sensores. Sua arquitetura é baseada em componentes. Um componente é convencionalmente considerado como sendo o bloco básico de construção de uma aplicação ¹ escrita em

¹Uma aplicação representa a ligação de todos os componentes envolvidos, como será visto mais adiante.

nesC (GAY et al., 2003). Esta característica torna o TinyOS um sistema modularizado e direcionado a dispositivos embarcados de tal maneira que o sistema operacional que é carregado em cada aplicação pode ser muito diferente dos outros. Adicionalmente, o TinyOS suporta apenas alocação estática de memória e seu modelo de programação baseado em componentes é codificado pela linguagem nesC (GAY et al., 2003), a qual será vista ainda nesta Seção.

Os componentes em TinyOS possuem 3 conceitos computacionais básicos: comandos, eventos e tarefas. Comandos e eventos são mecanismos para comunicação entre componentes enquanto que as tarefas são utilizadas para expressar concorrência entre os componentes. Um comando é tipicamente um requerimento para um componente realizar um serviço, onde um exemplo pode ser iniciar a leitura de um sensor, enquanto que um evento sinalizaria o término de tal serviço. Os eventos podem ser sinalizados de maneira assíncrona, por exemplo, devido a interrupções de *hardware* ou à chegada de uma mensagem. O comando retorna de maneira imediata quando acionado e o evento sinaliza a conclusão algum tempo depois. Ao invés de realizar uma computação de maneira imediata, comandos e eventos podem estabelecer uma tarefa, uma função executada pelo controlador de tempo do TinyOS em um instante futuro. As tarefas representam concorrência interna dentro de um componente e podem apenas acessar estados de informação dentro daquele componente.

A modularidade do TinyOS, por sua vez, permite que mais novidades sejam adicionadas a este sistema e que a implementação seja acelerada, ao mesmo tempo, proporcionando que códigos mais reduzidos possam ser utilizados, o que é uma grande vantagem sob o ponto de vista de um *hardware* com severas limitações de memória. A biblioteca de componentes do TinyOS inclui protocolos de rede, serviços distribuídos, *drivers* de sensores e ferramentas de aquisição de dados, sendo que todos estes componentes citados podem ser utilizados da maneira como estão ou podem ser refinados para uma aplicação adaptada às preferências do usuário. Os modelos de execução orientados por eventos do TinyOS possibilitam uma manipulação detalhada da potência, ao mesmo tempo que permitem flexibilidade de configuração necessária devido à natureza imprevisível da comunicação sem fio e interface como mundo físico.

O desenvolvedor que quiser compor uma aplicação para TinyOS deve escrever componentes em nesC e ligá-los a outros componentes que implementam os serviços requeridos. Alguns conceitos que são utilizados na programação para o TinyOS precisam ser aqui resumidos:

- aplicação: consiste em um ou mais componentes conectados;
- componente: é o bloco básico das aplicações em nesC. Um componente pode ser classificado como *módulo* ou *configuração* e, pode prover e usar *interfaces*;
- módulo: é um componente que implementa uma ou mais interfaces, ou seja, provê interfaces. A idéia é que o desenvolvedor possa construir uma aplicação como sendo um conjunto de módulos, conectando estes módulos por intermédio de uma configuração;
- configuração: é um componente que liga outros componentes, conectando interfaces utilizadas por componentes a interfaces providas por outros. Em linguagem nesC, uma aplicação é descrita como sendo uma configuração que especifica os componentes e como eles invocam uns aos outros;
- interface: é utilizada para prover uma definição abstrata da interação de dois componentes.

O nesC (GAY et al., 2003) é uma extensão da linguagem C para incorporar os conceitos e modelos de execução do TinyOS. Os programas são construídos pela montagem ou ligação de componentes, os quais possuem uma concorrência interna estabelecida por tarefas. Como mencionado anteriormente, as interfaces podem ser providas ou utilizadas pelos componentes, sendo que as interfaces providas objetivam representar a funcionalidade que o componente provê para seu usuário. Já as interfaces utilizadas representam a funcionalidade que o componente precisa para executar seu trabalho. Tais interfaces especificam um conjunto de funções que podem ser classificadas como comandos se implementadas pelo provedor da interface ou eventos no caso do usuário da interface. Desta forma uma única interface pode representar uma complexa interação entre componentes

5.3 TOSSIM

TOSSIM (LEVIS et al., 2003) é um simulador do sistema operacional TinyOS para redes de sensores sem fio e implementa um modelo de rede. Este simulador emula o comportamento do nó sensor MicaZ (CROSSBOW, 2009b), o qual será referenciado no restante desta Tese como *mote*. Este dispositivo sensor, ou mote, é fabricado pela empresa Crossbow (CROSSBOW, 2009a). O TOSSIM é capaz de simular aplicações, a execução de cada nó, a rede em si e suas interações. Seu propósito é permitir ao usuário a compilação do sistema em um computador e facilmente transportá-lo aos motes. Este simulador

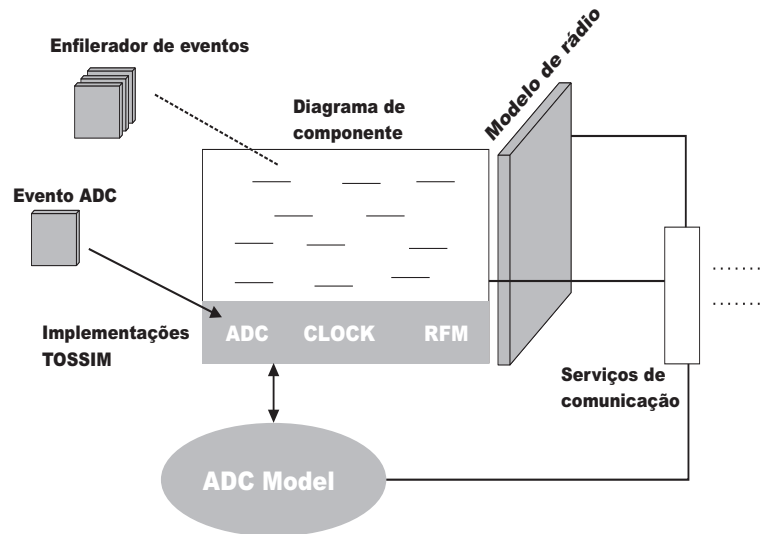


Figura 17: Esquema simplificado da arquitetura TOSSIM (adaptado de (LEVIS et al., 2003)).

baseia-se em quatro características principais desejáveis para um ambiente de simulação TinyOS:

- escalabilidade: o simulador deve ser capaz de manipular uma grande quantidade de nodos;
- abrangência: cobrir várias interações entre sistemas, de maneira a possibilitar a integração de outras propostas ao simulador;
- fidelidade: este ponto trata da necessidade do simulador em capturar o comportamento da rede em detalhes, revelando ao desenvolvedor aspectos que sequer foram previstos durante o projeto;
- conexão: faz-se alusão aqui ao objetivo de estabelecer uma conexão entre algoritmo e implementação, permitindo aos desenvolvedores testar e verificar o código que será executado no *hardware* real.

A arquitetura do TOSSIM está apresentada na Figura 17. Ela é composta de cinco partes: suporte para compilar componentes TinyOS para a infra-estrutura de simulação, um enfileirador de eventos, um pequeno número de componentes de *hardware* utilizados pelo TinyOS, mecanismos para modelar um conversor analógico-digital (ADC) e o rádio (RFM), e serviços de comunicação para possibilitar que programas externos possam interagir com a simulação.

O TOSSIM traduz as interrupções de *hardware* em eventos discretos de simulação. O enfileirador de eventos entrega as interrupções que direcionam a execução da aplicação

TinyOS. A abstração da rede pelo TOSSIM é dada de tal forma que a rede é composta por um diagrama no qual cada vértice é um nó e cada lado possui uma probabilidade de erro para os bits transmitidos e recebidos. Cada nó possui um estado particular representando o que ele ouve a partir do canal de rádio. Este tipo de abstração permite o teste sob perfeitas condições de transmissão onde a taxa de erro nos bits trocados é zero ou especificar erros de transmissão ao nível de bits. Isto possibilita ao TOSSIM identificar diversas causas de perda de pacotes e ruído em uma rede, incluindo símbolos de *start* perdidos ou corrupção de dados, muito embora o simulador não modele o ruído.

Cada evento de simulação está associado a um nó sensor específico da rede. Logo após executar um evento de simulação, o enfileirador executa as tarefas sob a fila de tarefas TinyOS do nó. Este processo do enfileirador TinyOS é realizado segundo uma FIFO e de maneira *run-to-completion*.

O TOSSIM provê um alto grau de precisão através da utilização de modelos de baixo nível de componentes de um nó sensor, além de executar um código que pode ser repassado ao *hardware*. Especificamente, este simulador compila o código fonte em nesC juntamente com as bibliotecas do TinyOS e, gera um arquivo binário, substituindo os módulos em *software* que se comunicam com o *hardware* por bibliotecas de emulação. Estas bibliotecas incluem temporizadores ou *timers*, canais de comunicação, sensores e o dispositivo de rádio.

5.4 Mobilidade

O modelo de rádio do TOSSIM consiste na probabilidade da ocorrência de erros durante a transmissão de bits em um processo de comunicação. O simulador considera uma taxa de erro relacionada com a etapa de transmissão e recepção de dados entre um nó e outro, sendo distinta para o caminho inverso. Este ponto em particular prevê a característica desejável (ALI et al., 2006) de assimetria dos canais de comunicação.

Um par de nodos participantes de uma rede e que são capazes de estabelecer troca de mensagens possui valores de taxa de erro relacionados com a sua comunicação. Para o TOSSIM os eventos de envio de dados se propagam ao canal de entrada simulado para cada nó conectado à rede. Cada nó sensor possui sua própria visão local do canal de rede. Este simulador gera gráficos de perdas a partir de topologias físicas, baseado em dados empíricos coletados de redes reais.

A ferramenta gera taxas de perdas de pacotes para cada par de nodos de acordo

com a distância de separação entre eles. Tais taxas são construídas pela amostragem de distribuições Gaussianas da probabilidade de perda de pacotes, chamados de modelos de perdas. Os modelos de perdas, por sua vez, são baseados nos dados empíricos acima mencionados. O TOSSIM traduz as taxas de perdas de pacotes em taxas independentes de erros de bit durante a comunicação.

O modelo de rádio do TOSSIM é dito externo ao simulador, o que indica o aspecto de modularização desta ferramenta. Na prática esta característica possibilita ao simulador receber melhoramentos ou adendos de maneira mais rápida não apenas de seus idealizadores, mas sim da comunidade científica, como por exemplo em (SHNAYDER et al., 2004; LEE; CERPA; LEVIS, 2007; PERLA et al., 2008).

O objetivo do estudo proposto pelos autores em (LEE; CERPA; LEVIS, 2007) foi modelar o ruído do ambiente para simular o sucesso na entrega de dados em um canal de comunicação. Lee e outros (LEE; CERPA; LEVIS, 2007) identificaram que as taxas de entrega de pacotes em uma rede tinham uma forte variação sobre a distância envolvida entre os dispositivos, bem como o ambiente envolvido. Ao identificarem que o TOSSIM simulava perdas ao invés de suas causas, verificaram que o simulador não poderia modelar de maneira precisa diferentes ambientes, transmissões concorrentes de dados ou variações no tamanho dos pacotes.

Por isto propuseram uma maneira de simular de forma precisa a entrega de pacotes. Tanto o Centroid quanto o CentroidM são algoritmos puramente *range-free*, baseados no aspecto de conectividade com os outros nodos da rede. Desta forma, este tipo de abordagem proposta em (LEE; CERPA; LEVIS, 2007) para a modelagem do ruído ambiental encaixa-se de maneira satisfatória para o escopo desta tese. Atualmente esta proposta de modelamento do ruído está integrada à distribuição do TOSSIM juntamente com o TinyOS (COMMUNITY, 2009).

Para medir o ruído² ambiental os autores em (LEE; CERPA; LEVIS, 2007) escreveram uma aplicação TinyOS que amostrou, a uma taxa de $1kHz$, a energia RF através da leitura do registrador de intensidade de sinal (RSSI - *Received Signal Strength Indication*) do dispositivo de rádio CC2420 (CHIPCON, 2009) presente no mote MicaZ (CROSSBOW, 2009b) da empresa Crossbow (CROSSBOW, 2009a). Este procedimento foi repetido durante aproximadamente 197 segundos e retornou $3 * 2^{16}$ amostras. Esta quantidade de dados constitui um arquivo chamado de rastreamento de ruído ou *noise trace*. Este *noise trace* alimenta a abordagem estatística selecionada pelos autores para modelar o ruído e retorna

²Entenda-se por ruído a energia RF gerada por fontes que não estão sob o controle do projetista.



Figura 18: Biblioteca Meyer na Universidade de Stanford, representando um ambiente com muitas fontes de ruído.

a simulação de entrega de pacotes ao TOSSIM.

A modelagem da rede pelo sucesso na entrega de pacotes proposta em (LEE; CERPA; LEVIS, 2007) não está confinada simplesmente a uma taxa de perdas. O modelo depende de um arquivo de rastreamento que é específico de uma determinada região. Desta forma, a simulação da rede adapta-se ao cenário visionado, possuindo relação direta com a comunicação entre nodos e entrega de pacotes. (LEE; CERPA; LEVIS, 2007) et.al mostraram um aumento da fidelidade da simulação de pacotes, em relação às abordagens existentes, por um fator igual a 2 considerando canais de comunicação sem fontes de ruído, 1,5 para muitas fontes de ruído e 5 no caso intermediário.

Dois ambientes distintos para os quais existe um *noise trace* em particular representando cada um deles foram considerados neste trabalho. Ambos *noise traces* empregados em (LEE; CERPA; LEVIS, 2007) foram disponibilizados por Lee (LEE; CERPA; LEVIS, 2007) ao autor desta Tese. Estes ambientes representam dois cenários distintos. Um destes cenários trata de um local com muitas fontes de ruído: a Biblioteca Meyer, Figura 18 na Universidade de Stanford, nos Estados Unidos. O arquivo de rastreamento construído para a Biblioteca Meyer considerou a interferência provocada por transmissões de dados no padrão IEEE 802.11. Nesta Tese, o ambiente livre de fontes severas de ruído foi representado pela região delimitada por um lago artificial, o Lago Lagunita mostrado na Figura 19, também localizado na Universidade de Stanford.

Além de aspectos relacionados à transmissão do sinal, um algoritmo de localização que procure estabelecer uma relação com o quesito mobilidade deve simular um movimento realístico dos nodos da rede (CAMP; BOLENG; DAVIES, 2002). Existem diversos tipos de modelos de mobilidade disponíveis na literatura, sendo que cada um pode ser direcionado para uma aplicação específica. Logo a seguir serão apresentados 2 destes modelos que são



Figura 19: Lago artificial Lagunita na Universidade de Stanford, representando um ambiente com pouco ruído.

amplamente utilizados na avaliação de desempenho de redes de sensores.

5.4.1 Mobilidade de Percurso Aleatório

O *Random Walk Mobility Model* é um modelo de mobilidade bastante simples que é baseado em direções e velocidades aleatórias. Este modelo foi inicialmente matematicamente descrito por Albert Einstein em 1926 e visa imitar o comportamento de entidades na natureza que se movam de maneira extremamente imprevisível.

Considerando um ponto de partida aleatório em um plano 2D, um nó móvel se movimenta a partir de sua posição atual para uma nova pela escolha aleatória da direção e da velocidade de viagem. No entanto, a velocidade é escolhida de acordo com limites estabelecidos [*velocidade mínima*, *velocidade máxima*], bem como a direção $[0, 2\pi]$. Cada movimento neste modelo de mobilidade ocorre ou em um intervalo de tempo t ou em uma distância constante viajada d , de forma que ao final de uma destas duas situações, uma nova direção e velocidade sejam calculadas. No *Random Walk Mobility Model* o nó pode também mudar a direção depois de viajar uma distância especificada ao invés de um tempo especificado como mostrado na Figura 20.

5.4.2 Mobilidade de Percurso Aleatório com Pausas

O *Random Waypoint Mobility Model* inclui pausas entre mudanças de direção e velocidade do nó móvel. O nó permanece parado em uma determinada posição por um período de tempo (tempo de pausa) até que ao cabo deste período o nó escolhe um destino aleatório dentro da área de simulação e uma velocidade, a qual está compreendida entre [*velocidade mínima*, *velocidade máxima*]. Com isto o nó viaja até seu destino escolhido

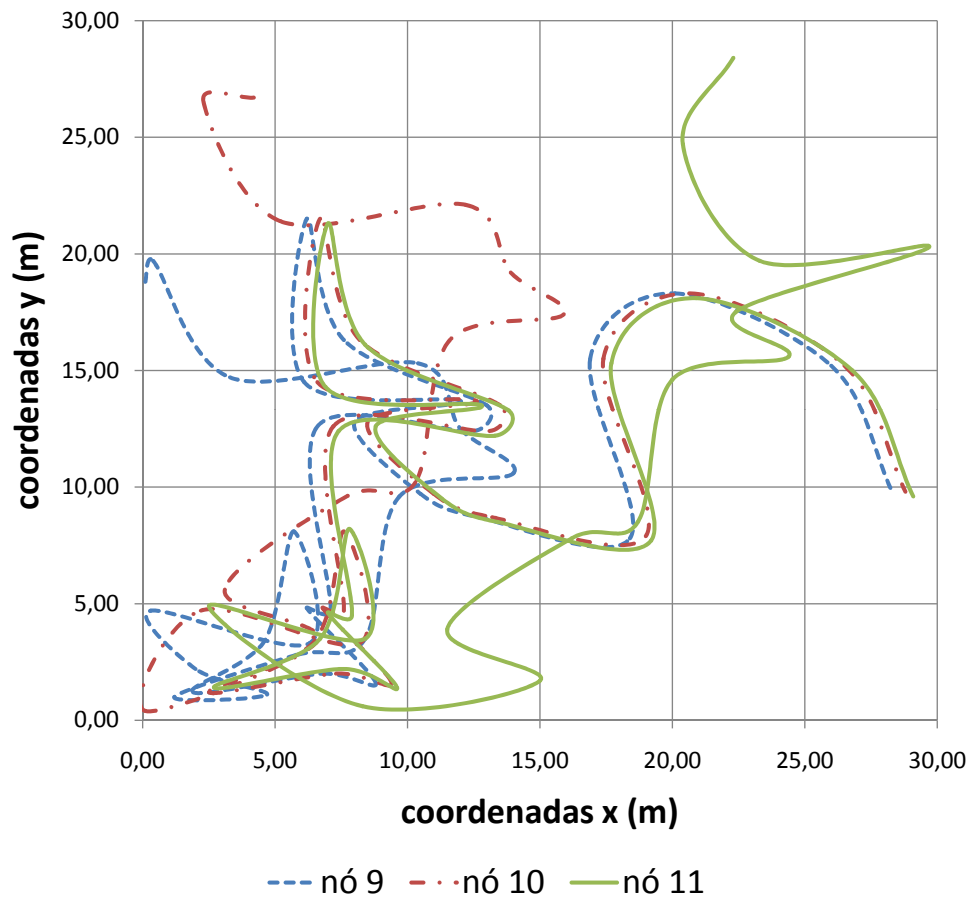


Figura 20: Exemplo de movimento de nodos utilizando o *Random Walk Mobility Model*.

a esta velocidade, ficando parado novamente pelo tempo de pausa antes de reiniciar o processo.

Os nodos na Figura 21 estão viajando dentro de uma área de $30m \times 30m$. O percurso desenvolvido nos 120 segundos da simulação inclui diversos momentos de pausa. Pode-se notar que os caminhos percorridos por estes nodos são menores do que aqueles executados pelos nodos da Figura 20.

Os modelos utilizados neste trabalho são uma versão estendida em relação aos apresentados em (CAMP; BOLENG; DAVIES, 2002). Estas versões estendidas aceitam a configuração de nodos fixos, os quais representam os nodos de referência da topologia. Além disto, a saída produzida pela execução do algoritmo que gera os modelos de mobilidade foi adaptada para a integração com o TOSSIM através de um programa escrito em Python. Esta rotina de comandos, ou *script*, alimenta o TOSSIM com as informações de posição de todos os nodos da rede ao longo do período de simulação de forma a estabelecer a trajetória do movimento. O número de arquivos gerados pode ser configurado de acordo com o intervalo de tempo desejado entre duas posições dos nodos e o tempo total de simu-

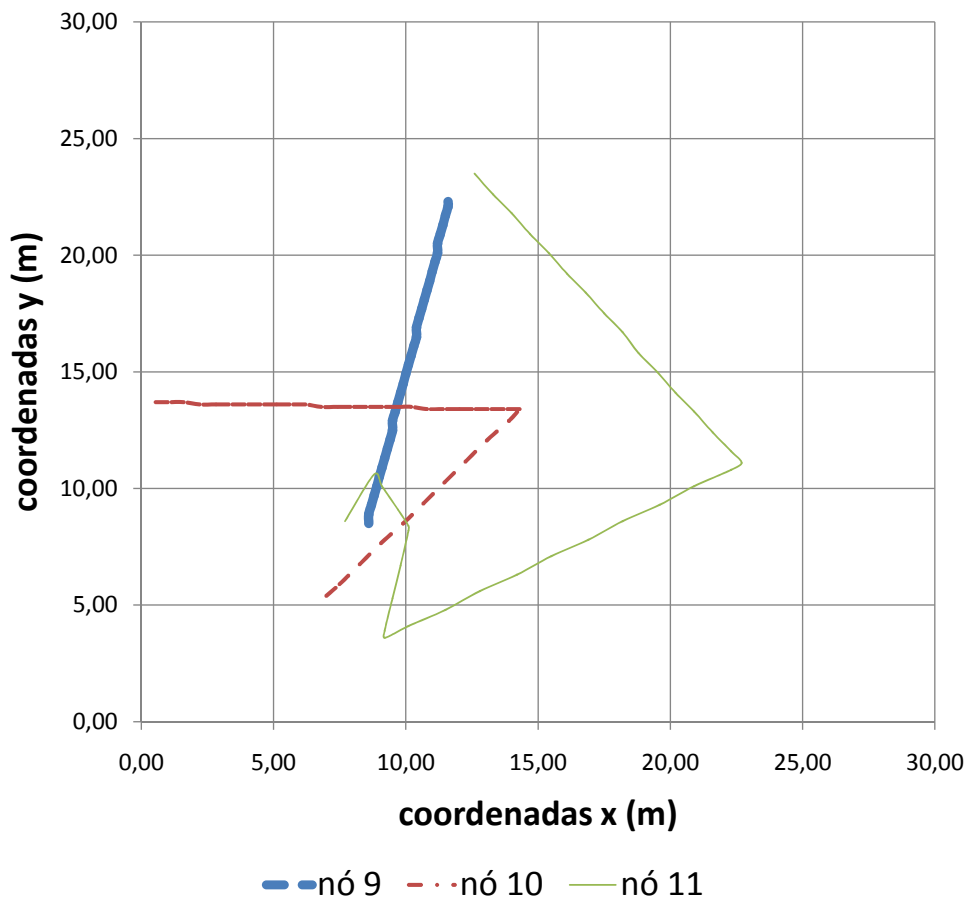


Figura 21: Exemplo de movimento de nós utilizando o *Random Waypoint Mobility Model*.

lação. Por exemplo, 10 segundos de simulação podem retornar 100 arquivos descrevendo o percurso desenvolvido pelo nó, caso o intervalo entre as posições seja definido em $0,1ms$.

5.5 Avrora

O simulador Avrora (TITZER; LEE; PALSBERG, 2005) possui como principal vantagem a precisão temporal sobre a aplicação que está sendo efetuada dentro do processador do *mote*³. O Avrora é capaz de simular uma rede de motes utilizando exatamente o mesmo código executado no processador. A capacidade computacional do processador simulado é estabelecida para no máximo 25MHz, o que é suficiente para a frequência de $7.3728MHz$ do microcontrolador ATMega128L. O número de *motes* suportados em uma topologia pode ultrapassar 10.000 unidades. Isto depende do poder de processamento do computador onde o simulador está sendo executado.

Cada nó dentro do Avrora é tratado como uma instancia de execução, ou *thread*. Os

³Definido na Seção 5.3.

motes executam as suas próprias *thread* e são vistos como um caixa preta geradora de eventos. Todos os motes da rede executam os seus códigos de maneira concorrente com os outros nodos da rede. O enfileirador de tarefas é o responsável por controlar o tempo das comunicações, quando um nó transmite e quando o dado é recebido. Este *software* utiliza a idéia do enfileiramento de tarefas descrita em (LAMPART, 1978). Em um sistema distribuído, um algoritmo é utilizado para sincronizar os relógios dos motes, os quais são utilizados para ordenar os eventos da rede. Devido à esta sua característica de precisão na contagem de ciclos de relógio, o Avrora foi utilizado nesta Tese no processo de aquisição do período necessário para a execução dos algoritmos Centroid e CentroidM.

5.6 Resumo

Neste Capítulo foi detalhada a metodologia utilizada para a reprodução do algoritmo de localização Centroid e desenvolvimento do CentroidM. Foram apresentadas as ferramentas para a validação funcional do método proposto no Capítulo 4. Para esta etapa foram mostrados os programas empregados e a solução interna para considerar a mobilidade no TOSSIM. O modelo de mobilidade foi adaptado à ferramenta de simulação de redes de sensores através da geração de diversos arquivos contendo a topologia da rede.

6 *CIRCUITO DEDICADO DO ALGORITMO PROPOSTO*

É abordada neste Capítulo a metodologia de trabalho que leva em conta o fluxo digital de projeto para um circuito de aplicação específica. Além dos aspectos diretamente relacionados com o algoritmo proposto nesta Tese, são introduzidos comentários relevantes acerca do fluxo de projeto de circuitos integrados utilizado. É importante salientar que existem inúmeras variações decorrentes de diferentes práticas adotadas e objetivos em um projeto, os quais resultam por sua vez na execução de etapas distintas das que serão aqui apresentadas nos fluxos. As etapas adotadas visam explanar acima de tudo acerca do processo envolvido na implementação do CentroidM. A arquitetura do algoritmo CentroidM, previamente definido na Seção 4.2.2, a ser realizado em *hardware* também é evidenciada neste Capítulo.

O objetivo inicialmente descrito nesta Tese é prover um sistema de localização dedicado baseado na conectividade dos nodos e na mobilidade da rede. A implementação almejada não constitui por si só o *hardware* que está presente em um nó sensor como por exemplo, blocos de transmissão e recepção ou unidade de processamento geral. O subsistema de localização trabalhará em conjunto com o restante dos outros dispositivos do nó sensor de maneira a constituir um *hard IP*¹ ou *soft IP*² a ser integrado como uma característica de mobilidade do nó, como mostrado na Figura 22.

O presente Capítulo enfoca a característica de mobilidade do algoritmo, com aumento da precisão dos resultados, atrelada à redução do consumo de potência. O processo de simulação do CentroidM em um simulador de redes de sensores foi descrito na Seção 5.1. A seguir será abordada a metodologia empregada para o projeto do circuito que implemente este subsistema de localização em *hardware*. O procedimento completo para o projeto de um circuito integrado envolve diversas etapas e foi desenvolvido dentro do

¹Trata-se de um IP que já passou pela etapa de síntese física. Normalmente este tipo de IP possui uma descrição GDSII definida.

²Pode ser referenciado como um IP que possui uma descrição VHDL sintetizável.

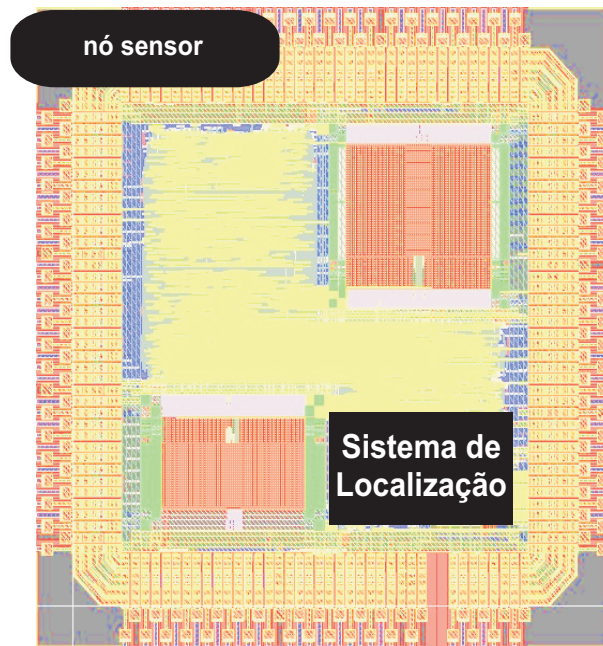


Figura 22: Nó sensor evidenciando o subsistema de localização.

ambiente Cadence (CADENCE, 2009) e utilizando tecnologia $0,13\mu m$. Estas tarefas estão relacionadas com a especificação do circuito, descrição de sua funcionalidade, projeto em linguagem de *hardware*, verificação funcional, mapeamento para a tecnologia alvo, análise de desempenho temporal e consumo.

6.1 Fluxo Digital do Circuito Integrado

Para percorrer as etapas do projeto do circuito integrado que implementa o algoritmo de localização CentroidM utilizou-se o fluxo digital apresentado nas Figura 23 e Figura 24. A Figura 23 apresenta, além das etapas que constituem a implementação lógica do circuito, dois procedimentos precedentes, quais sejam especificação e projeto HDL.

A primeira fase do fluxo tem a ver com a especificação do mesmo. Este primeiro procedimento é a base para o desenvolvimento de todo o restante do projeto. Em um fluxo ASIC, o documento de especificação balizará as decisões dos Engenheiros durante a vigência do projeto. Por exemplo, durante a etapa de verificação funcional, é o documento de especificação que servirá de base para a descoberta de erros no modelo de referência utilizado para a execução dos testes de caso.

Além de questões relacionadas diretamente com a funcionalidade do circuito integrado, o documento de especificação também engloba custos financeiros traduzidos: na restrição de área em silício disponível para a fabricação, tipo de encapsulamento, quantidade de

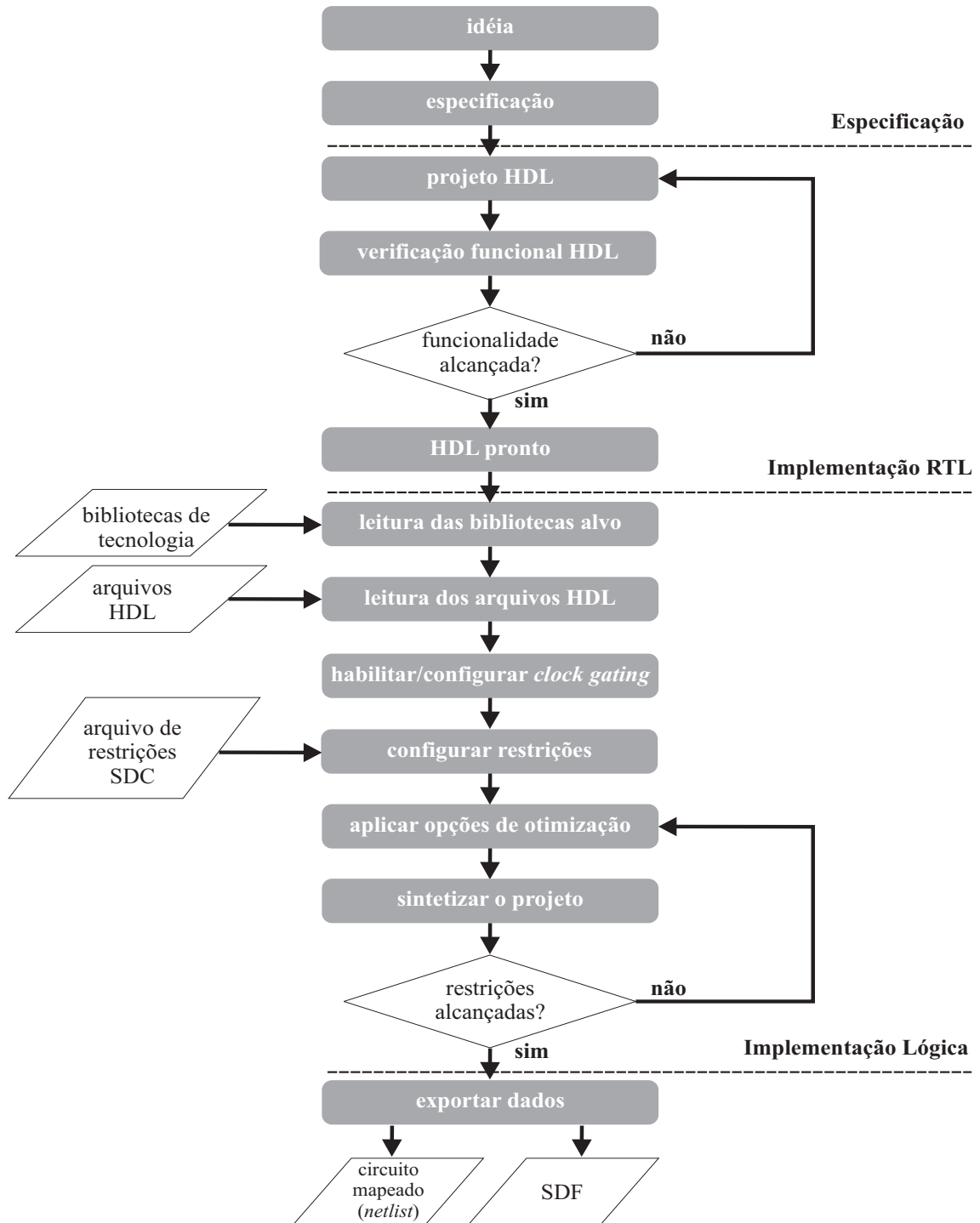


Figura 23: Fluxo digital para o projeto do CentroidM: etapa pré-leiaute.

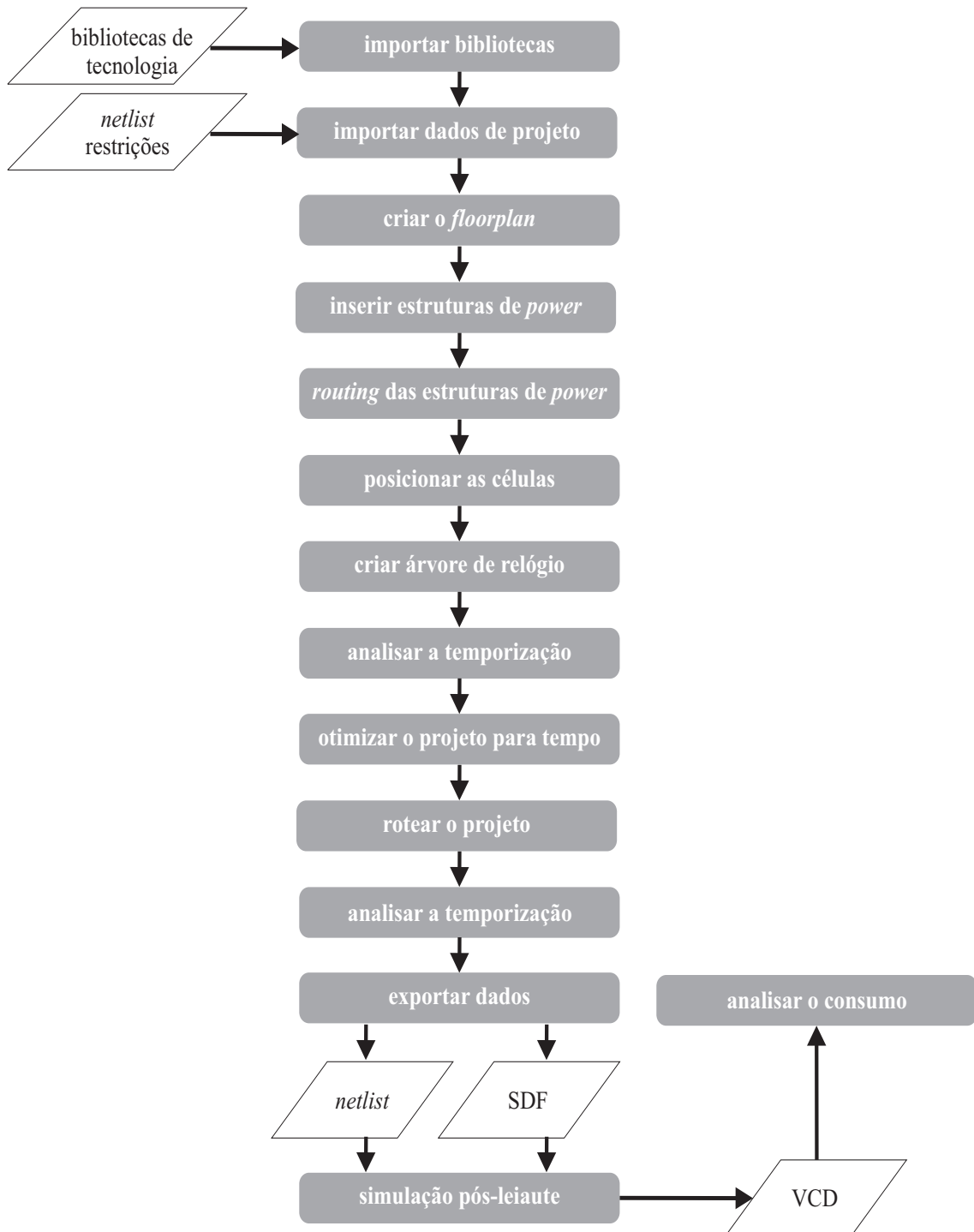


Figura 24: Fluxo de síntese física empregado para o CentroidM.

testes executados durante a fabricação ou mesmos aspectos pós-fabricação como comunicação do *chip* com outros dispositivos em uma placa de circuito impresso e a preparação da verificação real sob as condições de operação outrora determinadas. Enfim, o documento de especificação, embora possa ser negligenciado em detrimento de outras tarefas com o intuito de economizar tempo, representa uma fonte de consulta quando surgem questões durante o projeto sendo, portanto, uma das etapas de maior importância em um fluxo ASIC.

Logo a seguir à especificação, o processo descrito na Figura 23 tem continuidade com a descrição em linguagem de *hardware* do CentroidM. O resultado desta etapa é um código em VHDL, o qual é posteriormente submetido à verificação funcional para assegurar que as coordenadas estimadas pelo nó estejam sendo calculadas de maneira correta.

6.1.1 Síntese Lógica do CentroidM

Cumpridas as etapas de especificação, escrita do algoritmo em VHDL e verificação funcional, a descrição em *hardware* do CentroidM está pronta para ser sintetizada, isto é, o algoritmo será mapeado para portas lógicas que pertencem à biblioteca de tecnologia alvo. Com isto, faz-se necessário informar ao RTL Compiler (CADENCE, 2009)³ quais são os arquivos que contêm as informações tecnológicas. Para este projeto utilizou-se o processo CMOS 0,13 μ m disponível em (FARADAY, 2009). Cabe ressaltar no fluxo da Figura 23 a etapa de configuração para controlar o sinal de relógio⁴ nos registradores ou *clock gating*.

Mesmo que um dado seja carregado para o registrador com uma frequência muito inferior à de operação para a qual o circuito foi projetado, o sinal de relógio mantém o processo de comutação. Além disto, o sinal de relógio do circuito frequentemente alimenta uma carga capacitiva muito grande, o que faz deste tipo de sinal uma grande fonte de dissipação dinâmica de potência. O controle do sinal de relógio que é comum a alguns registradores, reduz comutações desnecessárias de partes do circuito. Para que a ferramenta de síntese lógica seja habilitada a construir blocos de controle para desligar o relógio em determinados registradores da maneira mais otimizada possível é necessário inserir esta opção antes de quaisquer comandos de síntese. Além disto, caso hajam células específicas para este propósito presentes na biblioteca é necessário liberar a sua utilização. Este tipo de célula é otimizada pelo distribuidor da tecnologia para realizar função de controlar o

³O RTL Compiler é a ferramenta padrão de síntese lógica do ambiente Cadence.

⁴Nesta Tese optou-se por utilizar a palavra **relógio** para referenciar ao sinal que determina a frequência de operação do circuito, muito embora o termo em inglês, *clock*, seja bastante difundido.

sinal de relógio e, portanto, são preferíveis à blocos construídos pelo RTL Compiler para tal fim.

Depois de obtida a síntese com sucesso para o CentroidM, o *netlist* resultante precisa ser verificado para violações de temporização, ou *timing*. Nesta fase do projeto apenas violações de *setup* nos registradores são analisadas pois o circuito ainda não possui informações acerca da árvore de relógio. O projeto é sintetizado com ênfase para maximizar o tempo de *setup*. Esta verificação de *timing* é pré-requisito para retornar à simulação funcional de posse dos atrasos estimados para os elementos do circuito. O arquivo denominado SDF ou *Standard Delay Format* contém a informação de *timing* calculada para cada célula do projeto e as interconexões. O produto da etapa de síntese lógica deve ser um *netlist* Verilog mapeado, livre de violações de *timing* e com sua funcionalidade assegurada pela simulação lógica.

6.1.2 Síntese Física do CentroidM

O procedimento inicial com a ferramenta de síntese física, evidenciado pelo fluxo da Figura 24, está ligado à disponibilização das informações tecnológicas. Estas informações são constituídas pelas bibliotecas de temporização e bibliotecas físicas. As bibliotecas de temporização ou *timing libraries* são responsáveis por armazenar o comportamento de cada uma das células que compõem a biblioteca em termos temporais. As restrições de temporização herdadas da etapa de síntese lógica e o circuito mapeado para a tecnologia alvo, além de outras configurações específicas de projeto também compõem o ajuste inicial.

O processo de planejamento da localização e forma geométrica de blocos ou módulos do circuito de topo é chamado de *floorplanning*. A etapa de *floorplanning* pode ser diferente para cada tipo de projeto, isto é, pode conter ou não etapas de posicionamento de células de PAD, distribuição da localização de pinos para comunicação inter-blocos ou com os PADs ou planejamento das linhas de alimentação do circuito. De maneira integrada ao *floorplanning* é executado um procedimento preliminar de posicionamento das células e ligação (*routing*) das conexões. Desta forma é possível antever pontos de congestionamento e tomar as medidas necessárias. Em caso mais extremos uma destas medidas é a re-execução do comando de *floorplanning* de forma a alterar a área inicialmente disponível para a alocação das estruturas.

Um projeto complexo exige um estudo prévio das interconexões dos módulos internos. Linhas de metal muito extensas irão impactar diretamente no desempenho de temporização do circuito. Por este motivo as etapas de *floorplanning*, posicionamento e roteamento

prévio são realizadas de maneira interativa. Isto enfatiza a importância do processo de *floorplanning*. Por outro lado, existem projetos nos quais o esforço nesta etapa pode ser minimizado dado a inexistência de restrições severas referentes a temporização da aplicação. Além disto, uma etapa de *floorplanning* pode inclusive ser realizada durante a fase de escrita do código em VHDL, por um projetista digital. Neste caso, o objetivo não será estabelecer o posicionamento final de todas as estruturas do *chip*, mas sim procurar antecipar a viabilidade do projeto dadas as especificações. É um procedimento que evita retornos no fluxo de projeto e economiza tempo de engenharia.

As mesmas verificações de temporização são executadas durante a síntese física, no entanto inserindo a preocupação com a verificação de *hold* e não mais considerando a árvore de relógio como sendo ideal. A quantidade de informações disponíveis também é maior para este tipo de análise visto que o local físico das células e tamanho das interconexões já estão definidos. O projeto pós-leiaute contém a informação da árvore de relógio. Nesta fase o sinal é propagado através de toda a rede de relógio para prover atrasos e transições reais. Antes de proceder à geração do SDF pós-leiaute é importante que as capacitâncias parasitas sejam extraídas para que o First Encounter possa calcular os atrasos dos elementos do circuito com maior precisão. Desta forma procede-se à última simulação funcional com o SDF anotado a partir da síntese física.

6.2 Descrição das Etapas de Verificação e Avaliação dos Algoritmos Centroid e CentroidM

A programação desenvolvida para reproduzir o comportamento do algoritmo foi avaliada funcionalmente e em termos de desempenho primeiramente considerando a execução dentro do processador comercial disponível nos notes. Além desta avaliação de desempenho, esta primeira fase foi importante para a validação comportamental das modificações propostas ao algoritmo trabalhado. Os dados da simulação com o TOSSIM são confrontados com os resultados de simulação obtidos através da implementação em linguagem de *hardware* da abordagem. A Figura 25 apresenta um diagrama que mostra as entradas e saídas utilizadas para a validação funcional.

A verificação funcional em nível HDL do CentroidM baseou-se na aplicação escrita em nesC e nos resultados de simulação entregues pelo simulador de redes TOSSIM. Os dados resultantes da simulação com o TOSSIM foram diretamente lidos pelo *testbench* VHDL sem a necessidade de uma interação adicional para mudança de sintaxe. Uma vez

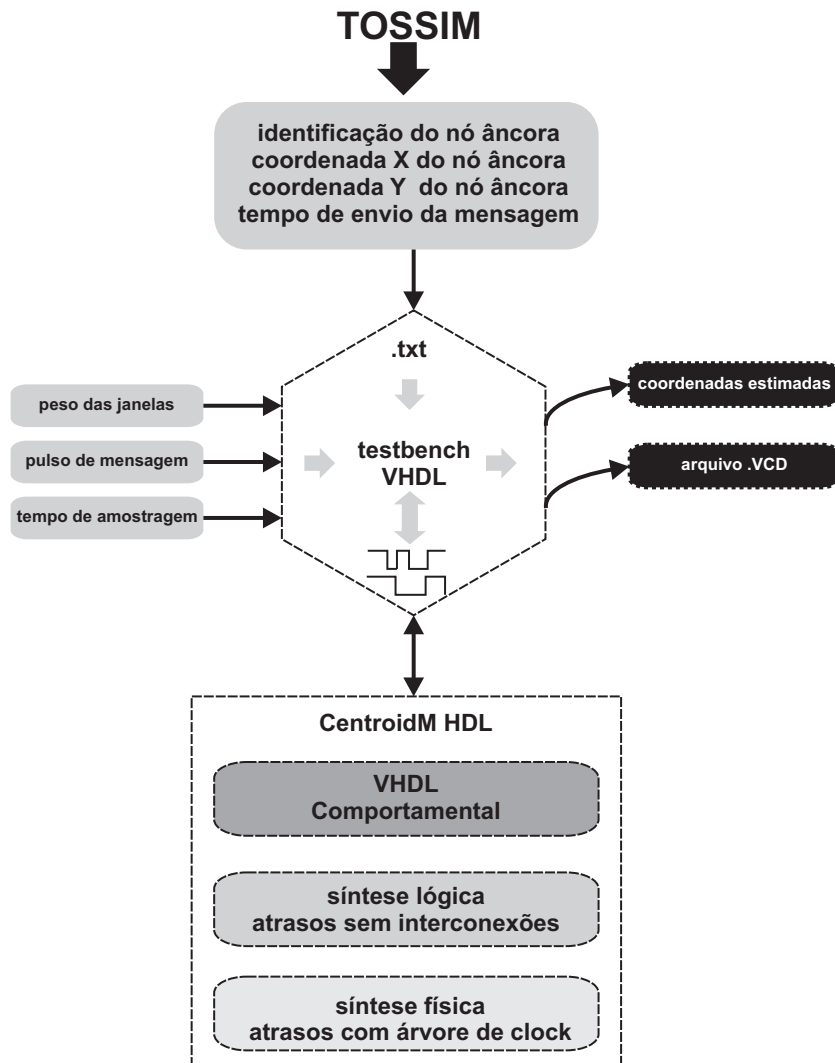


Figura 25: Esquema de verificação do circuito dedicado.

carregados pelo *testbench*⁵ os estímulos binários resultantes foram injetados nas entradas do circuito seguindo exatamente a mesma linha temporal do TOSSIM. Esta pequena automatização aumentou a confiabilidade do processo, tornando-o menos passível de erros humanos, ao mesmo tempo que possibilitou a extração dos resultados em um único arquivo para posterior análise. Isto possibilitou manter os mesmos vetores de teste empregados durante a simulação da rede no TOSSIM em todas as etapas de validação funcional do circuito.

Não foram desenvolvidas nesta Tese as etapas de elaboração de um modelo e a escrita de testes para inspecionar resultados intermediários do circuito. O procedimento de correção do código VHDL, quando necessário, baseou-se na análise visual dos dados de saída. Embora este procedimento seja inviável para circuitos maiores, os quais exigem

⁵Um *testbench* é uma entidade VHDL que instancia o circuito alvo e aplica os vetores que servirão de estímulo às suas entradas.

que verificações intermediárias sejam automatizadas, esta análise mostrou-se suficiente e eficaz para obter, ao cabo da verificação, um arquivo texto que contivesse as posições estimadas de acordo com o que foi calculado pela aplicação em nesC. Cabe ressaltar que em um processo de verificação executado numa empresa de projeto de circuitos integrados ou *design house*, prevê-se que a equipe de *front-end* digital divida-se em profissionais responsáveis pela codificação em VHDL do circuito, pelo modelo base e pela verificação de fato do código. Esta prática busca minimizar que os engenheiros que verificam sejam os mesmos que projetam e que elaboram o modelo do circuito. Todos estes profissionais devem estar a par do conteúdo do documento de especificação para as suas atividades.

O processo de verificação funcional HDL do algoritmo CentroidM foi executado em três momentos distintos do fluxo digital. Cada uma destas análises teve um propósito distinto. A primeira verificação objetivou analisar o comportamento do circuito durante o projeto HDL e garantir o seu funcionamento em condições ideais, ou seja, sem a inserção de quaisquer atrasos, apenas baseado na descrição VHDL comportamental interpretada pelo simulador. As duas simulações que foram executadas logo em seguida contêm dados de atraso do circuito alimentados pelas ferramentas de síntese lógica (RTL Compiler (CADENCE, 2009)) e síntese física (First Encounter (CADENCE, 2009)), de acordo com as portas lógicas mapeadas a partir da tecnologia alvo. A verificação com o *netlist* da síntese lógica levou em consideração apenas atrasos relacionados com o tempo de *setup* dos registradores, visto que não havia sido construída nenhuma árvore de relógio até esta etapa. A última simulação foi realizada com os atrasos das interconexões extraídos pelo First Encounter e o sinal de relógio não foi mais considerado ideal.

A simulação do código VHDL comportamental não possui uma carga computacional elevada em relação às simulações subsequentes. Isto torna a análise muito mais rápida e as re-execuções de simulação também factíveis com uma maior frequência. À medida que aumenta a quantidade de informação que deve ser processada pelo simulador, aumenta também o custo de tempo para executar o procedimento de verificação. Isto aconteceu nas simulações com os *netlists* provenientes da síntese lógica e síntese física. Em projetos com um número elevado de portas lógicas este processo pode levar semanas dependendo do número de casos que se almeje verificar, o que, por conseguinte, pode também inviabilizar a entrega do projeto na data prevista.

A estimativa de potência foi obtida com a ferramenta First Encounter (CADENCE, 2009). Os dados de atividade dos sinais são fornecidos através da simulação do circuito no programa ncsim (CADENCE, 2009) e exportados para um arquivo específico, chamado VCD ou *Value Change Dump*. O VCD é um arquivo gerado por ferramentas de simulação

RTL e contém a atividade de comutação do circuito baseado nos vetores de teste utilizados, ou seja, quanto mais os vetores de teste representarem a situação real do circuito, mais realístico também será o cálculo do consumo de potência. A seção principal deste arquivo contém uma série de mudanças de valores ordenados pelo tempo. Estas mudanças de valores indicam a atividade das linhas ou *nets* do circuito. Este arquivo VCD é obtido executando-se os estímulos gerados pelo TOSSIM e traduzidos em transições lógicas de zeros e uns ao algoritmo CentroidM mapeado.

O resultado de potência entregue pelo First Encounter é a soma das potências interna, de comutação e de fuga do circuito durante o período simulado. A dissipação estática de potência é devida à corrente de fuga ou outras correntes que fluem continuamente pela fonte de potência do circuito. O consumo de potência de curto-circuito ocorre devido à corrente direta da fonte de alimentação para o terra durante o processo de comutação de uma porta lógica. A dissipação dinâmica de potência devido à corrente de comutação durante a carga e descarga das capacitâncias de saída. A dissipação dinâmica ocorre durante o processo de comutação dos transistores PMOS e NMOS devido a corrente de curto-circuito e pelo processo de carga e descarga da capacitância de saída. Para a tecnologia CMOS empregada neste trabalho, a potência dinâmica ainda é a fonte dominante de consumo de potência.

6.3 Arquitetura

A Figura 26 mostra o diagrama de topo de uma possível inserção do CentroidM em um nó sensor. Ao mesmo tempo em que o CentroidM pode ser concebido como um ip externo, o mesmo também pode fazer parte do projeto completo de um nó sensor. Desta forma, o motor de localização poderia ser capaz de coletar dados diretamente dos blocos de comunicação. Isto está representado pelos barramentos tracejados na Figura 26. O CentroidM dispensaria a necessidade da utilização do microcontrolador no processo de recebimento das mensagens.

O diagrama de blocos do sistema de localização é mostrado na Figura 27. O sistema possui um bloco para operações aritméticas constituído de dois divisores (OPENCORES, 2009) e um multiplicador. O circuito multiplicador empregado é capaz de multiplicar números de até 16 bits considerando o sinal. A estrutura deste operador foi apresentada em (COSTA; BAMPI; MONTEIRO, 2002) e analisada em face a outras implementações em (COSTA, 2002) e (OLIVEIRA, 2005). Ambos circuitos são puramente combinacionais com entradas e saídas registradas e operam em um único ciclo de relógio. Apesar de

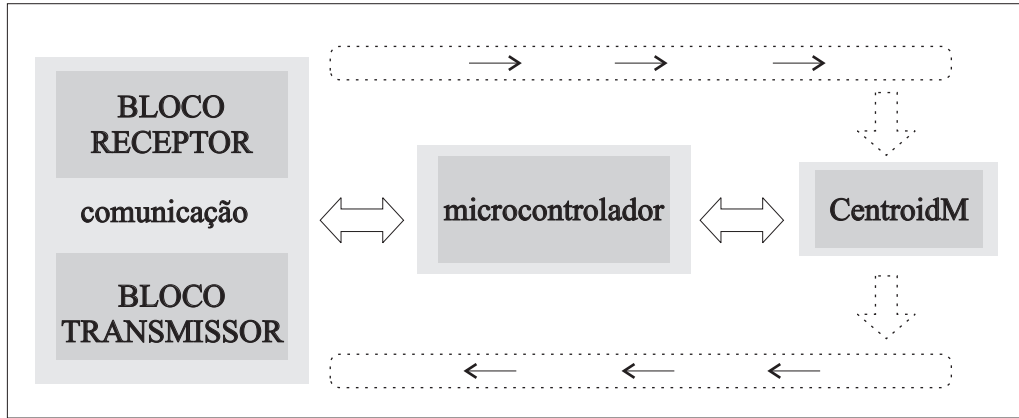


Figura 26: Diagrama de topo simplificado para o CentroidM.

serem utilizados operadores de 16 bits, as coordenadas estão saturadas em 8 bits, bem como a área sobre monitoramento. Os 16 bits são necessários em cálculos intermediários do algoritmo.

O sistema é composto por um bloco inicial responsável pelo tratamento dos dados recebidos. Como mostra a Figura 27, este primeiro bloco é dedicado ao recebimento dos dados de acordo com a identificação de cada nó âncora. As coordenadas x e y são então armazenadas em registradores para posterior utilização na estimativa final. Para cada mensagem recebida de um determinado nó âncora um contador específico é atualizado. Os parâmetros de configuração do algoritmo também são registrados nesta etapa.

Além de um contador dedicado a cada nó de referência responsável por indicar o total de mensagens recebidas durante o período de amostragem t , foram planejados outros contadores, um para cada janela temporal w , que atualizam os dados parciais dos nodos de

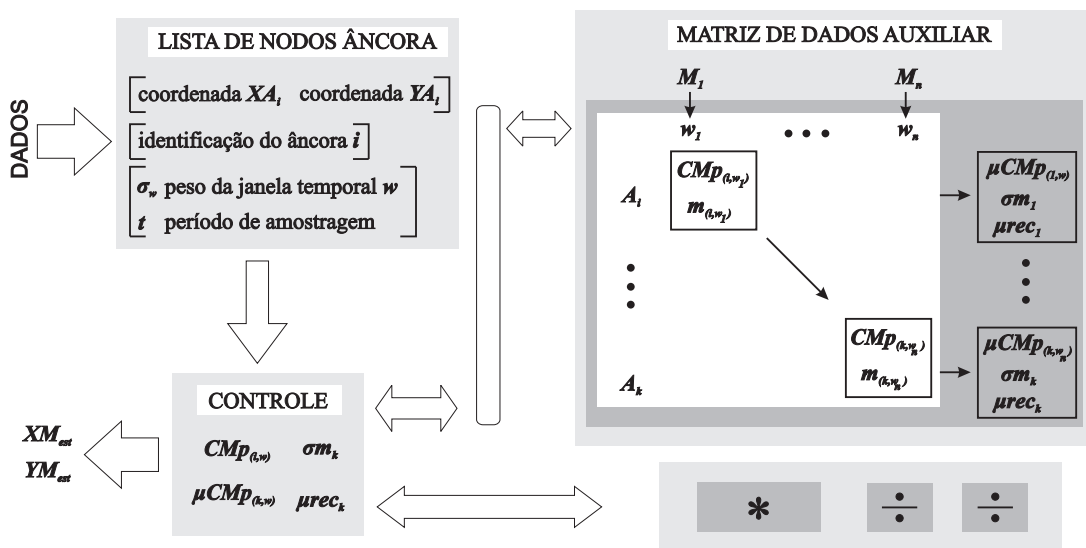


Figura 27: Diagrama de blocos do sistema de localização.

referência. A matriz de registradores apresentada na Figura 27 foi construída e otimizada durante a etapa de síntese lógica. Embora esta não seja a melhor abordagem em um fluxo ASIC, as dimensões reduzidas do CentroidM permitiram esta codificação em VHDL, ao mesmo tempo que tornam o código auto contido e portátil a outras tecnologias sem maiores esforços. Isto é possível uma vez que não foi utilizada uma macro de memória disponibilizada pelo fabricante. A matriz de memória é responsável por manter no máximo 50 registros acerca dos âncoras de acordo com os dados coletados.

O bloco de controle lida com os dados recebidos, enviando-os ao bloco de operações aritméticas ou armazenando-os e acessando-os da matriz de memória. Convém enfatizar que mesmo que o CentroidM realize diversas operações aritméticas, o bloco de controle da Figura 27 é a principal entidade do algoritmo desta Tese. Ele é responsável por manipular os dados do processo de recepção, da matriz de memória e dos operadores aritméticos. Este bloco é elaborado através de quatro máquinas de estado Mealy⁶ (COHEN, 1999), as quais utilizam explicitamente registradores de estado para manter uma representação do estado atual e as saídas são uma função de ambos, entradas e do estado atual.

A Figura 28 apresenta a ordem de execução temporal dos processos, desde o início dos cálculos das métricas de conectividade gerais até a entrega dos resultados de estimativas das coordenadas e, reproduz em linguagem HDL o comportamento do pseudo-código apresentado no Algoritmo 1⁷. Este processo dura 717 ciclos, o que equivale a $71,7\mu s$, considerando um ciclo de relógio a uma frequência de $10MHz$. O diagrama da Figura 28 descreve em cinco fases distintas como as máquinas de estado interagem entre si, bem como a dependência entre uma fase e outra. O processo de recebimento, indicado no topo da Figura 28 é executado sempre que uma mensagem é capturada pelo nó e, apenas ao final do tempo de amostragem t é que tem início as tarefas divididas nas cinco fases.

A flexibilidade do CentroidM permite ao usuário determinar alguns valores de entrada para o sistema. Como mencionado logo acima, é possível configurar os parâmetros de:

- tempo total de amostragem t ;
- tamanho total esperado da amostra. Este valor é empregado no cálculo das métricas de conectividade;
- peso σ_w de cada janela temporal w .

⁶Quando as saídas dependem apenas do estado atual, sem que haja interferência das entradas, a máquina de estados é do tipo Moore.

⁷Seção 4.2.2.

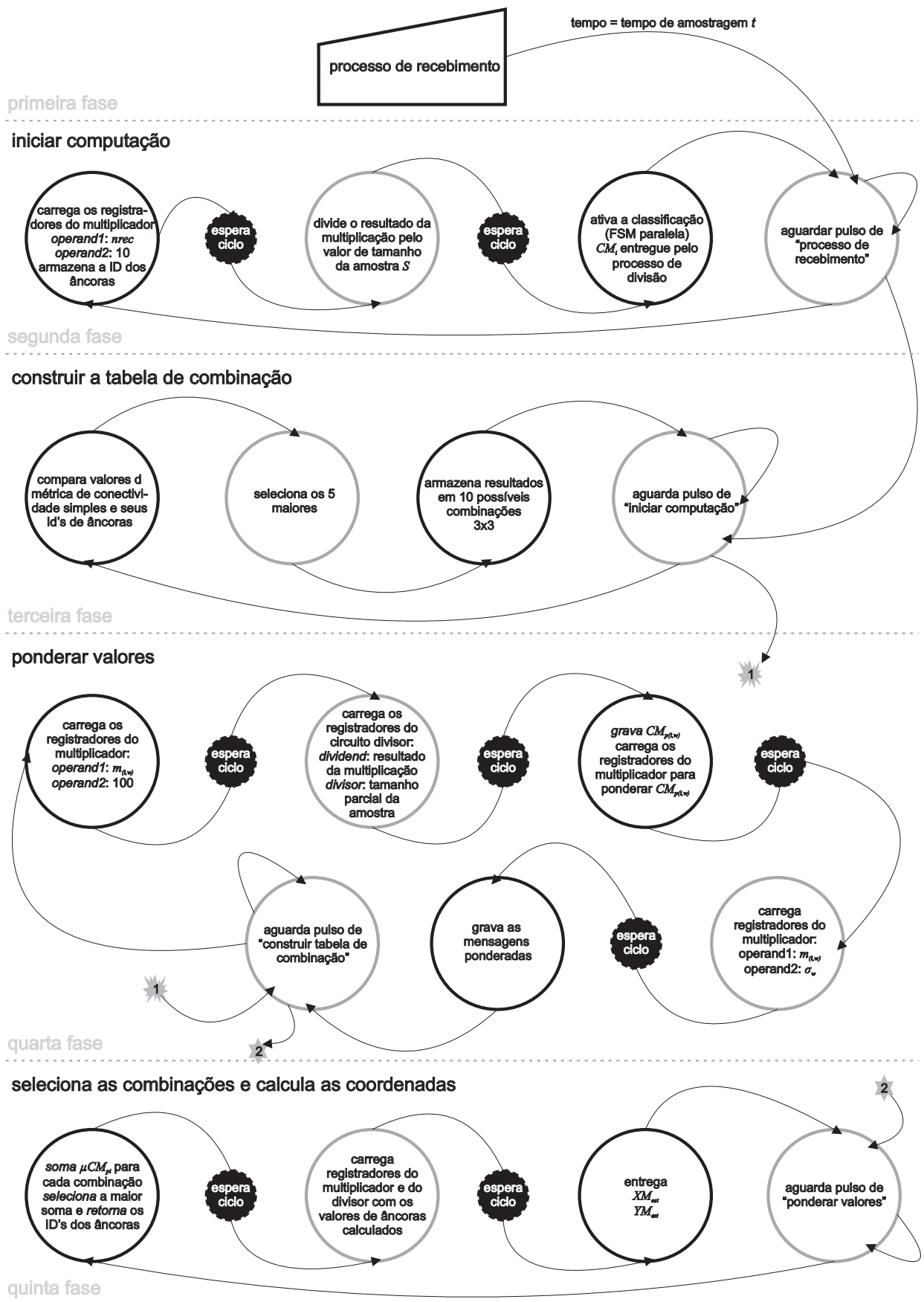


Figura 28: Diagrama do processo de execução das máquinas de estados do circuito dedicado.

Em contrapartida, devido ao fato do CentroidM estar sendo concebido em *hardware* nesta fase do projeto, algumas limitações advêm com esta abordagem. Tais limitações também existem para a implementação em *software*, no entanto podem ser contornadas ou alteradas de maneira dinâmica através da alocação de mais recursos, quando disponíveis. Enquanto *hardware*, tais limitações devem estar explícitas no projeto⁸. São elas:

- o projeto atual restringe o número de janelas temporais a quatro;
- optou-se por fixar o número máximo de âncoras igual a cinquenta;
- as coordenadas dos nodos em movimento bem como dos nodos de referência estão restritas a 8 bits.

6.4 Resumo

O fluxo digital apresentado envolveu o procedimento utilizado para o projeto do CentroidM em *hardware*, verificação funcional e estimativa de consumo e atraso. Para a etapa de verificação foram empregados os estímulos gerados pelo TOSSIM, já descritos no Capítulo 5. A implementação física foi executada seguindo os passos do fluxo de um ASIC com especial atenção ao ambiente de verificação e à análise da energia gasta pelo circuito. A verificação neste caso foi utilizada com o propósito de assegurar o comportamento funcional projetado do circuito e fornecer uma base de dados compatível para a estimativa do consumo de potência. Desta forma, o Capítulo 5 e o Capítulo 6 mostraram as duas metodologias de validação do método proposto no Capítulo 4 e projeto do circuito, considerando ferramentas educacionais e comerciais. Para o caso comercial foram descritos alguns procedimentos que se justificam quando o projeto em questão envolve um número grande de portas lógicas. Entretanto são técnicas que visam diminuir o processo de retrabalho e, caso seguidas, tornam menor a probabilidade de falhas ao final do processo.

⁸Estas limitações estão de acordo com os cenários simulados em *software*.

7 *RESULTADOS OBTIDOS E ANÁLISE DE DESEMPENHO*

Nesta parte do trabalho são apresentados os resultados obtidos com os algoritmos Centroid e CentroidM. A métrica mais importante para avaliar qualquer algoritmo de localização é o Erro de Localização¹ (EL), o qual pode ser traduzido pela Equação 7.1. Uma medida conveniente para esta métrica é a média do erro sobre todos os nodos envolvidos.

$$EL = \sqrt{(X_{estimado} - X_{real})^2 + (Y_{estimado} - Y_{real})^2} \quad (7.1)$$

Outra medida muito útil para avaliar o desempenho de um algoritmo é o seu tempo de processamento. Este tempo de processamento é mais tarde traduzido na energia² despendida pelo método, a ser executado sobre uma determinada plataforma. Embora este tópico seja muito pouco explorado pela comunidade científica nesta área, a energia despendida pelo processo de aquisição das coordenadas tem um papel importante na vida útil de uma rede, sobretudo quando é considerado um cenário móvel.

A simulações presentes neste Capítulo evidenciam o desempenho do CentroidM face ao Centroid original. São mostrados resultados para cenários estáticos e móveis, utilizando a ferramenta TOSSIM. Para os cenários onde a mobilidade dos nodos está presente, as simulações foram divididas de acordo com os modelos de mobilidade descritos na Seção 5.4. Também são variadas as velocidades mínima e máxima dos nodos em cada modelo de mobilidade. Para ambas situações, isto é, com ausência e presença de movimento, a rede foi submetida a um ambiente com muito e pouco ruído, o que alterou significativamente³ a quantidade de pacotes de informação captados pelos nodos.

¹O Erro de Localização equivale à distância entre dois pontos: a estimativa calculada e a posição real do nó.

²Energia = potência x tempo de execução.

³Houve uma redução média de cerca de 70% no número de mensagens recebidas de um ambiente com pouco ruído para outro com muito ruído.

7.1 Topologia Estática

Em uma rede estática, os valores estimados tendem para um resultado estacionário após algum tempo. Esta característica é proeminente em algoritmos que precisem de um tempo de convergência descrito em sua programação, necessário para obtenção de estimativas, como em (DATTA et al., 2006) ou (HU; EVANS, 2004) por exemplo. Tanto o Centroid original quanto o CentroidM não possuem quaisquer períodos de ajustes iniciais. Este fato, por um lado evidencia que não há um tempo destinado ao refinamento dos dados, mas por outro expõe o diminuto tempo de processamento do método, o qual alcança o melhor resultado de localização já na primeira estimativa. Além disto, o período de refinamento dos resultados, ou convergência, de algoritmos de localização está atrelado ao recebimento e envio de mensagens com outros nodos da rede.

Até meados do ano de 2004, muito pouco esforço havia sido empregado pela comunidade científica acerca do desenvolvimento de algoritmos de localização que operassem em redes móveis e, acima de tudo, cujos nodos não possuíssem controle sob seus movimentos. Obviamente excluem-se desta afirmação os métodos baseados em nodos providos de atuadores para o movimento, ou robôs. Os métodos desenvolvidos enfocavam o processo de localização sob um ponto de vista estático. O movimento, portanto, era algo considerado esporádico e/ou indesejável, enquanto que a abordagem indicada em tais casos era a de apenas re-executar o algoritmo para estabelecer as novas coordenadas. Por exemplo, quando o mantenedor da rede manualmente movimentava um nó da rede, por motivos de substituição devido à avaria ou mesmo por decidir um novo ponto sob controle dos sensores de tal nodo, havia um movimento artificial e talvez não previsto. Aquele nó em particular deveria então localizar-se na rede para que os seus registros extraídos do meio fizessem algum sentido.

Como já mencionado ao longo do Capítulo 4, o algoritmo Centroid (BULUSU; HEIDEMANN; ESTRIN, 2000) foi concebido, simulado e experimentado por seus autores em um cenário estático. Da mesma forma, como também já exposto, este algoritmo possui características extremamente favoráveis de serem implementadas em um sistema móvel. No entanto, é desejável que um algoritmo de localização, ou mesmo qualquer outro sistema projetado, possa ser utilizado para cobrir o maior número possível de situações. Por esta razão o CentroidM será confrontado com o Centroid original em um cenário sem a presença do movimento dos nodos.

A Figura 29 apresenta uma distribuição de nodos desconhecidos, ou seja, nodos a serem localizados e nodos âncora. Estão representados neste cenário 25 nodos, 9 dos

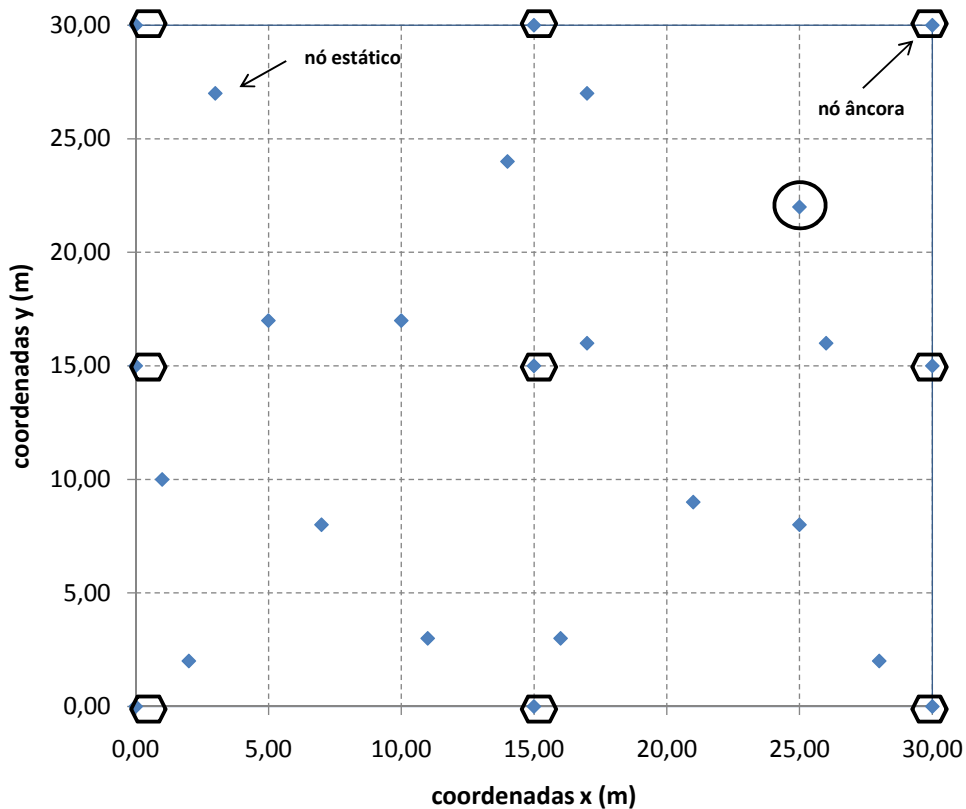


Figura 29: Representação de uma distribuição aleatória de nodos estáticos a serem localizados. Os pontos envoltos por hexágonos representam os nodos âncora da rede.

quais são âncoras e o restante são nodos desconhecidos. Os nodos desconhecidos foram dispostos de maneira aleatória em uma área definida em $30m \times 30m^4$. O alcance do rádio foi definido através de simulação para o valor de 18 metros⁵. Para chegar a este dado foram consideradas simulações em um ambiente com pouco ruído (Lago Lagunita) onde foi constatada a distância a partir da qual um nó passava a não mais receber mensagens de um nó de referência. É possível ainda que nodos distantes mais de 18 metros⁶ de nodos de referência recebam mensagens destes, visto que tal valor foi obtido por uma média através de diversas simulações.

O sucesso do algoritmo de localização depende da densidade de pontos de referência disponíveis na área de abrangência do raio de alcance dos nodos (SAVARESE; RABAEY; LANGENDOEN, 2002; SAVVIDES; HAN; STRIVASTAVA, 2001), o que traduz-se na conectividade da rede. A disposição dos nodos âncora indicados na Figura 29 impede a existência de pontos cegos neste cenário, ou seja, não há pontos na região abordada para os quais

⁴Esta área representa uma região 9 vezes superior à área abordada em (BULUSU; HEIDEMANN; ESTRIN, 2000).

⁵Este valor de 18 metros será referenciado apenas como o raio de alcance r dos nodos.

⁶Este valor não impacta diretamente nos resultados simulados e é utilizado apenas para a apresentação dos gráficos.

não seja possível receber mensagens de pelos menos um nó de referência.

A Figura 30 mostra a simulação de 500 segundos da topologia de rede apresentada na Figura 29. A flutuação das linhas, as quais representam o erro calculado em função do raio de alcance do dispositivo de rádio dos nodos, indica a não linearidade do raio de alcance dos dispositivos presentes no cenário simulado. Embora este aspecto retorne uma flutuação para os resultados observados na Figura 30, isto também aproxima a simulação de um cenário real. Além disto, resultados experimentais mostraram que a taxa de recebimento de pacotes varia significativamente não apenas de acordo com a região onde o nó se encontra, mas sobretudo com o passar do tempo (CERPA et al., 2005).

Tal irregularidade na recepção dos dados traduz-se no fato de ora um nó receber mais mensagens de um âncora, ora de outro. Este aspecto provoca distorções no processo de cálculo das coordenadas, pois ambos algoritmos baseiam-se em métricas de conectividade relacionadas com as mensagens recebidas. Cada ponto do gráfico da Figura 30 que forma as curvas apresentadas é o resultado da média aritmética do erro de todos os nodos desconhecidos da rede, em 151 momentos dos 500 segundos simulados. Desta forma, os valores apresentados também consideram as flutuações acumuladas para todos os nodos da rede em cada ponto, o que aumenta a distorção apresentada.

Caso selecione-se apenas um nó da rede para apresentar seu comportamento ao longo

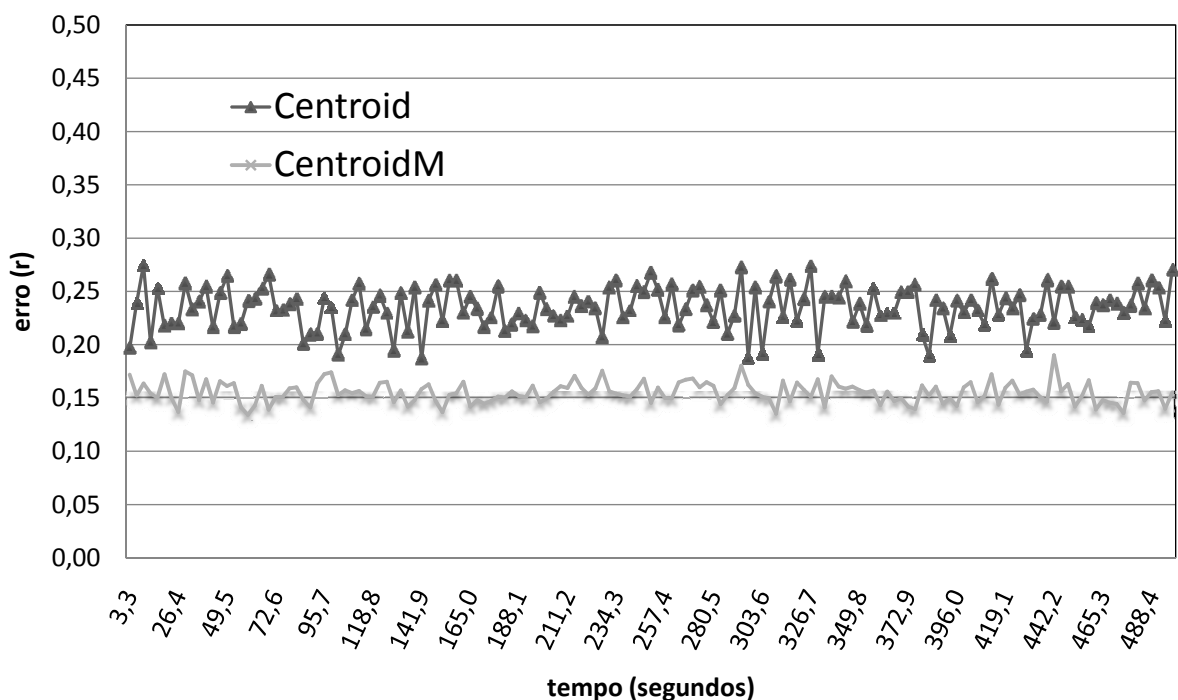


Figura 30: Simulação da rede estática descrita na Figura 29.

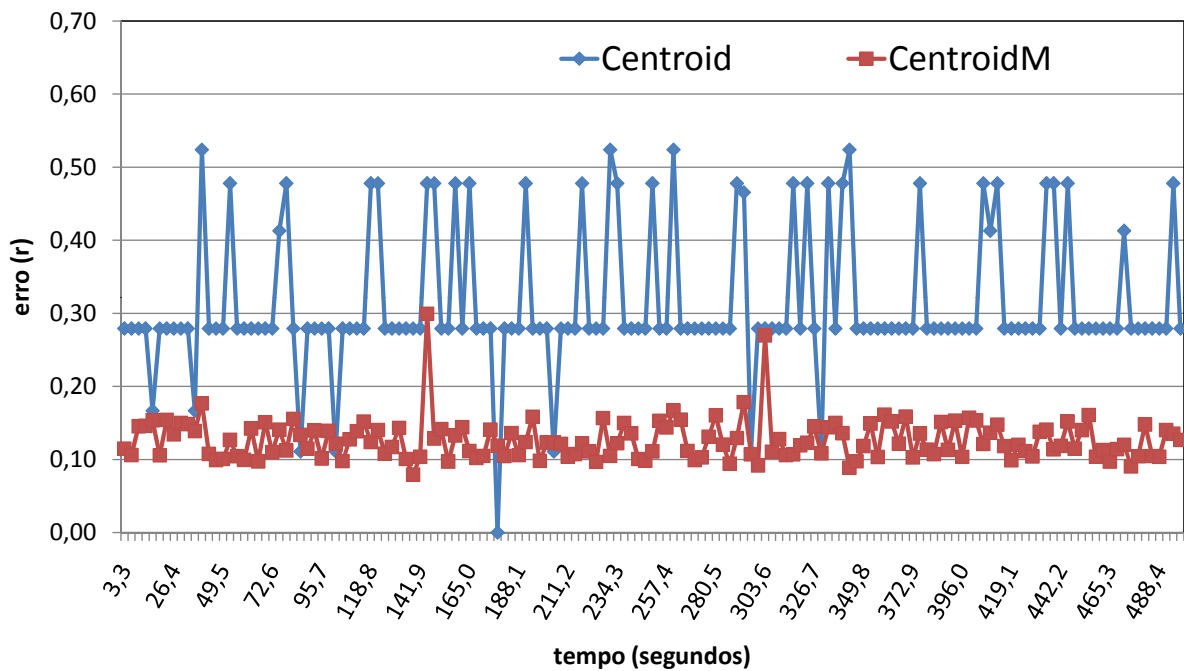


Figura 31: Erro para o nó envolto por uma circunferência na Figura 29 em um cenário com ausência de movimento.

de todo o período de simulação, ter-se-á uma representação como mostrada na Figura 31. O nó envolto pela circunferência da topologia mostrada na Figura 29 foi analisado durante o período de simulação. Os resultados apresentados no gráfico da Figura 31 referem-se ao erro calculado para as estimativas entregues pelo Centroid e pelo CentroidM. Como pode ser visto, há diversos momentos onde os erros calculados de ambos algoritmos assumem valores distintos.

No entanto, também é possível perceber que os valores de erro apresentados pela curva do algoritmo Centroid assumem posições muito distantes entre si. A quantificação da dispersão destes valores, ou seja, o desvio padrão para a curva do Centroid é igual a 0,10 enquanto que este valor equivale a 0,03 para o CentroidM. Verifica-se, portanto uma dispersão dos dados 239% maior para o Centroid em relação ao CentroidM. Apesar de o desvio padrão apresentar uma característica importante do CentroidM, este dado estatístico não é suficiente para a análise do desempenho do algoritmo. De forma complementar, a análise dos dados do gráfico da Figura 31 retornou um erro médio de posicionamento em função do raio r igual a 0,31 para o Centroid e 0,13 para o CentroidM. Com isto, tem-se que o CentroidM apresenta uma redução de 59% para o erro médio, aliado um desvio padrão 71% menor, quando comparado ao Centroid. Além disto, é importante notar que as estimativas entregues pelo Centroid flutuaram entre 7 valores diferentes de

Tabela 4: Primeiros pontos do gráfico mostrado na Figura 31.

tempo (s)	Centroid				CentroidM			
	Xest	Yest	Erro (m)	Erro (r)	Xest	Yest	Erro (m)	Erro (r)
3,30	30,00	22,50	5,02	0,28	25,12	19,93	2,07	0,12
6,60	30,00	22,50	5,02	0,28	25,91	20,33	1,91	0,11
9,90	30,00	22,50	5,02	0,28	25,52	24,56	2,62	0,15
13,20	30,00	22,50	5,02	0,28	25,45	24,59	2,63	0,15
16,50	25,00	25,00	3,00	0,17	25,19	24,76	2,76	0,15
19,80	30,00	22,50	5,02	0,28	25,88	20,31	1,90	0,11
23,10	30,00	22,50	5,02	0,28	25,79	24,66	2,77	0,15
26,40	30,00	22,50	5,02	0,28	26,02	24,19	2,42	0,13
29,70	30,00	22,50	5,02	0,28	25,67	24,61	2,70	0,15
33,00	30,00	22,50	5,02	0,28	25,51	24,62	2,67	0,15
36,30	25,00	25,00	3,00	0,17	25,20	24,49	2,50	0,14
39,60	30,00	30,00	9,43	0,52	25,13	25,18	3,18	0,18
42,90	30,00	22,50	5,02	0,28	25,79	20,23	1,94	0,11
46,20	30,00	22,50	5,02	0,28	25,43	20,27	1,79	0,10
49,50	30,00	22,50	5,02	0,28	25,42	20,24	1,81	0,10
52,80	30,00	15,00	8,60	0,48	25,59	24,20	2,28	0,13
56,10	30,00	22,50	5,02	0,28	25,37	20,16	1,88	0,10
59,40	30,00	22,50	5,02	0,28	25,85	20,42	1,79	0,10
62,70	30,00	22,50	5,02	0,28	25,87	24,41	2,56	0,14
66,00	30,00	22,50	5,02	0,28	25,13	20,25	1,75	0,10
69,30	30,00	22,50	5,02	0,28	25,50	24,67	2,72	0,15

OBS: os valores das coordenadas Xest e Yest estão em unidades de metro.

Xref (m)	Yref (m)
25,00	22,00

erro ao longo de todo o período de simulação⁷. Isto é devido à limitação de nodos de referência disponíveis ao alcance de rádio do nó e o consequente, também limitado, número de possíveis combinações de âncoras para o cálculo das estimativas.

A Tabela 4 apresenta os primeiros 21 pontos de um total de 151 que formam o gráfico da Figura 31. Pelos valores apresentados é possível verificar a flutuação dos dados de erro do Centroid em torno de 4 valores distintos, ao passo que para o CentroidM este número é de 8 para o mesmo período analisado. Este ponto em particular evidencia outra característica do método de localização CentroidM proposto nesta Tese. Após selecionar os nodos âncora que farão parte do processo de cálculo, o CentroidM quantifica a importância de cada um destes nodos pelo processo descrito no Capítulo 4 e transfere esta importância para os valores de coordenadas destes mesmos âncoras. Isto faz com que os valores calculados para o erro de posicionamento oscilem por um conjunto muito maior de possibilidades, mas permaneçam contudo mais próximos uns dos outros.

O gráfico da Figura 32 considera uma região de $50m \times 50m$ em uma situação onde todos os nodos estão parados. O que é mostrado com estes dados é a diminuição da variação dos erros em comparação com a simulação apresentada na Figura 30. Isto é devido aos nodos desconhecidos receberem menos *beacons*⁸ pelo fato de estarem mais

⁷O valor de erro igual a zero para o algoritmo Centroid na simulação apresentada na Figura 31 indica na verdade que, neste ponto, o Centroid não foi capaz de entregar uma estimativa.

⁸A literatura também faz referência à palavra *beacon*, mencionada na Seção 2.2.1, como sendo a

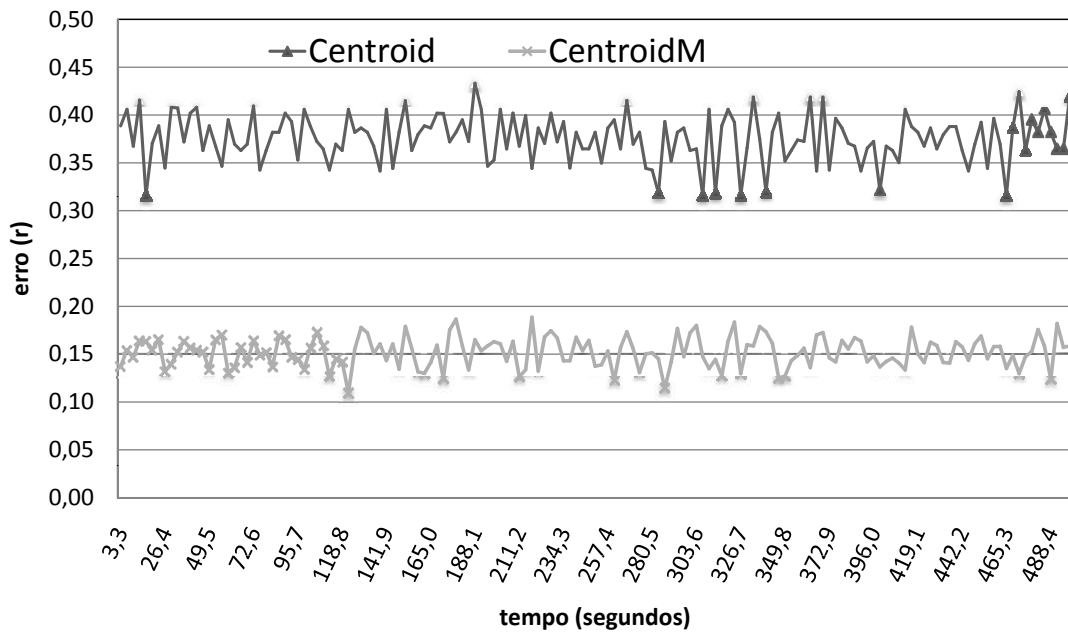


Figura 32: Erro médio de todos os nodos envolvidos na rede para uma área de $50m \times 50m$.

distantes dos nodos de referência.

A Figura 33 apresenta o erro para os algoritmos Centroid e CentroidM para um único nó, referente à simulação mostrada no gráfico da Figura 32. Este nó da Figura 33 permanece imóvel durante todo o tempo de simulação e acaba por receber os *beacons* sempre dos mesmos âncoras. Desta forma, a probabilidade de determinados âncoras possuírem sempre as mesmas métricas de conectividade é grande e por isto o erro ao longo do tempo, relativo ao algoritmo Centroid, permanece constante para este nó. O mesmo não ocorre para o CentroidM. O fato do CentroidM quantizar as coordenadas dos nodos de referência, faz com que a variação de valores de erro calculados também esteja presente neste cenário. Desta forma, o desvio padrão para o Centroid é igual a zero e equivalente a 0,05 para o CentroidM. Apesar disto, o erro médio deste nó ao longo de todo o período de simulação é de apenas $0,1r^9$, ou seja, uma redução de 72,2% em relação aos $0,36r$ entregues pelo Centroid. É claro que este ganho não se estende para todos os nodos da rede e os valores finais apresentados são uma média geral. A Tabela 5 reúne o resultado médio do erro de posicionamento para as duas topologias estáticas mencionadas.

As Figuras 34 e 35 mapeiam a informação de erro de posicionamento para os algoritmos Centroid e CentroidM dentro de uma área de $30m \times 30m$. Este cenário contém 952 nodos, espaçados em 1 metro. Os nodos âncora, em número de 9, estão localizados conforme a topologia definida na Figura 29. Cada nó desconhecido calculou a sua posição

mensagem enviada pelo nó âncora.

⁹CentroidM.

Tabela 5: Erro médio de todos os nodos envolvidos no cenário - impacto do aumento de área

	Centroid	CentroidM	Dif.(%) CentroidM→Centroid
Área $30m \times 30m$ (Figura 30)	0,235r	0,155r	-33,99
Área $50m \times 50m$ (Figura 32)	0,379r	0,152r	-59,89

36 vezes, sendo então extraída a média aritmética destes valores. Assim sendo, o total de cálculos de posição, para a construção dos gráficos, realizado pelos 952 nodos que compõem a rede foi de 34.272.

Como pode ser observado, o erro de localização para o algoritmo Centroid, mostrado na Figura 34, é menor nas regiões próximas às bordas externas do cenário proposto e às internas, construídas pela conexão imaginária dos nodos intermediários com o nodo central, delimitando 4 quadrantes. Isto é devido ao fato de os nodos destas regiões receberem mais *beacons* de nodos que formam uma relação de colinearidade entre si e o nó desconhecido, chegando a situações de erro nulo. Em contrapartida, o recebimento de mensagens de um número maior de nodos âncoras, à medida que o nó desconhecido localiza-se na região de sobreposição do raio de alcance de nodos de referência, faz com que um número também maior de valores de métricas de conectividade aproxime-se do limiar estabelecido. No entanto, apenas algumas delas atingirão tal limiar ao término do tempo de amostragem t .

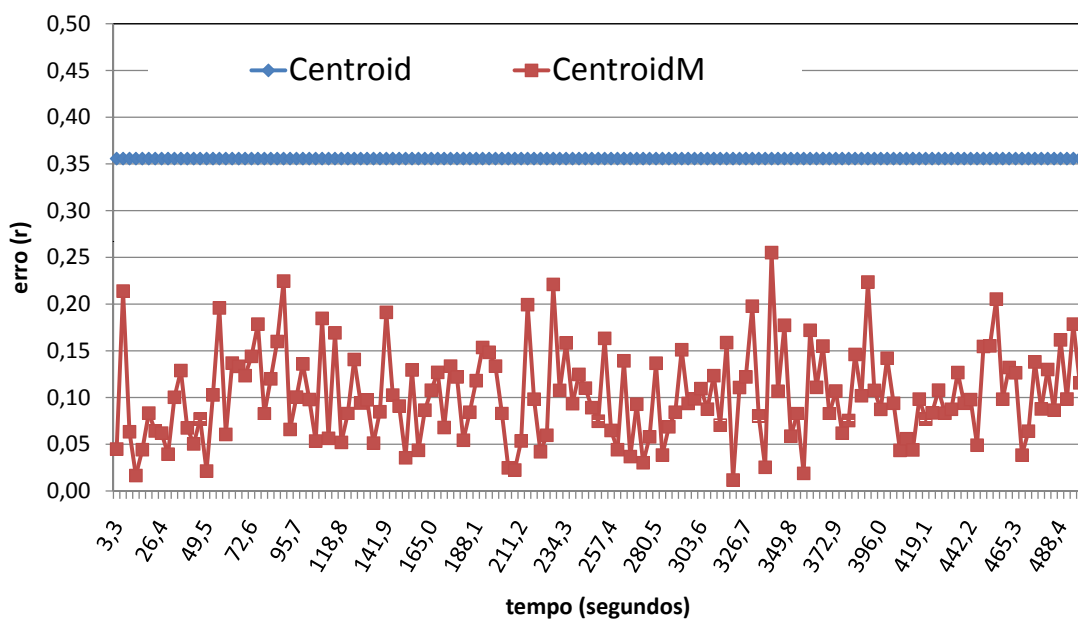


Figura 33: Erro para um único nó da simulação mostrada na Figura 32 ao longo de 500 segundos.

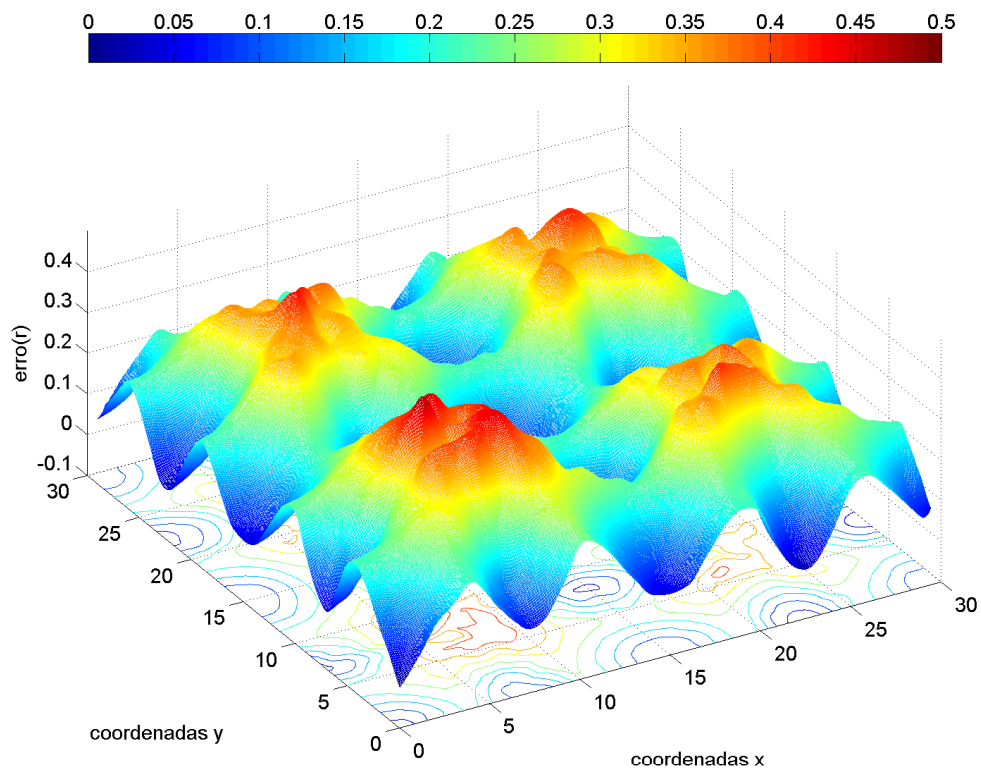


Figura 34: Erro de localização versus a posição para o algoritmo Centroid (BULUSU; HEIDEMANN; ESTRIN, 2000): mapeamento topológico do erro.

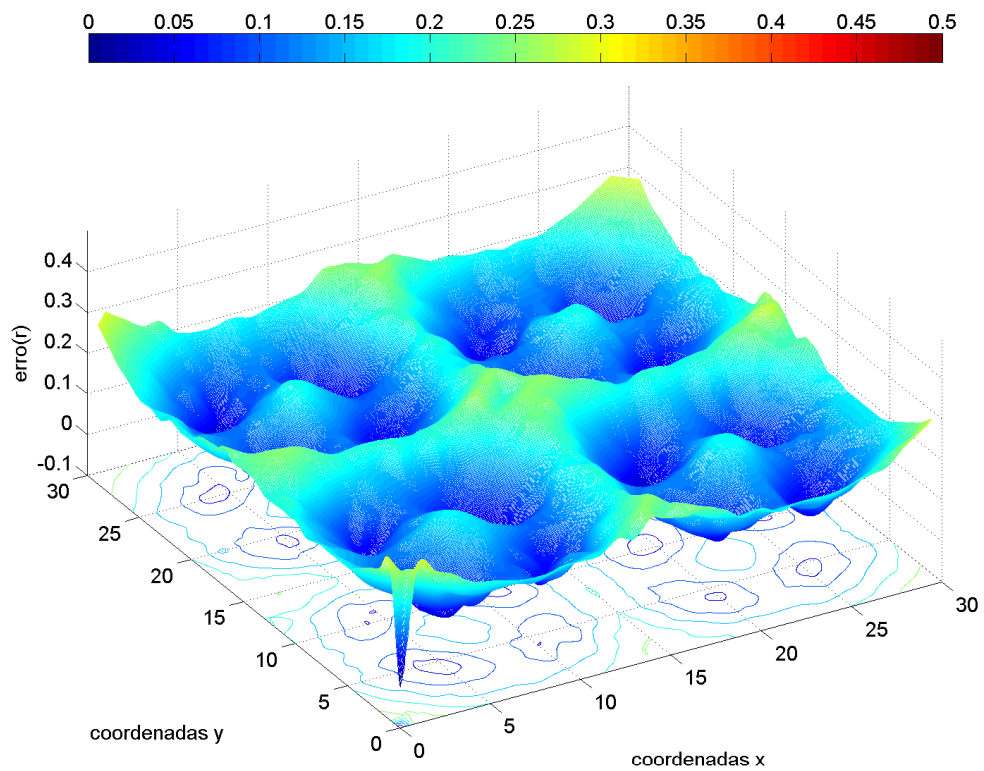


Figura 35: Erro de localização versus a posição para o trabalho **proposto** nesta Tese: mapeamento topológico do erro.

O mapeamento topológico do erro para o algoritmo CentroidM está indicado na Figura 35. Cabe ressaltar neste gráfico o índice de erro ligeiramente superior face ao Centroid nas regiões limítrofes dos 4 quadrantes. Este aspecto está ligado ao fato do método CentroidM optar pela escolha de 3 nodos âncora em todos os processos de cálculo das estimativas de posicionamento. Contudo, estes desvios são mitigados pelo fato do algoritmo quantizar as coordenadas dos nodos de referência selecionados. Embora a análise dos gráficos da Figura 35 para o CentroidM e, da Figura 34 para o Centroid, apresentem que o primeiro produz estimativas menos precisas em determinadas regiões que aquelas entregues pelo segundo, observa-se que os resultados mais negativos do Centroid são muito superiores aos do CentroidM. Este fato está relacionado com as etapas abordadas no Capítulo 4 de seleção dos nodos âncora e ponderação de suas coordenadas. A média do erro sobre todos os nodos da topologia resultou em um ganho 33% favorável ao CentroidM. É importante salientar ainda, que as regiões onde os nodos âncoras estão localizados, para ambos algoritmos, estão indicadas com um erro de valor igual a zero.

7.2 Estudo da Mobilidade

Uma vez estabelecido o desempenho do algoritmo CentroidM em um cenário estático, o método foi submetido ao movimento da rede. Para avaliar os algoritmos Centroid e CentroidM diversas simulações foram executadas. Os resultados de simulação extraídos representam uma média dos erros referentes a todas as posições calculadas por todos os nodos da rede, sejam eles móveis ou fixos. Os nodos a terem a sua posição descoberta foram submetidos a movimentos aleatórios, com e sem pausas durante o seu trajeto. Ainda, para cada um dos modelos de mobilidade foi analisada a influência de dois parâmetros na rede de sensores, quais sejam:

- velocidade dos nodos âncora;
- variação na recepção de pacotes.

Como será visto logo a seguir, o desempenho do CentroidM também evidencia uma diferença significativa em relação ao Centroid quando submetido ao cenário com movimento. Outros algoritmos de localização baseados na mobilidade possuem o desempenho prejudicado com a ausência de movimento. O MCL (HU; EVANS, 2004) por exemplo, já mencionado no Capítulo 2, apresenta valores de erro entre $0,8r^{10}$ e $1,0r$ para o caso no

¹⁰Em (HU; EVANS, 2004), para obter este resultado os autores estabeleceram o raio r de alcance do dispositivo de rádio dos nodos igual a 50 metros e representando um círculo perfeito.

Tabela 6: Impacto da mobilidade sobre o erro calculado considerando o *Random Walk Mobility Model*.

variação da velocidade	0,0m/s ->	0,5m/s ->	1,0m/s ->	1,5m/s ->	2,0m/s ->	2,5m/s ->	3,0m/s ->	3,5m/s ->
	0,5m/s	1,0m/s	1,5m/s	2,0m/s	2,5m/s	3,0m/s	3,5m/s	4,0m/s
erro (r)								
Centroid	0,270	0,243	0,271	0,273	0,259	0,270	0,293	0,292
CentroidM	0,156	0,148	0,156	0,169	0,160	0,162	0,164	0,166
	-42,04%	-39,10%	-42,49%	-38,20%	-38,28%	-39,96%	-44,01%	-43,04%

qual os nodos âncora estão parados e os nodos desconhecidos movem-se a uma velocidade de $0, 2r$. De fato, não há em (HU; EVANS, 2004) simulações acerca de um cenário estático. Este método, publicado no ano de 2004, foi o primeiro trabalho neste sentido a considerar movimento não controlado dos nodos e obtém vantagens sobre este movimento, ao passo que é ineficaz em uma rede estática.

7.2.1 Mobilidade Aleatória

O cenário sob interesse para a execução do algoritmo de localização considera uma rede de sensores onde os nodos estão em movimento contínuo e imprevisível, isto é, não há conhecimento prévio acerca de sua trajetória futura, tampouco se os mesmos possuem qualquer tipo de controle sobre a sua direção, sentido ou velocidade. O modelo estudado nesta Seção não considera pausas durante o trajeto do nodo. Apesar da mobilidade ser algo indesejável em alguns esquemas de localização, os resultados a seguir evidenciam que este aspecto não prejudica a precisão do método proposto nesta Tese.

A Tabela 6 reúne os valores de erro calculado para diferentes variações da velocidade dos nodos da rede, considerando o *Random Walk Mobility Model*. Como pode ser comparado na Tabela 6 o CentroidM apresenta uma redução de erro de medição que varia de $-38, 20\%$ a $-44, 01\%$ em relação ao Centroid original. É possível também observar a não linearidade dos dados à medida que os valores de velocidade são alterados. Não é possível configurar na simulação qualquer controle sob a região que os nodos irão percorrer. Como visto na Seção 7.1, o mapeamento topológico do erro calculado indica uma variação significativa deste dado de acordo com a região que o nó se encontra. Assim sendo, embora a redução do erro em valores absolutos não siga um padrão ao longo das simulações, o algoritmo CentroidM apresenta um impacto significativo na redução do erro para todos os casos observados. O gráfico da Figura 36 mostra estes dados, os quais foram simulados para uma região de $30m \times 30m$.

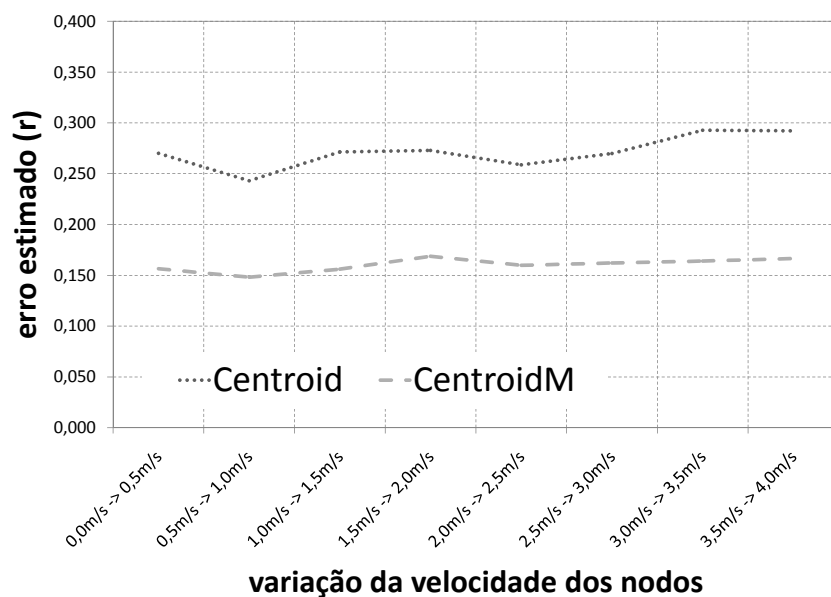
Em um ambiente altamente ruidoso, a recepção do sinal é afetada (LEE; CERPA; LEVIS, 2007) de forma que um número menor de pacotes é recebido pelos nodos em relação

Tabela 7: Impacto sobre o erro calculado considerando um ambiente ruidoso.

variação da velocidade	0,0m/s -> 0,5m/s	0,5m/s -> 1,0m/s	1,0m/s -> 1,5m/s	1,5m/s -> 2,0m/s	2,0m/s -> 2,5m/s	2,5m/s -> 3,0m/s	3,0m/s -> 3,5m/s	3,5m/s -> 4,0m/s
com ruído	0,164	0,155	0,159	0,144	0,160	0,155	0,173	0,170
sem ruído	0,155	0,146	0,155	0,166	0,157	0,160	0,163	0,164

a um ambiente com muito pouco ruído. Todos os cenários de variação de velocidade apresentados no gráfico da Figura 36 foram submetidos a uma redução drástica de 70% no número de pacotes recebidos pelos nodos a terem a sua localização descoberta. A Tabela 7 apresenta os valores de erro para o CentroidM em um ambiente ruidoso e com pouco ruído.

Pelo gráfico da Figura 37 pode-se observar a proximidade dos resultados de cálculo de erro para o CentroidM em um ambiente ruidoso e com pouco ruído. Os dados referentes ao algoritmo Centroid para o ambiente ruidoso não estão sendo mostrados. Sob estas circunstâncias, o método Centroid proposto em (BULUSU; HEIDEMANN; ESTRIN, 2000) realiza um número muito pequeno de estimativas, menos de 5% em relação ao CentroidM. Este fato deve-se à diminuição do número de mensagens recebidas pelos nodos desconhecidos, fazendo com que este montante não ultrapasse o limiar preestabelecido para poder incluir um nó de referência no processo de cálculo. Ao descreverem as suas trajetórias, os nodos aproximam-se e afastam-se de nodos âncora da rede. Isto provoca o recebimento de mensagens de um número maior de nodos de referência que, aliado ao fato de um menor número total de mensagens $N_{recv}(i, t)$ de posição recebidas de cada um deles, provoca

Figura 36: Impacto da mobilidade sobre o erro calculado para o *Random Walk Mobility Model*.

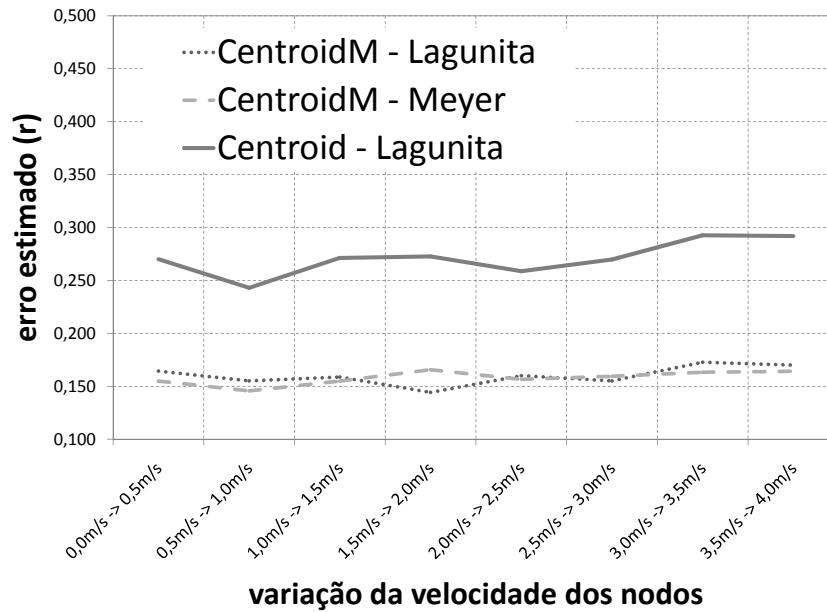


Figura 37: Impacto do ruído - *Random Walk Mobility Model*.

menores valores de métricas de conectividade CM_i para cada nodo de referência A_i . Como no algoritmo Centroid apenas as coordenadas dos nodos de referência cujas métricas de conectividade forem maiores que um limiar estabelecido CM_{thresh} são consideradas, são geradas muito poucas estimativas.

7.2.2 Mobilidade Aleatória com Inserção de Pausas

Da mesma forma como abordado na Seção anterior para o *Random Walk Mobility Model*, observa-se no gráfico da Figura 38 que o algoritmo CentroidM produz reduções significativas dos valores dos erros calculados em relação ao Centroid original na presença do *Random Waypoint Mobility Model*. O algoritmo Centroid, com a inserção das pausas durante o movimento consegue obter melhores resultados em relação às simulações com o *Random Walk Mobility Model*. Tais paradas entre um período de viagem e outro, disponibilizam mais tempo ao Centroid para que o mesmo possa receber mensagens e, portanto entregar melhores estimativas. Contudo, este fato não é suficiente para apresentar a curva do Centroid nas simulações com presença de ruído mostradas no gráfico da Figura 39. Apenas 7% do total de estimativas computadas pelo CentroidM são entregues pelo Centroid.

A partir dos dados apresentados é possível inferir que o algoritmo Centroid (BULUSU; HEIDEMANN; ESTRIN, 2000) não possui um bom desempenho para a localização de nodos em redes móveis. De acordo com a teoria do algoritmo fica fácil compreender isto. A

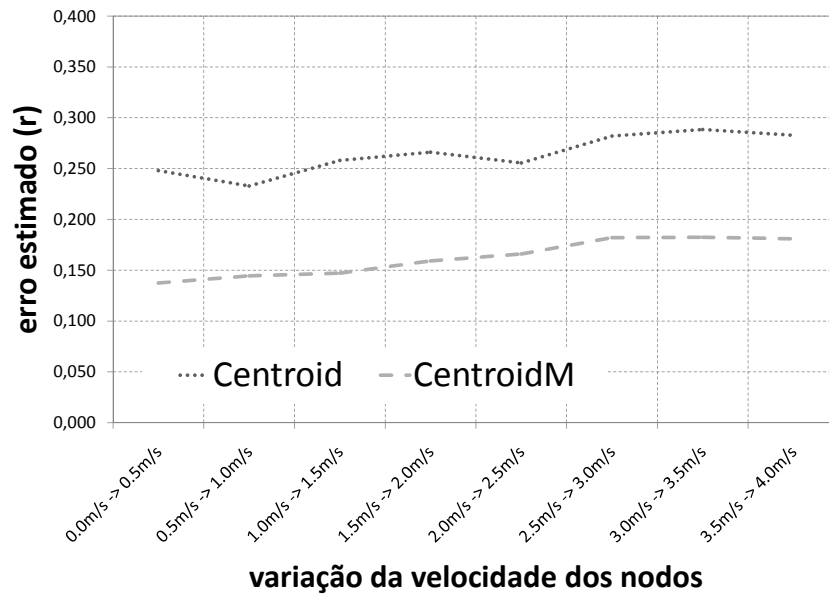


Figura 38: Impacto da mobilidade para o *Random Waypoint Mobility Model*.

escolha dos âncoras a serem utilizados no processo de cálculo da posição está diretamente atrelada ao número de mensagens recebidas de cada âncora. Caso este número não ultrapasse um determinado limite mínimo, esta âncora é descartada. No entanto, existe a possibilidade de um nó móvel ouvir muitas mensagens de uma âncora no início do processo de amostragem e iniciar a recepção de *beacons* de outra âncora, para o qual está se aproximando, no final do período de amostragem. Isto provocará que apenas as coordenadas do primeiro sejam levadas em consideração para a estimativa da localização do nó móvel, incorrendo em erros de cálculo.

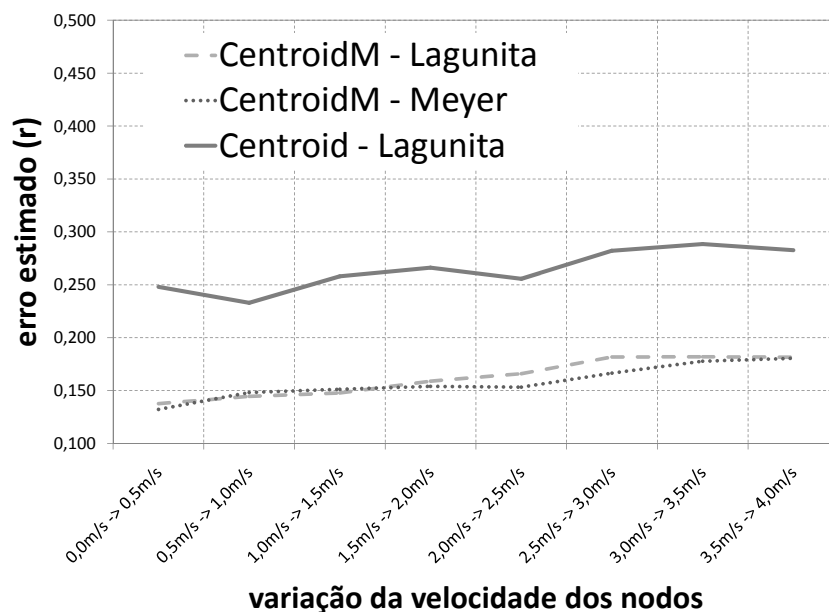


Figura 39: Impacto do ruído - *Random Waypoint Mobility Model*.

A Tabela 8 apresenta os valores de posição estimada para os algoritmos Centroid e CentroidM, onde os 36 pontos apresentados descrevem o movimento do nó número 11 da Figura 21. Como pode ser observado na Tabela 8, existem momentos que o algoritmo Centroid não consegue entregar estimativas de posicionamento. Apesar do erro mais elevado no início do movimento, a média calculada para o algoritmo ficou estabelecida em $0,17r$, ou seja, cerca de 55% menor que o valor de $0,38r$ apresentado pelo Centroid original.

A estimativa da localização realizada de forma distribuída envolve uma trilateração, mas sem as medidas diretas das distâncias entre os nodos desconhecidos e os nodos de

Tabela 8: Valores de erro para o nó número 11 da Figura 21.

Xref	Yref	tempo (s)	Centroid				CentroidM			
			Xest	Yest	Erro (m)	Erro (r)	Xest	Yest	Erro (m)	Erro (r)
12,60	23,50	3,30	15,00	30,00	6,93	0,38	7,11	25,67	5,90	0,33
13,30	22,60	6,60	15,00	30,00	7,59	0,42	7,37	25,85	6,76	0,38
14,00	21,80	9,90	15,00	30,00	8,26	0,46	11,20	26,12	5,15	0,29
14,70	20,90	13,20	15,00	30,00	9,10	0,51	11,04	21,60	3,73	0,21
15,40	20,10	16,50	nan	nan	-	-	10,70	21,13	4,82	0,27
16,10	19,20	19,80	nan	nan	-	-	18,58	21,03	3,08	0,17
16,80	18,40	23,10	15,00	30,00	11,74	0,65	19,36	21,05	3,68	0,20
17,50	17,50	26,40	30,00	15,00	12,75	0,71	20,82	19,47	3,86	0,21
18,20	16,70	29,70	nan	nan	-	-	20,05	20,83	4,52	0,25
18,80	15,80	33,00	nan	nan	-	-	21,28	13,97	3,09	0,17
19,50	15,00	36,30	nan	nan	-	-	21,77	15,07	2,28	0,13
20,20	14,10	39,60	30,00	15,00	9,84	0,55	22,11	11,12	3,53	0,20
20,90	13,30	42,90	30,00	15,00	9,26	0,51	25,90	11,04	5,48	0,30
21,60	12,40	46,20	30,00	15,00	8,79	0,49	25,83	6,54	7,23	0,40
22,30	11,60	49,50	30,00	15,00	8,42	0,47	21,89	11,32	0,50	0,03
22,70	11,10	52,80	30,00	15,00	8,28	0,46	26,49	6,09	6,28	0,35
22,30	10,80	56,10	30,00	15,00	8,77	0,49	25,61	10,86	3,31	0,18
20,90	10,10	59,40	30,00	15,00	10,34	0,57	20,28	10,34	0,67	0,04
19,60	9,30	62,70	nan	nan	-	-	19,88	9,63	0,43	0,02
18,20	8,60	66,00	15,00	15,00	7,16	0,40	18,47	9,50	0,94	0,05
16,90	7,80	69,30	15,00	7,50	1,92	0,11	18,66	9,46	2,42	0,13
15,50	7,10	72,60	15,00	7,50	0,64	0,04	18,19	9,15	3,39	0,19
14,20	6,30	75,90	15,00	7,50	1,44	0,08	18,02	9,07	4,72	0,26
12,80	5,60	79,20	15,00	7,50	2,91	0,16	12,37	5,92	0,53	0,03
11,50	4,80	82,50	15,00	7,50	4,42	0,25	11,38	5,56	0,77	0,04
10,10	4,10	85,80	15,00	7,50	5,96	0,33	11,02	5,43	1,62	0,09
9,20	3,60	89,10	15,00	7,50	6,99	0,39	10,68	9,38	5,96	0,33
9,20	3,90	92,40	15,00	7,50	6,83	0,38	10,46	4,91	1,62	0,09
9,50	5,30	95,70	15,00	7,50	5,92	0,33	10,71	5,43	1,21	0,07
9,80	6,70	99,00	10,00	5,00	1,71	0,10	10,46	5,18	1,66	0,09
10,10	8,20	102,30	15,00	7,50	4,95	0,27	10,70	9,76	1,67	0,09
10,10	8,40	105,60	15,00	7,50	4,98	0,28	10,79	9,74	1,50	0,08
9,70	9,10	108,90	15,00	7,50	5,54	0,31	10,32	9,63	0,81	0,05
9,10	10,10	112,20	15,00	15,00	7,67	0,43	10,40	9,94	1,31	0,07
8,80	10,60	115,50	0,00	15,00	9,84	0,55	9,39	10,42	0,62	0,03
7,70	8,60	118,80	nan	nan	-	-	9,74	10,39	2,72	0,15

erro médio (r)

->

0,38

0,17

nan -> significa que não houve estimativa calculada pelo algoritmo

referência efetuadas por uma abordagem *range-based*. O CentroidM não limita o número mínimo de nodos envolvido no processo de cálculo por um único valor limiar da métrica de conectividade, mas sim seleciona sempre 3 nodos âncora para realizar o processo de localização, independentemente de tais nodos terem ou não alcançado o valor limiar estabelecido. Desta forma, o algoritmo sempre entrega uma estimativa de localização. Esta flexibilidade contribui para que o CentroidM calcule estimativas mais precisas que as entregues pelo Centroid. Cada nodo desconhecido que perfaz estes cálculos recebe as mensagens apenas dos nodos de referência e organiza, classifica e utiliza os dados de uma forma muito mais eficiente que no Centroid original.

Da mesma forma como analisado para o caso estático na Seção 7.1, as Figuras 40 e 41 apresentam o mapeamento topológico do erro dos algoritmos de localização, contudo considerando agora o movimento aleatório do nó número 11 da Figura 21. A análise visual das Figura 40 e Figura 41 e dos dados mostrados na Tabela 8 de todo o cenário evidencia que o algoritmo CentroidM manteve baixos valores de erro, ao passo que o Centroid alcançou valores maiores, inclusive em regiões limítrofes onde apresentava um desempenho superior ao CentroidM. Apesar de a topologia da rede considerando a disposição dos nodos âncoras seja a mesma empregada na análise topológica do caso estático, o método Centroid apresentou pontos onde não foi possível realizar estimativas. Os picos indicando o erro unitário na Figura 41 referem-se a coordenadas onde não houve entrega de estimativas. O CentroidM classifica os dados em janelas temporais. Por isto o CentroidM é um algoritmo que não deteriora seus resultados face à mobilidade da rede. Tal comportamento é atribuído à informações que o CentroidM armaneza e utiliza durante o período de amostragem e cálculo das coordenadas:

- tabela com as mensagens (número de mensagens recebidas) ponderadas de cada âncora;
- tabela com as métricas de conectividades gerais de cada âncora;
- tabela com as métricas de conectividades parciais de cada âncora;
- tabela com as métricas de conectividades ponderadas de cada âncora.

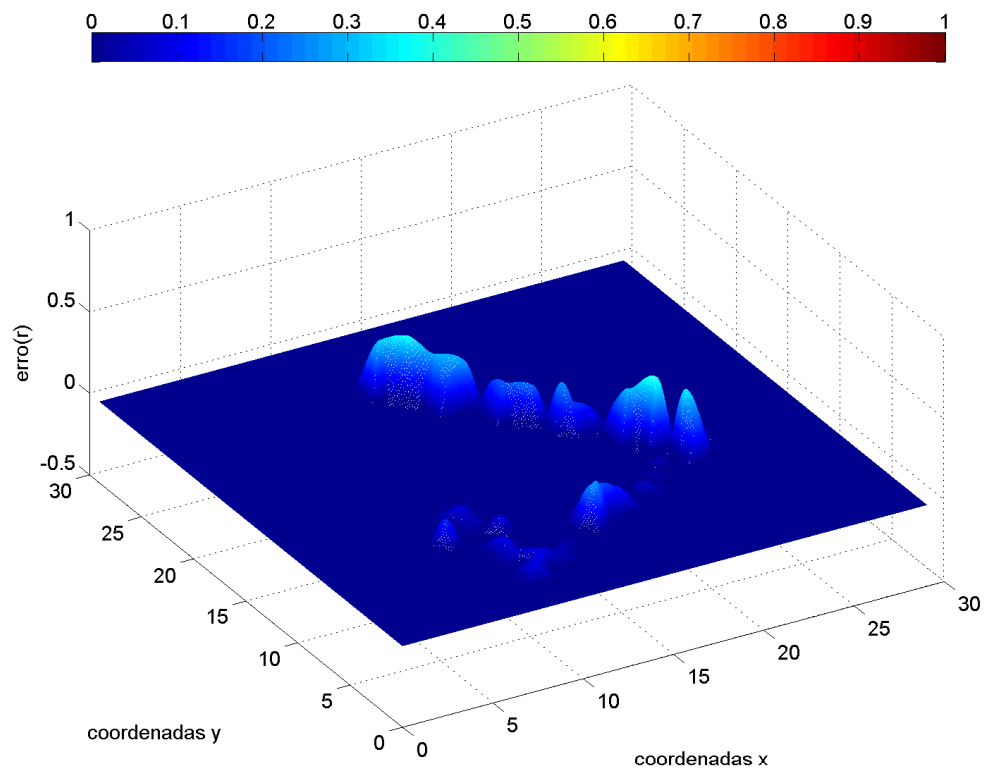


Figura 40: Erro de localização referente ao algoritmo **CentroidM** para o movimento de caminhada aleatória do nó número 11 descrito na topologia da Figura 21.

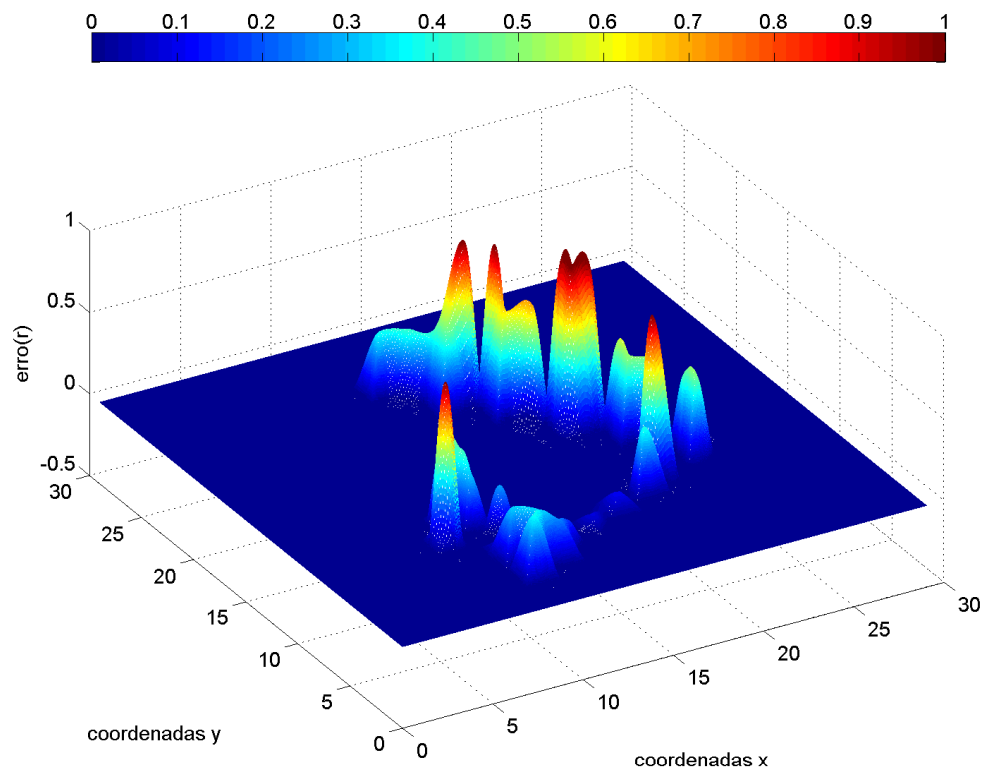


Figura 41: Erro de localização referente ao algoritmo **Centroid** para o movimento de caminhada aleatória do nó número 11 descrito na topologia da Figura 21.

7.3 Equilíbrio do Consumo de Energia: CentroidM versus Centroid

À medida que algumas funcionalidades foram adicionadas ao CentroidM para elevar a precisão dos resultados obtidos, aumentava-se também a quantidade de tempo envolvida no processamento necessário para entregar a estimativa de posição. Este fato resultou em um custo computacional relativamente elevado. As simulações com o Avroa indicaram que o CentroidM ocupa o processador ATMega128L por até 2,76 vezes mais tempo que o Centroid original para realizar um procedimento de localização. De acordo com a Tabela 9, o tempo em que o processador está ocupado consumindo potência em modo ativo é muito maior para o CentroidM. O período indicado leva em consideração o tempo gasto no tratamento inicial das informações recebidas de cada mensagem, durante o período de amostragem t , e o montante necessário para a entrega das coordenadas, ao final deste mesmo período.

Tabela 9: Centroid versus CentroidM - implementação em *software*.

	Tempo	Energia
Centroid	39,25ms	1,05mJ
CentroidM	108,21ms	2,90mJ

Este fato, contudo, pode ser contornado ao se diminuir o número de mensagens enviadas pelos nodos âncora ou recebidas pelos nodos desconhecidos. Por exemplo, naturalmente com uma redução de 70% no número de mensagens recebidas pelos nodos em um ambiente ruidoso, o erro médio de todos os nodos da rede decai muito pouco para o CentroidM, ao passo que a energia gasta face ao Centroid original cai dramaticamente de 2,76 para 1,3 vezes. Para esta situação já não é mais possível comparar resultados entre o Centroid e o CentroidM considerando ambos no mesmo cenário, pois o primeiro não oferece estimativas suficientes para tanto, enquanto que o segundo sempre calcula um par de coordenadas. Ainda assim, o CentroidM é 30% mais lento que o Centroid. Para igualar o consumo de ambas abordagens é necessário buscar uma diminuição a cerca de 11% do número de mensagens recebidas.

O gráfico da Figura 42 mostra o comportamento do erro face à uma diminuição do número de mensagens recebidas pelos nodos para o CentroidM. A simulação do mesmo cenário levou em consideração o aumento gradual de período T entre o envio de dois *beacons* pelos nodos de referência. Embora o aumento do erro tenha sido superior a 60%, ainda assim o valor final ficou cerca de 10% abaixo do erro médio de $0.29r$ calculado para o Centroid, para o cenário simulado.

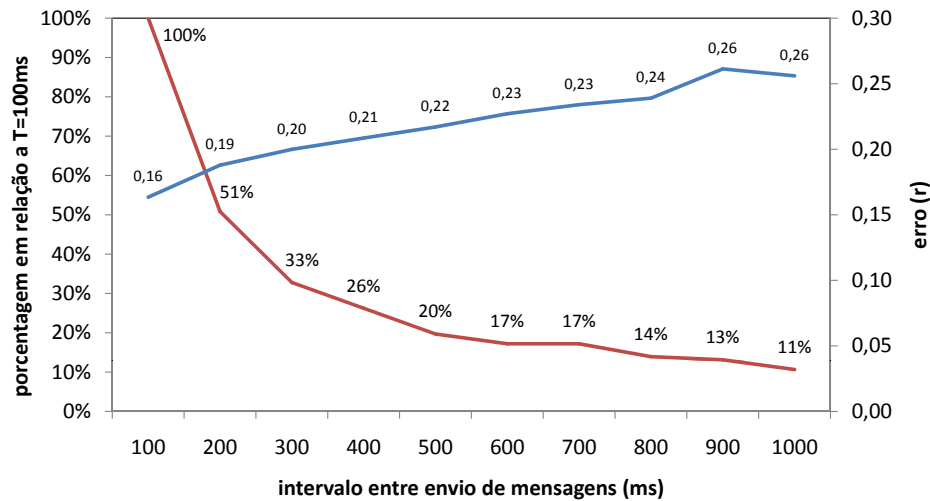


Figura 42: Impacto da redução de mensagens recebidas pelos nodos e aumento do erro para o CentroidM.

7.4 Versão em *Hardware* do CentroidM

A versão em *hardware* do CentroidM foi projetada baseada no modelo descrito em nesC previamente simulado no TOSSIM e Avrora. Este passo foi necessário e permitiu que dois marcos fossem alcançados. Primeiro, foi possível validar funcionalmente o CentroidM no cenário estático e considerando o movimento. Para ambas situações foi provado o ganho em termos de precisão das estimativas frente ao Centroid original. Além das simulações considerando as variações da velocidade para um ambiente com pouco ruído, todos estes cenários também foram submetidos a um ambiente ruidoso. Em segundo lugar, o dado de energia gasta pôde ser inferido através do Avrora a partir da emulação dos algoritmos, dentro de um microcontrolador de baixo consumo utilizado em dispositivos comerciais.

7.4.1 Implementação em Alto Nível

O algoritmo foi descrito utilizando a linguagem VHDL. Posteriormente o circuito foi verificado através de um *testbench* escrito para operar em conjunto com os dados simulados pelo TOSSIM. Desta maneira manteve-se a coerência entre as simulações pelo emprego do mesmo conjunto de vetores de entrada utilizado na simulação em *software*, garantindo a qualidade do processo de verificação funcional e posterior aquisição do consumo de potência. É claro que a construção de um algoritmo de localização para redes de sensores móveis com aumento de precisão nos resultados teve um custo. Apesar de ter melhorado a precisão para ambos cenários estático e móvel, também se elevou o tempo de computação, como já descrito na Seção 7.3. Para transpor esta barreira foi implementada uma solução integrada para o CentroidM. Esta solução, em contrapartida, não substi-

tui outras funcionalidades do microcontrolador. O propósito deste circuito de aplicação específica é de prover a um sistema móvel, o qual seja dependente da informação de localização, um *hardware* dedicado para desempenhar tal função, com o intuito de reduzir energia gasta no processo.

A Figura 43 mostra os sinais de entrada e de saída do circuito dedicado. Destacam-se as portas de saída denominadas por $XestM_w_o$ e $YestM_w_o$. Estas portas representam o resultado em 8 bits para a parte inteira das coordenadas estimadas x e y . Foram destinados 4 bits para a parte fracionária, representadas pelas portas de saída $XestMfrac_w_o$ e $YestMfrac_w_o$. O valor do período de amostragem t representado pela porta de entrada $timer_compute_milli_i$ foi definido em 3,3 segundos e, portanto, as estimativas das coordenadas são apresentadas nesta frequência. A visualização tal como mostrada na Figura 43 foi importante para inspecionar sinais, registradores e estados do circuito na busca por falhas evidenciadas em resultados errados nas portas de saída. Como já mencionado na Seção 6.2, o processo de verificação funcional não incluiu a inserção de *testcases* parciais com o intuito de inspecionar e alertar acerca de falhas do projeto ao longo do fluxo. A verificação foi realizada ao final da execução do *testbench*, através da geração de um arquivo contendo os valores de coordenadas estimados. Posteriormente este arquivo foi comparado com a saída entregue pelo *software* TOSSIM, de forma que esta etapa ocorreu em três momentos distintos do fluxo:

1. inspeção do circuito VHDL projetado;
2. verificação do circuito Verilog mapeado para a tecnologia alvo, com anotações de atraso provenientes da etapa de síntese lógica;
3. verificação pós-leiaute.

Uma vez que a funcionalidade do algoritmo foi verificada com o auxílio do simulador lógico, uma questão que precisa ser respondida é a energia gasta pelo circuito integrado. Existem pelos menos dois fatores que influenciam de maneira substancial a aquisição de resultados de consumo de potência em um fluxo ASIC. O primeiro deles diz respeito ao tipo de estímulo aplicado ao circuito em questão. Optou-se neste trabalho, como já mencionado anteriormente, dar ênfase aos vetores de teste gerados durante a simulação da rede de sensores pelo fato de reproduzirem um caso real de funcionamento. O outro são as condições de operação a que o circuito está submetido quando da análise. Os resultados de consumo levaram em consideração a tensão máxima de 1,32V e frequência igual a 10MHz.

O consumo de potência máximo obviamente não pode exceder à implementação em *software*, contudo é preciso qualificar os vetores de entrada utilizados no processo de aquisição do consumo. O aspecto mais importante no processo de aquisição da atividade de comutação de um circuito está na decisão do comportamento dos vetores de teste. O simulador lógico *nc* (CADENCE, 2009) foi utilizado para obter a atividade de comutação em cada porta do circuito. O algoritmo CentroidM foi submetido ao mesmo conjunto de vetores de teste utilizado na validação em *software*. É importante ressaltar que os estímulos utilizados não foram gerados aleatoriamente, mas sim reproduziram o comportamento das entradas do circuito simuladas de acordo com um programa próprio para redes de sensores. Uma geração randômica neste caso não seria eficiente para inspecionar o circuito em termos funcionais pois o processo de verificação não inclui um modelo específico para ser confrontado com a saída do circuito. Este aspecto é importante no sentido de não introduzir um possível pessimismo desnecessário nesta fase do projeto no tocante à estimativa do consumo, ao mesmo tempo em que mantém a coerência com os dados da etapa anterior responsável pela validação funcional.

7.4.2 Implementação Física do CentroidM

O fluxo de implementação física foi realizado seguindo duas abordagens distintas. Para cada uma delas foi extraído o valor de consumo do circuito durante o período de cálculo das coordenadas. A primeira abordagem levou em consideração a síntese lógica sem a inserção de quaisquer técnicas de redução de consumo, obtendo-se então os resultados reunidos na Tabela 10. Como pode ser visto, o circuito dedicado reduz a energia gasta durante a estimativa da localização em uma ordem de magnitude face ao código executado

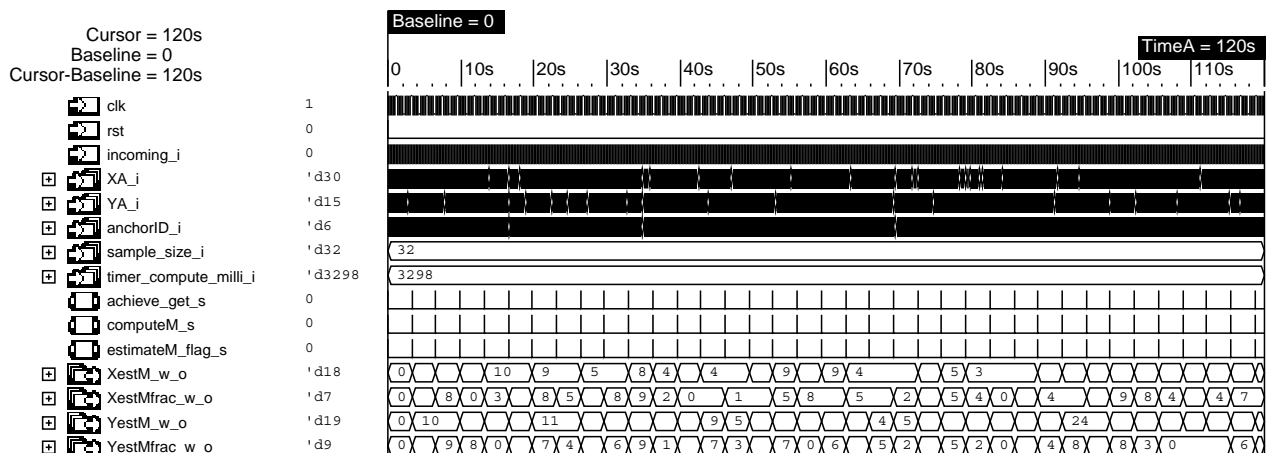


Figura 43: Dados visuais de entrada e saída da implementação em *hardware* do CentroidM para uma simulação de 120 segundos.

no microprocessador. O valor de consumo de potência apresentado na Tabela 10 para a implementação em *hardware* do CentroidM foi entregue pela ferramenta First Encounter de acordo com a atividade de comutação anotada para todos os elementos do circuito, durante os $71,7\mu s$ de duração dos cálculos a uma frequência de $10MHz$, seguindo o procedimento descrito na Seção 6.2.

Tabela 10: Dados para o CentroidM - **primeira** abordagem.

	Tempo de processamento	Potência	Energia
CentroidM (<i>software</i>)	24,19ms	26,79mW	648,13 μ J
CentroidM (<i>hardware</i>)	71,7 μ s	1,45mW	103,99nJ

Embora o valor de $103,99nJ$ apresentado na Tabela 10 seja substancialmente menor que os $648,13\mu J$ da implementação em *software*, este valor pode ainda ser reduzido com o emprego do controle do sinal de *clock*, ou *clock gating*, que segue para os pinos dos registradores. Esta é uma técnica de otimização de potência no nível lógico comumente utilizada em um fluxo de projeto ASIC. Os dados em um registrador tipicamente não são acessados a cada ciclo de *clock* ou não precisam sofrer atualizações a cada ciclo de *clock*. Através da definição de condições que determinem a inatividade dos registradores do circuito, a redução de potência pode ser obtida pelo controle do sinal de *clock* que chega ao pino destes registradores. Para implementar esta funcionalidade, as células de *clock gating* presentes na biblioteca foram utilizadas. Esta simples abordagem reduziu significativamente a atividade de comutação de tais registradores, o que refletiu nos resultados de consumo apresentados na Tabela 11. De fato, o circuito sintetizado para o CentroidM inclui todos registradores necessários para operar o algoritmo e portanto grande parte do circuito é dedicada a elementos sequenciais, o que impacta diretamente na redução apresentada.

Tabela 11: Dados para o CentroidM - **segunda** abordagem.

	Tempo de processamento	Potência	Energia
CentroidM (<i>software</i>)	24,19ms	26,79mW	648,13 μ J
CentroidM (<i>hardware</i>)	71,7 μ s	0,39mW	27,94nJ

As simulações pós-leiaute realizadas com a segunda abordagem resultaram em um valor de consumo igual a $0,39mW$ de potência ativa a uma frequência de $10MHz$. Este valor inclui todo o acesso às informações armazenadas dos nodos âncora, bem como os cálculos para estimar a posição do nó. O leiaute do CentroidM está mostrado na Figura 44 e os resultados para este circuito dedicado estão reunidos na Tabela 12.

O CentroidM foi projetado em sua versão em *hardware* para suportar até 50 nodos

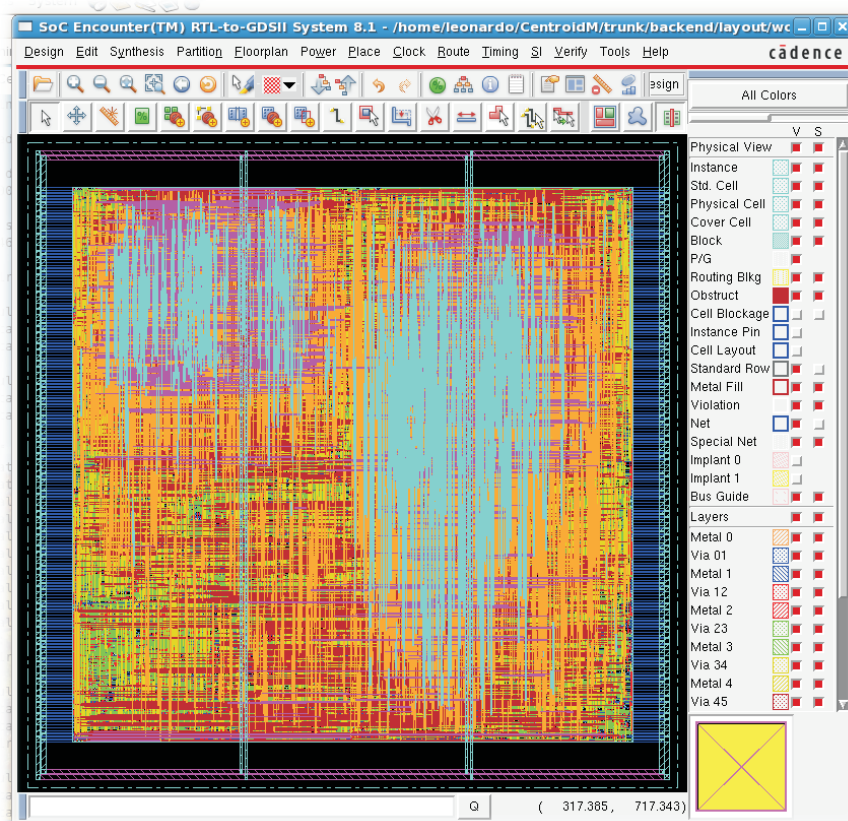


Figura 44: Leiaute do circuito dedicado em tecnologia $0,13\mu$ com uma área estimada em $0,32mm^2$.

âncora distintos ao mesmo tempo. Na realidade esta situação representa o pior caso, isto é, quando um nó viaja por toda a região coberta por nodos de referência, recebendo mensagens de todos estes dispositivos. No entanto, é possível que a rede possa ter mais de 50 nodos âncora, uma vez que, dependendo da velocidade do nó desconhecido, o mesmo não será capaz de ouvir mensagens de 50 nodos distintos dentro do período de amostragem especificado. Além disto, o algoritmo implementado em *hardware* não associa uma determinada identificação com uma posição, isto é, não existe uma tabela pré-estabelecida a qual contenha tais informações. Isto possibilita também que nodos âncora possam ser adicionados ou retirados da rede sem que seja necessário repassar esta informação a quaisquer nodos desconhecidos.

O autor desta Tese teve conhecimento de apenas um trabalho no campo de localização em redes de sensores que seguiu a linha de transpor a um circuito integrado uma aplicação em *software* com o propósito de reduzir o consumo de potência. Em (KARALAR et al., 2004) os autores exploraram a viabilidade acerca da implementação de um sistema de localização para redes de sensores estáticas. De fato não foi apresentado um novo algoritmo de localização ou um melhoramento sobre uma abordagem existente, mas sim analisada a redução em termos de energia gasta pela implementação em *hardware* face a estimativas

Tabela 12: Sumário do sistema de localização integrado para redes de sensores móveis.

parâmetro	valor
tecnologia	0,13 μ
Área	0,32mm ²
Contagem de portas lógicas	85349
Potência em modo ativo	0,39mW
Potência em modo suspenso	0,11mW
<i>Clock</i>	10MHz
Âncoras permitidos	50
Tamanho da palavra	8 bits

considerando a execução do algoritmo selecionado em microprocessadores comerciais. O subsistema de localização apresentado opera em redes estáticas com até 16 nodos âncoras, no entanto não foram apresentados dados de avaliação do posicionamento entregue pelo algoritmo.

Na realidade, não existe uma razão preponderante para construir um sistema de localização integrado para uma rede que opere em um cenário estático. Por outro lado, um nó móvel precisa realizar o processo de localização com uma frequência maior. A utilização massiva de uma rotina de cálculos aliada à necessidade de restringir o consumo de potência, justifica o projeto de um circuito dedicado que implemente esta função em uma rede móvel. Em contrapartida à maior parte da pesquisa na área de algoritmos de localização para redes de sensores móveis, buscou-se nesta Tese, além do aumento da precisão nos resultados entregues, apresentar uma solução em *hardware* com o propósito de reduzir significativamente a energia gasta pelo nó em um processo de localização, uma vez que isto é imperativo para a vida útil da bateria do dispositivo.

7.5 Resumo

Neste Capítulo o desempenho do sistema de localização tanto em sua primeira fase de comprovação da funcionalidade quanto da implementação em HDL do sistema foram explorados. Foi observado que os melhoramentos propostos para o CentroidM penalizaram o seu desempenho em termos de tempo de processamento. Devido à sua simplicidade o Centroid realiza seus cálculos de maneira muito mais veloz que o CentroidM. Resultados de simulação indicaram que o Centroid precisa de 36,27% do tempo utilizado pelo CentroidM para realizar uma estimativa de localização. No início do Capítulo os algoritmos Centroid e CentroidM foram analisados para o caso estático. Observou-se através do levantamento topológico do erro de localização que, apesar do algoritmo Centroid ser mais eficiente em algumas áreas específicas da região sob estudo, a média do erro de todos os nodos

envolvidos na simulação resultava em um valor 33% favorável ao CentroidM. Como pôde ser verificado, apesar do algoritmo CentroidM apresentar uma penalidade em tempo de processamento em relação ao método Centroid original, a abordagem desenvolvida nesta Tese pode apresentar uma redução de cerca de 10% em erro de localização ao igualar a energia gasta no processo. O resultados destas estimativas revelaram que o algoritmo de localização CentroidM projetado em *hardware* tem a sua energia estimada em $27,94nJ$, ao mesmo tempo que mantém o ganho em termos de precisão das estimativas frente ao Centroid original para o caso de um cenário estático e, também considerando os nodos em movimento de maneira aleatória. Para o cenário da rede segundo o modelo de mobilidade *Random Walk*, o erro médio do Centroid ficou em $0,271r$, quanto que para o CentroidM este valor foi de apenas $0,157r$. Isto considerou um cenário de pouco ruído. Na presença de ruído, o Centroid conseguiu entregar apenas 4% do total de estimativas, ao passo que o CentroidM alcançou um erro médio de $0,160r$, ou seja, praticamente manteve o erro. A inserção de ruído no ambiente fez com que os nodos desconhecidos recebessem cerca de 70% menos mensagens que em um ambiente sem ruído. Foi este aspecto que provocou o fato de não haver coordenadas calculadas pelo Centroid.

8 CONCLUSÕES

Esta Tese apresentou o CentroidM, um algoritmo de localização para redes de sensores que foi concebido sob o método Centroid (BULUSU; HEIDEMANN; ESTRIN, 2000). As modificações propostas tornaram o Centroid compatível com redes de sensores móveis. A nova proposta mostrou reduzir de maneira significativa os erros de localização tanto para topologias estáticas quanto para o caso em que os nodos se encontram em movimento. Pelo melhor aproveitamento das informações que o nó recebe dos nodos âncora, o processo de localização foi substancialmente melhorado. Os resultados de simulação do algoritmo proposto, chamado de CentroidM, mostram que ele permite a um nó atingir maior precisão na estimativa de suas coordenadas a um custo reduzido. A redução do erro médio em um cenário estático alcançou o valor de 33% enquanto que em uma rede móvel, onde não há controle acerca do movimento, este número foi superior a 40%. O custo para o CentroidM alcançar tal rendimento resultou no CentroidM ocupando o processador durante 2,75 vezes mais tempo que o Centroid.

Este trabalho focou em uma topologia de rede onde os nodos sensores a serem localizados estão em movimento não controlado. O objetivo foi estimar as coordenadas geográficas dos nodos em movimento apenas com informações de posições conhecidas, enviadas por nodos especializados na rede. Para este propósito, um algoritmo de localização eficiente foi modelado e avaliado em um software para redes de sensores. A primeira parte do trabalho preocupou-se em introduzir um conjunto de algoritmos de localização projetados para redes de sensores sem fio. Os algoritmos foram classificados segundo a maneira pela qual o cálculo das distâncias entre os nodos a serem localizados e entre os nodos de referência era realizado. Depois de uma revisão detalhada de diferentes algoritmos, foi selecionado um método de localização baseado apenas na informação de conectividade para servir de base ao projeto do CentroidM. Este algoritmo inclui características mencionadas ao longo desta Tese que evidenciam a sua robustez, tornando-o favorável para uma implementação em redes de sensores móveis. Neste algoritmo, o nó pode julgar apenas se está dentro do raio de alcance r de outro nó ou não, mas o mesmo não pode obter informações

mais precisas, como por exemplo medir a distância através de um método *range-based*.

A estratégia empregada para alcançar os objetivos elencados para esta Tese envolveu duas abordagens. Em um primeiro momento, reduziu-se o erro em um cenário estático e também considerando o movimento. Contudo, esta etapa acarretou em aumento computacional relevante e estimado em 2,75 vezes. No entanto, tal custo pode ser diluído de duas formas: construção de um circuito dedicado ou redução da precisão. Mostrou-se, desta forma, que esta carga computacional excedente pode ser eliminada ao custo de se reduzir a precisão em localização de um patamar de 40% para 10%. Logo a seguir foi apresentada a estratégia de redução do consumo de potência para o processo de localização através de um circuito dedicado. O projeto e a verificação do algoritmo evidenciara a redução em uma ordem de grandeza¹. Os resultados obtidos mostraram que uma operação de localização gastou o equivalente a $27,94nJ$ a uma frequência de $10MHz$ e tensão de $1,32V$ para um período de $70,7\mu s$.

Para concluir, nesta Tese a tarefa de localização com processamento distribuído em um cenário de movimento não controlado para os nodos foi obtida através de um método puramente *range-free* baseado apenas no recebimento de mensagens, sem necessidade de comunicação inter-nodos. O método proposto mostrou ser mais eficiente tanto em um cenário estático quanto móvel. O processo completo de localização em *hardware* consumiu $0,39mW$ e alcançou uma eficiência energética superior ao dispositivo programável. O CentroidM pode ser desenvolvido para aplicações que não exijam alto grau de precisão dos cálculos da localização do nodo. No tocante ao consumo de potência e mobilidade da rede, o relaxamento do foco sobre a precisão das medidas é aceitável.

Abaixo estão resumidas algumas características da funcionalidade do método CentroidM e sua validação:

- não existe erro acumulado ao longo do tempo;
- localização totalmente baseada na recepção da informação: não há comunicação inter-nodos;
- o algoritmo não necessita de qualquer período de treinamento: alcança o melhor resultado já na primeira estimativa;
- as simulações descritas nesta Tese consideram ambientes com uma perda massiva de pacotes na recepção pelos nodos desconhecidos;

¹1000 vezes.

- não há restrição para o aumento de nodos desconhecidos na rede em relação ao número de nodos âncora. A restrição está atrelada à quantidade suportada pela área onde se encontra a aplicação;
- algoritmo totalmente *range-free*: a "quantização" das distâncias é dada apenas pelos recebimentos de mensagens, alocadas temporalmente;
- mobilidade aleatória dos nodos é considerada: o algoritmo não recebe quaisquer auxílios referentes a sensores de velocidade, direção, tampouco o movimento considera atuadores para alterar a rota com quaisquer propósitos de aumentar a precisão do método.

Tal qual ocorre em muitos projetos, é difícil concluir e esgotar de maneira completa uma tarefa. Mesmo que se tenha atingido as especificações iniciais planejadas para o trabalho, existe ainda um conjunto de possíveis melhoramentos sobre o algoritmo que constituem trabalhos futuros. Dentre tais possibilidades, destacam-se 3:

- o primeiro melhoramento pretendido para o algoritmo é a inclusão do aspecto de mobilidade para os nodos âncora. Até o presente momento o método foi validado para o funcionamento em uma rede de sensores onde os nodos desconhecidos eram móveis e os nodos âncora estáticos. Este aspecto ao mesmo tempo que torna a rede mais flexível, inclui também novos desafios, de tal maneira que a característica de não cooperação entre nodos deva ser relaxada para atingir este objetivo;
- seguindo a linha anterior, é importante testar o algoritmo para modelos de mobilidade de simulem o comportamento de populações ou grupos, de forma a que o nodos de referência móveis possam ser eleitos pela rede de acordo com o local onde estejam;
- por último, um passo que promove o sistema de localização à auto-suficiência é prototipá-lo em um único circuito integrado, formando um *chip* com o sistema digital e as seções RF de comunicação. O projeto digital do algoritmo de localização pode ser facilmente construído, uma vez que o código é parametrizável. Desta forma, o desempenho do circuito a cenários distintos pode ser explorado.

REFERÊNCIAS

- ABBAS, M.; TEKIN, I. Rfid coverage extension using microstrip-patch antenna array [wireless corner]. *IEEE Antennas and Propagation Magazine*, v. 51, n. 1, p. 185–191, 2009.
- ABBAS, A.; JAIN, B. Topology control in heterogeneous mobile ad hoc networks. *IEEE International Conference on Personal Wireless Communications - ICPWC*, p. 47–51, 2005.
- AKYILDIZ, I.; SU, W.; SANKARASUBRAMANIAM, Y.; CAYIRCI, E. A survey on sensor networks. *IEEE Communications Magazine*, v. 40, p. 102–114, 2002.
- ALI, M.; SAIF, U.; DUNKELS, A.; VOIGT, T.; RÖMER, K.; LANGENDOEN, K.; POLASTRE, J.; UZMI, Z. A. Medium access control issues in sensor networks. *ACM SIGCOMM Computer Communication Review*, v. 36, n. 2, p. 33–36, 2006.
- ATMEL. *Atmega128L datasheet*. Jun 2009. http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf.
- BAGGIO, A.; LANGENDOEN, K. Monte-carlo localization for mobile wireless sensor networks. *MSN 2006*, p. 317–328, 2006.
- BERGAMO, P.; MAZZINI, G. Localization in sensor networks with fading and mobility. *13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, v. 2, p. 750–754, 2002.
- BHUVANESWARI, P.; BALAKUMAR, R.; VAIDEHI, V.; BALAMURALIDHAR, P. Solar energy harvesting for wireless sensor networks. *First International Conference on Computational Intelligence, Communication Systems and Networks - CICSYN*, p. 57–61, 2009.
- BLUMENTHAL, J.; REICHENBACH, F.; TIMMERMANN, D. Precise positioning with a low complexity algorithm in ad hoc wireless sensor networks. *Praxis der Informationsverarbeitung und Kommunikation*, v. 28, n. 2, p. 80–85, 2005.
- BOONTRAI, D.; JINGWANGSA, T.; CHERNTANOMWONG, P. Indoor localization technique using passive rfid tags. *9th International Symposium on Communications and Information Technology ISCIT*, p. 922–926, 2009.
- BUCHER, R.; MISRA, D. A synthesizable vhdl model of the exact solution for three-dimensional hyperbolic positioning system. *VLSI Design*, v. 15, n. 2, p. 507–520, 2002.
- BULUSU, N.; HEIDEMANN, J.; ESTRIN, D. Gps-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 2000.

- CADENCE. *Cadence Products*. Jun 2009. <http://www.cadence.com/products/pages/default.aspx>.
- CAFFERY, J.; STUBER, G. J. Subscriber location in cdma cellular networks. *IEEE Transactions on Vehicular Technology*, v. 47, n. 2, p. 406–416, 1998.
- 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.
- CAPKUN, C.; HAMDY, M.; HUBAUX, J.-P. Gps-free positioning in mobile ad-hoc networks. *34th Hawaii International Conference on System Sciences*, 2001.
- CARUSO, A.; CHESSA, S.; DE, S.; URPI, A. Gps free coordinate assignment and routing in wireless sensor networks. *24th Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM*, v. 1, p. 150–160, 2005.
- CERPA, A.; WONG, J. L.; POTKONJAK, M.; ESTRIN, D. Temporal properties of low power wireless links: modeling and implications on multi-hop routing. *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, p. 414–425, 2005.
- CHANDRAKASAN, A. P.; BRODERSEN, R. W. *Low Power Digital CMOS Design*. [S.l.]: Kluwer Academic Publishers, 1995.
- CHEN, H.; HUANG, P.; MARTINS, M.; SO, H. C.; SEZAKI, K. Novel centroid localization algorithm for three-dimensional wireless sensor networks. *4th International Conference on Wireless Communications, Networking and Mobile Computing - WiCOM*, p. 1–4, 2008.
- CHEN, H.; SEZAKI, K.; DENG, P.; SO, H. C. An improved dv-hop localization algorithm for wireless sensor networks. *3rd IEEE Conference on Industrial Electronics and Applications - ICIEA 2008*, p. 1557–1561, 2008.
- CHENG, X.; THAELER, A.; XUE, G.; CHEN, D. Tps: a time-based positioning scheme for outdoor wireless sensor networks. *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2004*, v. 4, p. 2685–2696, 2004.
- CHIPCON. *CC2420 datasheet*. Jun 2009. <http://www.alldatasheet.com/datasheet-pdf/pdf/125399/ETC/CC2420.html>.
- CHUAN, X. Research on improved dv-hop localization algorithm based on weighted least square method. *2008 IEEE International Symposium on Knowledge Acquisition and Modeling Workshop - KAM*, p. 773–776, 2008.
- COHEN, B. *VHDL Coding Styles and Methodologies*. [S.l.]: Springer, 1999.
- COMMUNITY, T. Mar 2009. <http://www.tinyos.net>.
- COSTA, E.; BAMPI, S.; MONTEIRO, J. A new architecture for signed radix-2m pure array multipliers. *IEEE International Conference on Computer Design: VLSI in Computers and Processors*, p. 112–117, 2002.
- COSTA, E. A. C. d. *Operadores Aritméticos de Baixo Consumo para Arquiteturas de Circuitos DSP*. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul, 2002.

- CROSSBOW. *Crossbow site*. Jun 2009. <http://www.xbow.com>.
- CROSSBOW. *Micaz datasheet*. Jun 2009. http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAZ_Datasheet.pdf.
- DATTA, S.; KLINOWSKI, C.; RUDAFSHANI, M.; KHALEQUE, S. Distributed localization in static and mobile sensor networks. *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications - WiMob 2006*, p. 69–76, 2006.
- DELLAERT, F.; FOX, D.; BURGARD, W.; THRUN, S. Monte carlo localization for mobile robots. *IEEE International Conference on Robotics and Automation*, p. 1322–1328, 1999.
- DENG, B.; HUANG, G.; ZHANG, L.; LIU, H. Improved centroid localization algorithm in wsns. *3rd International Conference on Intelligent System and Knowledge Engineering - ISKE 2008*, v. 1, p. 1260–1264, 2008.
- DIL, B.; DULMAN, S.; HAVINGA, P. Range-based localization in mobile sensor networks. *Third European Workshop on Wireless Sensor Networks*, Lecture Notes in Computer Science, n. 3868, p. 164–179, 2006.
- FARADAY. *Faraday Library*. Jun 2009. <http://www.design-reuse.com/news/3008/faraday-free-library-offerings-umc-130nm-0-13-micron-logic-process.html>.
- FORGHANI, Y. A new weighted centroid localization algorithm in wireless sensor networks. *11th International Conference on Computer and Information Technology - ICCIT 2008*, p. 89–93, 2008.
- FREUND, J. E.; SIMON, G. A. *Estatística Aplicada: Economia, Administração e Contabilidade*. [S.l.]: Kluwer Academic Publishers, 2000.
- FRIEDMAN, D.; HEINRICH, H.; DUAN, D.-W. A low-power cmos integrated circuit for field-powered radiofrequency identification tags. *IEEE International Solid-State Circuits Conference - 43rd ISSCC*, p. 294–295, 1997.
- GAURA, E.; NEWMAN, R. Wireless sensor networks: The quest for planetary field sensing. *IEEE Conference on Local Computer Networks*, p. 596–603, 2006.
- GAY, D.; LEVIS, P.; BEHREN, R. V.; WELSH, M.; BREWER, E.; CULLER, D. The nesc language: A holistic approach to networked embedded systems. *Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation*, p. 1–11, 2003.
- GETTING, I. Perspective/navigation-the global positioning system. *IEEE Spectrum*, v. 30, n. 12, p. 36–38, 43–47, 1993.
- GUSTAFSSON, F.; GUNNARSSON, F. *Positioning Using Time-Difference Of Arrival Measurements*. 2003. Disponível em: <citeseer.ist.psu.edu/548251.html>.
- HARTL, G.; LI, B. infer: A bayesian inference approach towards energy efficient data collection in dense sensor networks. *25th IEEE International Conference on Distributed Computing Systems ICDCS*, p. 371–380, 2005.

- HE, T.; HUANG, C.; BLUM, B. M.; STANKOVIC, J. A.; ABDELZAHER, T. Range-free localization schemes for large scale sensor networks. *MobiCom*, 2003.
- HIGHTOWER, J.; BORRIELLO, G. Location systems for ubiquitous computing. *Location Aware Computing*, p. 57–66, 2001.
- HILL, J.; SZEWCZYK, R.; WOO, A.; HOLLAR, S.; CULLER, D.; PISTER, K. System architecture directions for networked sensors. *Architectural Support for Programming Languages and Operating Systems*, p. 93–104, 2000.
- HLA, T.; AUNG, Z. Design approach to fish data identification tag via rfid. *International Conference on Future Computer and Communication ICFCC*, p. 505–509, 2009.
- HSIEH, Y.-L.; WANG, K. Efficient localization in mobile wireless sensor networks. *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, v. 1, p. 292–297, 2006.
- HU, L.; EVANS, D. Localization for mobile sensor networks. *In Tenth Annual International Conference on Mobile Computing and Networking*, 2004.
- HUANG, Q.; SELVAKENNEDY, S. A range-free localization algorithm for wireless sensor networks. *IEEE 63rd Vehicular Technology Conference VTC 2006-Spring*, v. 1, p. 349–353, 2006.
- JI, W.-W.; LIU, Z. An improvement of dv-hop algorithm in wireless sensor networks. *International Conference on Wireless Communications, Networking and Mobile Computing WiCOM*, p. 1–4, 2006.
- JUANG, P.; OKI, H.; WANG, Y. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebrantet. *Tenth International Conference Architectural Support for Programming Languages and Operating Systems*, 2002.
- KANG, D.; NAMGOONG, Y.; YANG, S.; CHOI, S.; SHIN, Y. A simple asynchronous uwb position location algorithm based on single round-trip transmission. *The 8th International Conference Advanced Communication Technology ICACT 2006*, v. 3, p. 4, 2006.
- KAPLAN, E. D. *Understanding GPS : principles and applications*. [S.l.]: Boston, Artech House, 1996.
- KARALAR, T. C.; YAMASHITA, S.; SHEETS, M.; RABAEY, J. An integrated, low power localization system for sensor networks. *First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, 2004.
- 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.
- LADD, A. M.; BEKRIS, K. E.; RUDYS, A.; KAVRAKI, L. E.; WALLACH, D. S. Robotics-based location sensing using wireless Ethernet. *Wireless Networks*, v. 11, n. 1–2, p. 189–204, 2005.
- LAMPART, L. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, v. 21, n. 7, p. 558–565, 1978.

- LANGENDOEN, K. Apples, oranges, and testbeds. *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, p. 387–396, 2006.
- LANGENDOEN, K.; REIJERS, N. Distributed localization in wireless sensor networks: a quantitative comparison. *The International Journal of Computer and Telecommunications Networking*, v. 43, n. 4, p. 499–518, 2003.
- LEE, H.; CERPA, A.; LEVIS, P. Improving wireless simulation through noise modeling. *6th International Symposium on Information Processing in Sensor Networks - IPSN 2007*, p. 21–30, 2007.
- LEE, S.; SONG, Y.; NAM, S. Fast rssi circuit using novel power detector for wireless communication. *International SoC Design Conference - ISOCC*, p. I9–I12, 2008.
- LEVIS, P.; LEE, N.; WELSH, M.; CULLER, D. Tossim: accurate and scalable simulation of entire tinyos applications. *Proceedings of the 1st international conference on Embedded Networked Sensor Systems*, p. 126–137, 2003.
- LI, T.; EKPENYONG, A.; HUANG, Y.-F. A location system using asynchronous distributed sensors. *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2004*, v. 1, p. 628, 2004.
- LIU, C.; WU, K. Performance evaluation of range-free localization methods for wireless sensor networks. *24th IEEE International Performance, Computing, and Communications Conference IPCCC 2005*, p. 59–66, 2005.
- LIU, X.; ZHANG, C.; HU, J. Adaptive weights weighted centroid localization algorithm for wireless sensor networks. *4th International Conference on Wireless Communications, Networking and Mobile Computing - WiCOM 2008*, p. 1–4, 2008.
- LUO, J.; HUBAUX, J.-P. Joint mobility and routing for lifetime elongation in wireless sensor networks. *24th Annual Joint Conference of the IEEE Computer and Communications Societies*, v. 3, p. 1735–1746, 2005.
- MAHESHWARI, R.; JAIN, S.; DAS, S. R. A measurement study of interference modeling and scheduling in low-power wireless networks. *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, p. 141–154, 2008.
- MALPANI, N.; WELCH, J. L.; WAIDYA, N. Leader election algorithms for mobile ad hoc networks. *Proc. Fourth International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, p. 96–103, 2000.
- MENDOZA, M.; SILVA, V.; DIAZ, A. An efficient algorithm for localization in wireless sensor networks based on internal array of nodes within cells. *International Conference Workshops on Parallel Processing, 2005. ICPP 2005 Workshops*, p. 405–412, 2005.
- MILLS, K. L. A brief survey of self-organization in wireless sensor networks: Research articles. *Wireless Communications and Mobile Computing*, v. 7, n. 7, p. 823–834, 2007.
- MONDINELLI, F.; KOVACS-VAJNA, Z. Self-localizing sensor network architectures. *IEEE Transactions on Instrumentation and Measurement*, v. 53, n. 2, p. 277–283, 2004.

- MOZUMDAR, M.; GREGORETTI, F.; LAVAGNO, L.; VANZAGO, L.; OLIVIERI, S. A framework for modeling, simulation and automatic code generation of sensor network application. *5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks -SECON*, p. 515–522, 2008.
- NICULESCU, D. Communication paradigms for sensor networks. *IEEE Communications Magazine*, v. 43, n. 3, p. 116–122, 2005.
- NICULESCU, D.; NATH, B. Ad hoc positioning system (aps). *IEEE Global Telecommunications Conference - GLOBECOM*, v. 5, p. 2926–2931, 2001.
- 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.
- 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. d. *Prototipação e Análise de Circuitos Multiplicadores Array de Baixo Consumo*. Dissertação (Mestrado) — Universidade Federal de Santa Maria, 2005.
- OPENCORES. *One-clock Division Algorithm*. Jan 2009. http://www.opencores.org/project,single_clock_divider.
- 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.
- PARK, S.; AHN, H.-S.; YU, W. A simple object tracking system using itdoa without time synchronization. *The 9th International Conference on Advanced Communication Technology*, v. 3, p. 2026–2028, 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, D.; DIAS, W.; BRAGA, M.; BARRETO, R.; FIGUEIREDO, C.; BRILHANTE, V. Model to integration of rfid into wireless sensor network for tracking and monitoring animals. *11th IEEE International Conference on Computational Science and Engineering*, p. 125–131, 2008.
- PERLA, E.; CATHAIN, A. .; CARBAJO, R. S.; HUGGARD, M.; GOLDRICK, C. M. Powertossim z: realistic energy modelling for wireless sensor network environments. *Proceedings of the 3rd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*, p. 35–42, 2008.
- POLASTRE, J.; SZEWCZYK, R.; CULLER, D. Telos: enabling ultra-low power wireless research. *Fourth International Symposium on Information Processing in Sensor Networks - IPSN*, p. 364–369, 2005.

- RABAEY, J.; AMMER, M.; SILVA J.L., J. da; PATEL, D.; ROUNDY, S. Picoradio supports ad hoc ultra-low power wireless networking. *Computer*, v. 33, n. 7, p. 42–48, 2000.
- REICHENBACH, F.; BLUMENTHAL, J.; TIMMERMANN, D. Improved precision of coarse grained localization in wireless sensor networks. *9th EUROMICRO Conference on Digital System Design: Architectures, Methods and Tools - DSD 2006*, p. 630–640, 2006.
- REN, H.; MENG, M. Q.-H.; CHEN, X. Investigating network optimization approaches in wireless sensor networks. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, p. 2015–2021, 2006.
- RISTIC, B.; MORELANDE, M.; FARINA, A.; DULMAN, S. On proximity-based range-free node localisation in wireless sensor networks. *9th International Conference on Information Fusion ICIF'06*, p. 1–8, 2006.
- RONG, P.; SICHITIU, M. Angle of arrival localization for wireless sensor networks. *3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks - SECON'06*, v. 1, p. 374–382, 2006.
- RUBENSTEIN, D. I. The ecology of female social behavior in horses, zebras, and asses. *Animal Societies: Individuals, Interactions, and Organization.*, p. 13–18, 1994.
- SAVARESE, C.; RABAEY, J.; BEUTEL, J. Location in distributed ad-hoc wireless sensor networks. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, v. 4, p. 2037–2040, 2001.
- SAVARESE, C.; RABAEY, J.; LANGENDOEN, K. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. *USENIX Technical Annual Conference*, 2002.
- SAVVIDES, A.; HAN, C.-C.; STRIVASTAVA, M. B. Dynamic fine-grained localization in ad-hoc networks of sensors. *Proceedings of the 7th annual international conference on Mobile computing and networking*, p. 166–179, 2001.
- SAVVIDES, A.; PARK, H.; SRIVASTAVA, M. B. The bits and flops of the n-hop multilateration primitive for node localization problems. *Proceedings of the First ACM International Workshop on Sensor Networks and Applications*, p. 112–121, 2002.
- SCHAU, H.; ROBINSON, A. Passive source localization employing intersecting spherical surfaces from time-of-arrival differences. *IEEE Transactions on Acoustics, Speech, and Signal Processing [IEEE Transactions on Signal Processing]*, v. 35, n. 8, p. 1223–1225, 1987.
- SHAMELI, A.; SAFARIAN, A.; ROFOUGARAN, A.; ROFOUGARAN, M.; CASTANEDA, J.; FLAVIIS, F. D. A uhf near-field rfid system with fully integrated transponder. *IEEE Transactions on Microwave Theory and Techniques*, v. 56, n. 5, p. 1267–1277, 2008.
- SHANG, Y.; RUML, W. Improved mds-based localization. *Conference of the IEEE Communications Society*, 2004.

- SHENOY, S.; TAN, J. Simultaneous localization and mobile robot navigation in a hybrid sensor network. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, p. 1636–1641, 2005.
- SHNAYDER, V.; HEMPSTEAD, M.; CHEN, B. rong; ALLEN, G. W.; WELSH, M. Simulating the power consumption of large-scale sensor network applications. *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, p. 188–200, 2004.
- SICHITIU, M.; RAMADURAI, V. Localization of wireless sensor networks with a mobile beacon. *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, p. 174–183, 2004.
- 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.
- SREENATH, K.; LEWIS, F.; POPA, D. Localization of a wireless sensor network with unattended ground sensors and some mobile robots. *IEEE Conference on Robotics, Automation and Mechatronics*, p. 1–8, 2006.
- SSU, K.-F.; OU, C.-H.; JIAU, H. Localization with mobile anchor points in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, v. 54, n. 3, p. 1187–1197, 2005.
- STARNER, T. Human-powered wearable computing. *IBM Syst. J.*, IBM Corp., Riverton, NJ, USA, v. 35, n. 3-4, p. 618–629, 1996. ISSN 0018-8670.
- STEVENS-NAVARRO, E.; VIVEKANANDAN, V.; WONG, V. W. Dual and mixture monte carlo localization algorithms for mobile wireless sensor networks. *IEEE Wireless Communications and Networking Conference - WCNC*, p. 4024–4028, 2007.
- SUN, Z.; YU, R.; MEI, S. A power-aware and range-free localization algorithm for sensor networks. *Asia-Pacific Conference on Communications APCC'06*, p. 1–5, 2006.
- TAY, J.; CHANDRASEKHAR, V.; SEAH, W. Range-free localization using dynamic hop size computation in wireless sensor networks. *IEEE International Conference on Industrial Informatics*, p. 126–131, 2006.
- THRUN, S.; FOX, D.; BURGARD, W.; DELLAERT, F. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, v. 128, n. 1-2, p. 99–141, 2001.
- TIAN, S.; ZHANG, X.; WANG, X.; SUN, P.; ZHANG, H. A selective anchor node localization algorithm for wireless sensor networks. *International Conference on Convergence Information Technology, 2007*, p. 358–362, 2007.
- TILAK, S.; KOLAR, V.; ABU-GHAZALEH, N.; KANG, K.-D. Dynamic localization control for mobile sensor networks. *24th IEEE International Performance, Computing, and Communications Conference*, p. 587–592, 2005.
- TITZER, B. L.; LEE, D. K.; PALSBERG, J. Aurora: scalable sensor network simulation with precise timing. *Proceedings of the 4th international symposium on Information processing in sensor networks*, 2005.

TOLLE, G.; POLASTRE, J.; SZEWCZYK, R.; CULLER, D.; TURNER, N.; TU, K.; BURGESS, S.; DAWSON, T.; BUONADONNA, P.; GAY, D.; HONG, W. A macroscope in the redwoods. *ACM*, New York, NY, USA, p. 51–63, 2005.

UFSM. *Estrutura e apresentacao de monografias, dissertacoes e teses: MDT, Universidade Federal de Santa Maria*. 6 ed.. ed. [S.l.]: Editora UFSM, 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.

WANG, C.; LIU, K.; XIAO, N. A range free localization algorithm based on restricted-area for wireless sensor networks. *The Third International Multi-Conference on Computing in the Global Information Technology - ICCGI 2008*, p. 97–101, 2008.

WANG, J.; ZHANG, C.; CHI, B.; WANG, Z.; WANG, Z. A fully integrated cmos uhf rfid reader transceiver for handheld applications. *IEEE Custom Integrated Circuits Conference CICC*, p. 495–498, 2009.

WERNER-ALLEN, G.; JOHNSON, J.; RUIZ, M.; LEES, J.; WELSH, M. Monitoring volcanic eruptions with a wireless sensor network. *Proceedings of the Second European Workshop on Wireless Sensor Networks*, p. 108–120, 2005.

WILLIAMS, C.; YATES, R. Analysis of a micro-electric generator for microsystems. *International Conference on Solid-State Sensors and Actuators*, p. 369–372, 1995.

WU, C.; SHENG, W.; SONG, W.-Z. A dynamic mds-based localization algorithm for mobile sensor networks. *International Conference on Robotics and Biomimetics*, p. 496–501, 2006.

WU, K.; LIU, C.; KING, V. Very low cost sensor localization for hostile environments. *IEEE International Conference on Communications ICC 2005*, v. 5, p. 3197–3201, 2005.

XIAO, J.; REN, L.; TAN, J. Research of tdoa based self-localization approach in wireless sensor network. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, p. 2035–2040, 2006.

XIAO, L.; JIN-KUAN, W.; YUN, W. A novel localization algorithm based on received signal strength for mobile wireless sensor networks. *International Conference on Microwave and Millimeter Wave Technology - ICMMT 2008*, v. 1, p. 92–95, 2008.

XU, X.; HUANG, L.; LI, X.; WANG, J.; XU, H. A fine-grained localization algorithm in wireless sensor networks. *IEEE Wireless Communications and Networking Conference - WCNC 2009*, p. 1–6, 2009.

YU, G.; YU, F.; FENG, L. A three dimensional localization algorithm using a mobile anchor node under wireless channel. *IEEE International Joint Conference on Neural Networks - IJCNN 2008*, p. 477–483, 2008.

YUAN, L.; CHEN, W.; XI, Y. A review of control and localization for mobile sensor networks. *The Sixth World Congress on Intelligent Control and Automation*, v. 2, p. 9164–9168, 2006.

ZHANG, B.; YU, F. An energy efficient localization algorithm for wireless sensor networks using a mobile anchor node. *International Conference on Information and Automation - ICIA 2008*, p. 215–219, 2008.

ZHANG, Y.; LIU, W.; FANG, Y.; WU, D. Secure localization and authentication in ultra-wideband sensor networks. *IEEE Journal on Selected Areas in Communications*, v. 24, n. 4, p. 829–835, 2006.

ZHAO, S.; SUN, M.; ZHANG, Z. Gasa-hop localization algorithm for wireless sensor networks. v. 2, p. 152–156, 2009.