



Universidade Federal de Pernambuco  
Centro de Informática

Pós-graduação em Ciência da Computação

**PROTOCOLOS ASSÍNCRONOS PARA  
TREINAMENTO DE LOCALIZAÇÃO EM  
REDES DE SENSORES CENTRADAS EM  
ATOR**

José Souza de Jesus

DISSERTAÇÃO DE MESTRADO

Recife

20 de abril de 2012

Universidade Federal de Pernambuco  
Centro de Informática

José Souza de Jesus

**PROTOCOLOS ASSÍNCRONOS PARA TREINAMENTO DE  
LOCALIZAÇÃO EM REDES DE SENSORES CENTRADAS EM  
ATOR**

*Trabalho apresentado ao Programa de Pós-graduação em  
Ciência da Computação do Centro de Informática da Uni-  
versidade Federal de Pernambuco como requisito parcial  
para obtenção do grau de Mestre em Ciência da Com-  
putação.*

Orientador: *Paulo André da Silva Gonçalves*

Recife

20 de abril de 2012

*A cada um que deixei esperando, apreensiva e esperançosamente. Pai, mãe, irmãos, amigos.*

## AGRADECIMENTOS

Primeiramente, a Deus pelo sopro de vida e inspirações de sabedoria, paciência e determinação, dons naturais a qualquer ser humano. Aos meus pais, José de Jesus e Elisabete Souza de Jesus, que me ensinaram a andar de maneira livre e independente, mas nunca sozinho. A meus irmãos, Jaqueline, Ivanildo (*in memorian*), Joaci (*in memorian*), Jomar, Josemar e Adriana, companheiros do dia a dia. E a todos meus familiares que me acompanharam por recados, ligações, orações e pensamentos.

A meus grandes amigos cujo incentivo me fez chegar até aqui e me fará ir mais longe. Felizmente, a lista é grande e, infelizmente, será abreviada: meus inestimáveis amigos Maurício, Fred, Pablo, João, Paulo, Tércio, Roberto e os demais companheiros nos raros momentos de ócio; Ana, Cláudio, Adna, Anderson, Davison, Jocélia, Tainã e meus demais irmãos de caminhada; e é claro Carol e Márcia, pessoas que me apoiaram desde o início desta investida. Aos moradores de Recife que me acolheram tão bem, em especial, Sra. Josefa e Sr. Paulo. A todos os moradores dessa bela cidade, sobretudo aqueles que aqui conheci e me fizeram agradável companhia, principalmente Vanessa Ferraz e Mônica Paula.

A todos os gigantes que vieram antes de mim, cujo esforços em prol da ciência trouxeram grandes inovações e novas possibilidades. Ao Departamento de Ciência da Computação da Universidade Federal da Bahia que me forneceu formação indispensável e muito bem aproveitada durante mais esta etapa. A Universidade Federal de Pernambuco e todo seu corpo docente e de funcionários. Entre esses, enfaticamente, a Paulo André Gonçalves que cumpriu bem seu papel de orientador, contribuindo não só para este trabalho, mas para minha formação profissional. A todos os colegas que também contribuíram com minha formação. Aos companheiros de pesquisa, piadas, sugestões e reclamações, Pedro, Felipe, Luca, Kadna, entre outros. A todos meus colegas do grupo de pesquisa em redes de computadores. À CAPES (Coordenação de Aperfeiçoamento de Pessoal de

Nível Superior) pelo auxílio prestado a mim e a muitos outros estudantes/pesquisadores dedicados.

Aos professores convidados para compor a banca de avaliação deste trabalho, doutores Nazareno Andrade (Universidade Federal Campina Grande) e José Suruagy (Universidade Federal Pernambuco), pelas críticas e sugestões que contribuíram positivamente com esta pesquisa.

*Veni, vidi, vici*

—JÚLIO CESAR

## RESUMO

Este trabalho foca o treinamento de localização em redes de sensores centradas em ator. Nesse ambiente, o ator é capaz de organizar sensores nas proximidades e formar uma rede de curta duração para fins específicos. É considerada uma rede heterogênea formada por sensores dotados de ciclos de trabalho distintos. Nessas redes, os sensores se localizam em relação ao ator através do uso de protocolos assíncronos de treinamento. Na literatura, existem dois protocolos propostos para isso: *Flat+* e *Binary Training*. Entretanto, tais protocolos foram propostos sob a condição de um canal de comunicação ideal, fato improvável em sistemas reais. Este trabalho analisa o desempenho dos protocolos *Flat+* e *Binary Training* em canais com desvanecimento e constata que imperfeições no canal de comunicação levam a problemas de convergência nos algoritmos e a erros de localização. Este trabalho também propõe um protocolo assíncrono de treinamento de localização que lida melhor com imperfeições no canal. As avaliações de desempenho do protocolo proposto mostram que o mesmo não apresenta problemas de convergência e possui maior acurácia de localização se comparado aos demais protocolos analisados.

**Palavras-chave:** redes de sensores e atores, protocolo de localização, desvanecimento, análise de desempenho

## ABSTRACT

This work focuses on location training in wireless actor-centric sensor networks. In this scenario, the actor can organize sensors nearby and settle short-lived networks with well defined objectives. We consider a heterogeneous network formed by sensors with distinct dutty-cicle schemes. In such network, the sensors use asynchronous training protocols to locate themselves taking the actor as a reference. There are two protocols proposed with such objective: Flat+ and Binary Training. However, those protocols were proposed under assumptions of ideal communication channel, fact impossible in real systems. This work analyses the performance of Flat+ and Binary Training under fading channels and concludes that imperfect communication channels cause convergence problems and localization errors. This work also proposes a novel asynchronous localization protocol that overcome channel imperfections. Our performance evaluation shows that the proposed protocol does not have convergence problems and improve localization accuracy when compared to other protocols.

**Keywords:** sensor and actor network, localization protocols, fading channel, performance evaluation



# SUMÁRIO

<b>Capítulo 1—Introdução</b>	1
1.1 Motivação . . . . .	1
1.2 Objetivos . . . . .	3
1.3 Organização . . . . .	3
<b>Capítulo 2—Treinamento de Localização em Redes de Sensores e Atores</b>	5
2.1 Redes de Sensores e Atores . . . . .	5
2.2 Localização em Redes de Sensores . . . . .	7
2.3 Modelo de Rede . . . . .	10
Resumo . . . . .	12
<b>Capítulo 3—Trabalhos Relacionados</b>	14
3.1 O Problema de Localização em Redes de Sensores . . . . .	14
3.2 Protocolos <i>Range-based</i> . . . . .	16
3.3 Protocolos <i>Range-Free</i> . . . . .	17
3.4 Localização em Redes de Sensores e Atores . . . . .	18
Resumo . . . . .	19
<b>Capítulo 4—Protocolos Assíncronos de Treinamento</b>	20
4.1 O Protocolo <i>Flat+</i> . . . . .	20
4.2 O Protocolo <i>Binary Training</i> . . . . .	22
Resumo . . . . .	25

<b>Capítulo 5—Análise Experimental dos Protocolos Flat+ e Binary Training</b>	<b>27</b>
5.1 Modelo de Simulação . . . . .	27
5.1.1 Descrição do Simulador . . . . .	27
5.1.2 Modelagem do Canal de Comunicação . . . . .	28
5.1.3 Descrição dos Experimentos . . . . .	29
5.1.3.1 Parâmetros do Modelo . . . . .	30
5.1.3.2 Descrição dos Cenários . . . . .	30
5.1.3.3 Métricas Utilizadas . . . . .	31
5.2 Desempenho dos Protocolos . . . . .	32
5.2.1 Justificativas para os Erros Encontrados . . . . .	34
Resumo . . . . .	35
<b>Capítulo 6—Protocolo Strong Flat</b>	<b>37</b>
6.1 Descrição do algoritmo . . . . .	37
6.2 Análise Teórica de Desempenho . . . . .	40
6.2.1 Sensor Periódico . . . . .	40
6.2.2 Sensor Livre . . . . .	41
Resumo . . . . .	43
<b>Capítulo 7—Análise Comparativa dos Protocolos</b>	<b>44</b>
7.1 Análise de Desempenho do Protocolo <i>Strong Flat</i> . . . . .	44
7.1.1 Cenário com $\beta_{hit} > 1$ . . . . .	47
7.2 Comparação Entre os Protocolos Estudados . . . . .	49
7.2.1 Análise Teórica de Desempenho . . . . .	49
7.2.2 Avaliações Experimentais . . . . .	50
7.2.3 Análise de Complexidade . . . . .	52
Resumo . . . . .	53
<b>Capítulo 8—Conclusões</b>	<b>54</b>

SUMÁRIO

xi

**Apêndice A—Cálculo da Quantidade Média de Testes**

57

## LISTA DE FIGURAS

2.1	Multilateração . . . . .	8
2.2	Algoritmo Centroide. . . . .	9
2.3	RSCA dividida em 5 coroas e 10 setores. . . . .	11
2.4	RSCA dividida em 5 coroas. . . . .	12
2.5	Tipos de sensores na rede. . . . .	12
4.1	Comportamento do ator <i>Flat+</i> . . . . .	20
4.2	Comportamento do ator <i>Binary Training</i> . . . . .	23
5.1	Efeito do desvanecimento . . . . .	29
5.2	Convergência e precisão dos protocolos. . . . .	33
5.3	Erros causados pela perda de <i>beacons</i> . . . . .	35
6.1	Pior caso <i>Strong Flat</i> periódico . . . . .	41
6.2	Pior caso <i>Strong Flat</i> livre . . . . .	42
7.1	Efeito dos valores de $\beta_{miss}$ . . . . .	45
7.2	Análise do tempo em escuta do <i>Strong Flat Periodic</i> . . . . .	46
7.3	Efeito dos valores de $\beta_{miss}$ sobre o índice de treinamento. . . . .	46
7.4	Efeito dos valores de $\beta_{miss}$ sobre o índice de convergência. . . . .	47
7.5	Comparação entre <i>Flat+</i> e <i>Strong Flat</i> (1, 1) para sensores periódicos. . . . .	47
7.6	Efeito de $\beta_{hit}$ e $\beta_{miss}$ , com resíduo. . . . .	48
7.7	Efeito de $\beta_{hit}$ e $\beta_{miss}$ , sem resíduo. . . . .	49
7.8	Precisão dos protocolos. . . . .	51
7.9	Desempenho dos protocolos. . . . .	52

## LISTA DE TABELAS

5.1	Valores do expoente de atenuação $\alpha$ . . . . .	30
5.2	Valores dos parâmetros de simulação. . . . .	31
7.1	Comparação entre os protocolos estudados. . . . .	50
7.2	Complexidade dos algoritmos estudados. . . . .	52

## LISTA DE SÍMBOLOS

- $C_i$  I-ésima coroa mais externa. 11
- $L_n$  Quantidade de tempo, em *slots*, que caracteriza um ciclo de trabalho (escuta e inatividade) de um sensor. 11
- $b_i$  *Beacon* que identifica a coroa  $C_i$ . 11
- $b_{guess}$  *Beacon* específico que o sensor deseja escutar. 23
- $d_n$  Quantidade de tempo, em *slots*, no qual o sensor escuta o canal de comunicação. 11
- $k$  Quantidade de coroas. 11
- $p_i$  Potência de transmissão cujo alcance é a coroa  $C_i$ . 11
- $D$  Distância entre o transmissor e o receptor. 28
- $D_0$  Distância de referência. 28
- $P_L(D_0)$  Perda de sinal a uma distância de referência  $D_0$ . 28
- $P_r$  Potência recebida. 28
- $P_t$  Potência de transmissão. 28
- $P_{suc}$  Probabilidade de recepção com sucesso. 29
- $Q(x)$  Função-Q ou função acumulada da distribuição normal de probabilidade. 29
- $X_\sigma$  Variável aleatória de distribuição normal com média em zero e desvio padrão  $\sigma$ . 29
- $\alpha$  Expoente de atenuação. 28

- $\gamma$  Potência mínima necessária para decodificar um sinal. 29
- $\overline{P_r}$  Potência média recebida. 29
- $\sigma$  Desvio que representa as variações na força do sinal recebido. 29
- $IC$  Índice de Convergência. 32
- $I_\Delta$  Índice de treinamento em relação ao erro obtido  $\Delta$ . 32
- $\Delta$  Erro de estimativa, ou seja, a diferença entre a coroa estimada pelo sensor e a coroa real na qual o mesmo está posicionado. 32
- $\omega$  Quantidade total de *slots* nos quais o sensor escutou o canal de comunicação. 22
- $\psi$  Quantidade de iterações decorridas do início até o fim do treinamento. 22
- $\tau$  Tempo, em slots, decorrido do início até o fim do treinamento. 22
- $H_k[\ ]$  Vetor que armazena quantas vezes um *beacon*  $b_i$  foi escutado, no protocolo *Strong Flat+*. 37
- $M_k[\ ]$  Vetor que armazena quantas vezes um *beacon*  $b_i$  não foi escutado, no protocolo *Strong Flat+*. 37
- $R_k[\ ]$  Vetor que armazena informações sobre o recebimento do *beacon*  $b_i$  no protocolo *Flat+*. 20

## GLOSSÁRIO

- AoA** *Angle of Arrival*, ângulo de chegada. 7
- BER** *Bit Error Rate*, taxa de erro de *bits*. 30
- GPS** *Global Positioning System*, sistema de posicionamento global. 11
- KNN** *K-Nearest Neighbors*, *k* vizinhos mais próximos. 15
- NLoS** *Non-Line of Sight*, Comunicação com visada indireta. 14
- OQPSK** *Offset Quadrature Phase-Shift Keying*. 30
- PDA** *Personal Digital Assistant*, assistente digital pessoal. 6
- PER** *Packet Error Rate*, taxa de erro de pacotes. 30, 33
- RSCA** Rede de Sensores Centrada em Ator. 1, 10, 13, 18–20, 26, 43, 54
- RSSF** Rede de Sensores sem Fio. 1, 2, 5, 7, 14, 19
- RSSI** *Received Signal Strength Indicator*, indicador de força de sinal recebido. 7, 8, 14–16
- TDoA** *Time Difference of Arrival*, diferença entre tempos de chegada. 7, 14
- ToA** *Time of Arrival*, tempo de chegada. 7, 14



# INTRODUÇÃO

Sem dúvidas, uma das tecnologias mais favorecidas com a miniaturização de componentes eletrônicos e transmissão em baixa potência é a tecnologia das redes de sensores sem fio. Ainda hoje, essa tecnologia é considerada emergente e acessível a uma gama de aplicações industriais, domésticas, urbanas e militares. Há décadas vem sendo estudado o potencial dessa tecnologia, assim como seus desafios ainda existentes. Atualmente, pode ser dito que a pesquisa em redes de sensores atingiu maturidade. Mesmo assim, existem lacunas abertas e problemas não resolvidos, possibilitando várias linhas de pesquisa na área.

## 1.1 MOTIVAÇÃO

Redes de sensores sem fio (RSSF) são conhecidas por empregar grande número de dispositivos sensores em dada área, onde esses vão interagir entre si e formar uma rede sem infraestrutura fixa e, em geral, desconsiderando intervenção humana durante o funcionamento. RSSF podem ser aplicadas em diversas situações, como em aplicações de monitoramento e controle de propriedades do ambiente [Yick et al. 2008].

Num tipo particular de RSSF, um dispositivo mais robusto, chamado ator, é capaz de coordenar os sensores ao seu redor, assim como coletar suas informações e disponibilizá-las ao usuário [Akyildiz and Kasimoglu 2004]. Ainda nesse aspecto, sensores próximos a um ator formam uma sub-rede temporária, com o ator ao centro. Este ator é capaz de organizar os sensores à sua volta para que estes trabalhem de forma cooperativa para atingir um objetivo específico [Navarra and Pinotti 2011]. RSSF com essas características são chamadas de redes de sensores centradas em ator (RSCA).

Boa parte das aplicações necessitam de informações sobre a localização dos dispositivos para o correto tratamento dos eventos detectados. O problema da localização, do inglês *location awareness*, consiste em permitir aos dispositivos estimar a sua posição em relação

a um ponto de referência [Mao et al. 2007b, Yick et al. 2008]. Embora tecnologias como GPS (*Global Positioning System*) supram essa necessidade, nem sempre é possível utilizá-las por demandarem *hardware* especializado e condições geográficas favoráveis que possibilitem a recepção de sinal via satélite [Patwari et al. 2005, Mao et al. 2007b]. Logo, são necessárias soluções específicas para proporcionar localização em RSSF.

Na literatura foram propostos diversos protocolos de localização [Mao et al. 2007b, Yick et al. 2008]. Em geral, essas propostas utilizam técnicas para estimativa de distância, uso de nós âncora com posição conhecida e algoritmos de refinamento. Naturalmente, cada técnica utilizada nos protocolos de localização possui seus pontos positivos e suas deficiências. Por exemplo, estimativas de distância agregam erro à posição estimada pelo sensor, enquanto o uso de âncoras depende da distribuição dos dispositivos principalmente em redes altamente populadas.

Recentemente vem sendo explorado o conceito de treinamento de localização em RSCA, onde os nós sensores podem estimar sua posição apenas através da escuta de *beacons* transmitidos pelo ator [Navarra et al. 2010]. Treinamento de localização é uma técnica atraente por ser simples e bastante escalável mesmo para redes em larga escala. Apenas escutando *beacons* transmitidos pelo ator, o sensor pode estimar segundo um sistema de referência polar com o ator ao centro. Esse tipo de organização também favorece uma série de aplicações, por exemplo, aplicações que executem ações localizadas no ambiente (como controle de temperatura), organização de sensores em *clusters* baseados em posição geográfica, entre outros [Wadaa et al. 2005, Navarra and Pinotti 2011].

Em especial, dois protocolos de treinamento propostos na literatura possibilitam o treinamento de sensores em redes heterogêneas de larga escala, de maneira assíncrona: os protocolos *Flat+* [Barsi et al. 2009] e *Binary Training* [Barsi et al. 2011]. Através de análise de desempenho, foi constatada a eficiência desses protocolos em termos de tempo e consumo energético [Barsi et al. 2011]. Entretanto, os protocolos propostos consideram um canal de comunicação ideal, onde não há perda de dados. Em aplicações reais, tal condição não pode ser garantida. Desta forma, é necessário avaliar o comportamento desses protocolos em ambientes mais realistas e propor melhorias, de modo a tornar o treinamento possível mesmo sob comunicação com perda de pacotes.

## 1.2 OBJETIVOS

Como principal objetivo, este trabalho propõe um protocolo de treinamento de localização em RSCA mais robusto, cujo funcionamento é garantido mesmo considerando canais de comunicação imperfeitos com perda de pacotes. Essa solução mais robusta viabiliza a aplicabilidade dos protocolos de treinamento em cenários reais, visto que as soluções existentes não o faz devido a suas considerações sob condições ideais de transmissão. Adicionalmente, podem ser listados os seguintes objetivos específicos:

- Estudo dos protocolos existentes com análise de parâmetros relacionados à aplicabilidade em situações reais, tais como modelo de rede, características do transceptor, características do ambiente, entre outros;
- Avaliação de desempenho dos protocolos existentes, sobretudo considerando canais de comunicação com perdas através da modelagem de canais com desvanecimento, com identificação das deficiências dos protocolos analisados;
- Proposta de um novo protocolo resistente à perda de pacotes e de desempenho superior aos demais protocolos estudados em cenários mais realistas;
- Avaliação de desempenho do protocolo proposto, inclusive através da comparação com os demais protocolos estudados.

## 1.3 ORGANIZAÇÃO

O conteúdo deste documento está organizado como segue: O Capítulo 2 apresenta conceitos necessários para a compreensão deste trabalho. O mesmo capítulo define o problema do treinamento de localização e descreve o modelo de rede considerado. Em seguida, o Capítulo 3 apresenta o estado da arte e trabalhos relacionados. O Capítulo 4 descreve os dois protocolos estudados neste trabalho: *Flat+* e *Binary Training*. O Capítulo 5 apresenta uma análise de desempenho dos protocolos estudados. Para isso, o capítulo define o modelo de propagação, determina os parâmetros de simulação e descreve o cenário considerado. No Capítulo 6 é proposto um protocolo para treinamento de localização e são detalhadas algumas características de funcionamento. No Capítulo 7 é realizada uma

análise de desempenho do protocolo proposto, focando numa comparação envolvendo os outros dois protocolos estudados. Por fim, no Capítulo 8 são apresentadas as conclusões obtidas por este trabalho assim como propostas para trabalhos futuros.

# TREINAMENTO DE LOCALIZAÇÃO EM REDES DE SENSORES E ATORES

Este capítulo faz uma contextualização sobre treinamento de localização em redes de sensores centradas em ator. São definidos conceitos necessários para o desenvolvimento e compreensão deste trabalho. Tratando-se de redes de sensores, muita pesquisa já foi desenvolvida e extensamente documentada. Diversos trabalhos na literatura compilam os avanços obtidos em estudos nesta área, abordando de problemas clássicos a questões mais específicas [Yick et al. 2008, Boukerche 2008, Zheng and Jamalipour 2009]. Sendo assim, informações mais genéricas sobre redes de sensores sem fio são omitidas neste trabalho.

## 2.1 REDES DE SENSORES E ATORES

Redes de sensores sem fio (RSSF) são redes compostas por dispositivos pequenos, dispersos geograficamente, com o objetivo de captar informações do ambiente e disponibilizá-las ao consumidor. Os sensores podem interagir cooperativamente na execução de alguma tarefa. O uso prático de RSSF tornou-se possível com o avanço das técnicas de miniaturização de componentes e circuitos [Yick et al. 2008]. Os dispositivos sensores são projetados para possuírem baixo custo de produção, resultando em processamento e memória reduzidos, baixas potências de transmissão e alimentação por baterias. As RSSF são de fácil implantação e podem funcionar sem infraestrutura fixa, possibilitando o uso em locais de difícil acesso e aplicações de alto risco. Normalmente a comunicação acontece por rádio-frequência, mas também são utilizadas tecnologias como *bluetooth* [Ke et al. 2009] e ondas sonoras [Domingo and Prior 2008]. Redes de sensores sem fio podem ser utilizadas com diversos propósitos: operações militares, agricultura, monitoramento de ambientes, indústria, medicina, entre outros [Akyildiz et al. 2002, Yick et al. 2008].

Em redes de sensores e atores, a informação coletada pelos sensores é disponibilizada ao usuário através de dispositivos chamados de atores. Esses são menos restritos que os sensores, dotados de uma fonte de energia irrestrita ou recarregável, um melhor transmissor, memória e processamento mais robustos. A quantidade de sensores disponibilizada na rede é muito maior que a quantidade de atores. O uso de uma abordagem baseada em ator favorece aplicações onde é necessária coleta de dados em tempo real com leituras rápidas e localizadas, por exemplo utilizando dispositivos atores móveis como robôs ou PDA (computador móvel pessoal), assim como a intervenção direta dos atores no ambiente [Akyildiz and Kasimoglu 2004].

Normalmente as redes de sensores são projetadas para funcionar de maneira independente, sem intervenção humana. Então, o sistema deve possuir a capacidade de auto-organização, ou seja, inicializar e manter o funcionamento de maneira autônoma, automática e transparente. A autonomia do sistema, como um todo, deve existir mesmo sob situações adversas como indisponibilidade do canal de comunicação, dano aos dispositivos, exaustão de bateria, entre outros. Também é desejável o funcionamento da rede mesmo se utilizada uma disposição imprevisível dos nós [Yick et al. 2008].

Para o correto funcionamento, uma rede de sensores e atores depende de mecanismos de coordenação ator-sensor e ator-ator [Akyildiz and Kasimoglu 2004]. A interação ator-sensor decorre da coordenação local de um dispositivo ator sobre os sensores ao seu redor. Já a coordenação entre atores é necessária, por exemplo, quando desejada a realização de ações conjuntas em reação a fenômenos no ambiente, como o controle de temperatura. Note que é inviável atribuir ao ator, ou a qualquer outra entidade centralizadora, a tarefa de controlar todos os sensores globalmente. Então, a capacidade de auto-organização dos sensores pode ser aplicada em caráter local, relativo a setores ou sub-redes, cabendo ao ator atividades de gerenciamento e coordenação.

O principal desafio em redes de sensores é viabilizar o funcionamento de aplicações mesmo sob suas restrições. Um dos pontos críticos para qualquer aplicação em redes de sensores é maximizar o tempo de vida da rede através da economia de energia. Outra questão importante está na garantia de serviços de rede necessários às aplicações [Yick et al. 2008]: identificação, roteamento, localização, segurança, entre outros. O fornecimento desses serviços deve levar em consideração as limitações dos dispo-

sitivos, as características do ambiente e os requisitos da aplicação.

## 2.2 LOCALIZAÇÃO EM REDES DE SENSORES

Para algumas aplicações é essencial a informação sobre a posição dos nós sensores na utilização adequada dos dados coletados. O problema de localização consiste em determinar a posição de objetos no espaço, segundo algum sistema de referência [Zheng and Jamalipour 2009]. Em RSSF, localização consiste em habilitar o sensor a estimar sua própria posição no espaço segundo algum ponto de referência [Patwari et al. 2005, Mao et al. 2007b], por exemplo, um dispositivo ator.

Existem duas abordagens distintas para localização em redes de sensores, onde a diferença básica está no método de estimar a posição em relação ao ponto de referência [He et al. 2005, Chaurasia and Payal 2011]. Esquemas *range-based* dependem da estimativa da distância euclidiana entre o nó e um ponto de referência, geralmente a um salto (dentro do alcance de transmissão). As alternativas *range-free* dispensam essa estimativa e determinam a posição relativa através de sistemas de coordenadas discretos, normalmente utilizando *beacons* transmitidos por nós referência específicos. Em geral, técnicas *range-based* são mais precisas, enquanto abordagens *range-free* são mais independentes de *hardware* [Zheng and Jamalipour 2009, Wang et al. 2009, Chaurasia and Payal 2011]. Os erros de estimativa são agravados por fatores relacionados ao meio de propagação, tais como obstáculos, temperatura, umidade, propagação por múltiplos caminhos, interferência ou ruído. Técnicas *range-based* são mais suscetíveis a esses efeitos. Uma alternativa para melhorar a precisão, para ambos os casos, é utilizar-se de mecanismos de cooperação entre sensores no processo de localização [Patwari et al. 2005].

Soluções *range-based* necessitam estimar a distância euclidiana para o processo de localização. As técnicas mais comuns para isso são AoA<sup>1</sup> (ângulo de chegada da portadora), ToA (tempo de chegada de pulsos da portadora), TDoA (diferença entre os tempos de chegada de pulsos da portadora) e RSSI (força do sinal recebido). AoA utiliza a medição do ângulo de chegada da onda portadora. ToA utiliza medidas de tempo de chegada de fases da portadora, enquanto TDoA utiliza a diferença entre os tempos de chegada.

---

<sup>1</sup>De fato, AoA não mede distância diretamente, mas é citada aqui por ser utilizada em alguns métodos *range-based*.

RSSI (*Received Signal Strength Indicator*) estima a distância através da força do sinal recebido. Todas estas técnicas necessitam de *hardware* específico e são suscetíveis a erros de estimativa influenciados pelo do meio [Mao et al. 2007b]. RSSI é mais utilizada por ser mais simples, demandar *hardware* comum de menor custo e por sofrer menos com as condições do meio se comparada às demais [Mao et al. 2007b, Chaurasia and Payal 2011]. Contudo, medidas RSSI são assimétricas<sup>2</sup> e apresentam alta variabilidade temporal e espacial [Patwari and Kaseria 2011]: dados dois nós estáticos, é possível que os valores medidos para força de sinal variem com o tempo. Logo, é necessário tomar várias medidas para aumentar a precisão.

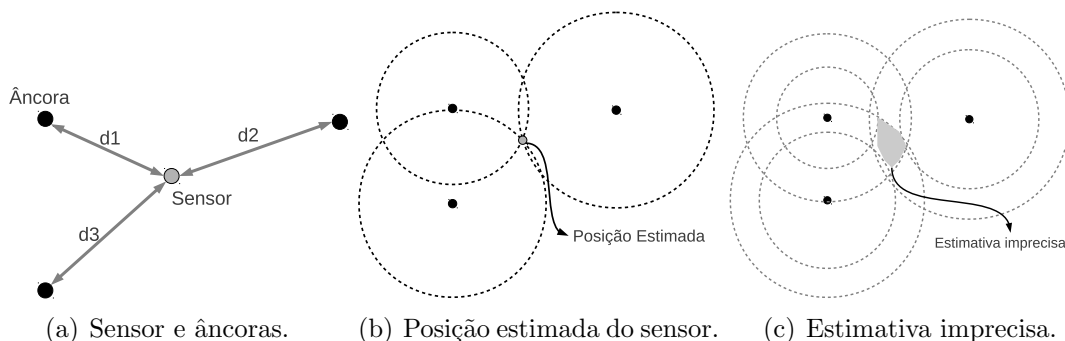


Figura 2.1: Multilateração utilizando 3 âncoras.

Através da escuta de *beacons*, por exemplo, o sensor pode estimar a distância a alguns pontos de referência de posição conhecida, também chamados de nós âncora, e determinar sua posição [Zheng and Jamalipour 2009]. Esse método é chamado de multilateração<sup>3</sup>. Naturalmente, quanto maior for a quantidade de âncoras, melhor será a estimativa de localização do sensor. A Figura 2.1 mostra um exemplo de multilateração com três nós âncora (trilateração). Na Figura 2.1(a) um sensor estima sua distância até os nós de referência. Se o *beacons* transmitidos pelos nós âncora contiverem informações sobre as coordenadas do nó âncora transmissor, o sensor pode definir sua posição como a intersecção de circunferências com centro no âncora e raio igual à distância medida, como mostra a Figura 2.1(b). A presença de erros ou divergências nas medidas de distância pode resultar em situações onde não haja intersecção exata entre as circunferências [Chang and Liao 2009]. A Figura 2.1(c) ilustra uma situação onde a posição

<sup>2</sup>A força do sinal recebido de A para B pode ser diferente da força do sinal recebido de B para A.

<sup>3</sup>Tradução livre para *multilateration*



estimada é imprecisa devido à variabilidade da distância estimada (a posição estimada do sensor é a área hachurada).

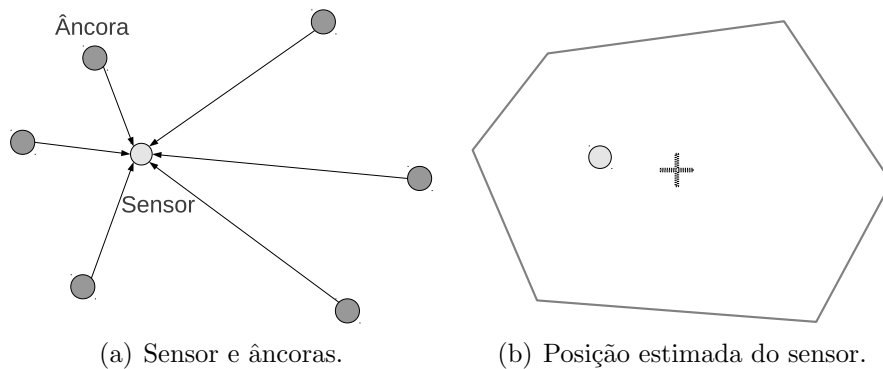


Figura 2.2: Algoritmo centroide. A cruz representa a posição estimada pelo sensor.

Por dispensar medidas de distância euclidiana, soluções *range-free* geralmente utilizam modelos discretos para determinar a posição do sensor, como polígonos, número de saltos, setores, grades, entre outros. Por exemplo, a Figura 2.2 simplifica o funcionamento do algoritmo centroide [Bulusu et al. 2000]. A Figura 2.2(a) mostra os nós âncora identificados pela transmissão frequente de *beacons* contendo suas coordenadas. Na Figura 2.2(b), o sensor estima sua posição calculando o centro de massa do polígono formado pelos nós âncora no alcance de transmissão, cujos *beacons* foram recebidos.

Na literatura não há um consenso sobre qual abordagem é mais eficiente. A melhor estratégia para localização em redes de sensores depende da aplicação, das limitações do *hardware* utilizado, do ambiente, entre outros fatores. Para métodos que utilizam múltiplas âncoras, a precisão de localização depende também da distribuição dos nós âncora. Para redes em larga escala, isso representa um grande desafio [He et al. 2005, Chang and Liao 2009]. Por fim, perda de *beacons* influencia negativamente em protocolos de localização, tanto *range-free* quanto *range-based*.

Recentemente, a ideia de treinamento de localização para redes de sensores e atores foi introduzida na literatura [Barsi et al. 2009, Barsi et al. 2011]. Esse tipo de treinamento é um método *range-free*, sendo vantajoso em relação a outros métodos por dispensar o uso de múltiplas âncoras. Além disto, os protocolos dispensam a sincronização e a comunicação entre os sensores durante o processo de treinamento, resultando em alta escalabilidade. O treinamento é baseado apenas na escuta de *be-*

*acons* transmitidos pelo ator, de modo a permitir os sensores definir sua localização baseando-se em coroas e setores. Protocolos e aplicações são favorecidas por este esquema uma vez que tal esquema de organização permite algumas facilidades, tais como [Wadaa et al. 2005, Navarra and Pinotti 2011]:

- O uso de coroas concêntricas permite mecanismos de roteamento mais naturais para a transmissão do sensor para o ator e vice-versa;
- A localização em coroas e setores facilita na formação de *clusters* geograficamente estabelecidos. Por exemplo, dois sensores pertencem ao mesmo *cluster* se estiverem localizados na mesma coordenada (coroa e setor);
- Certas aplicações de monitoramento de propriedades (e.g. umidade, temperatura) podem identificar e administrar correções para problemas por coroas/setores.

## 2.3 MODELO DE REDE

Neste trabalho são consideradas redes de sensores centradas em ator<sup>4</sup>, ou simplesmente RSCA, onde o ator coordena os sensores, de maneira local, de modo a serem formadas sub-redes de curta duração destinadas à realização de tarefas específicas [Akyildiz and Kasimoglu 2004, Barsi et al. 2009, Barsi et al. 2011]. Após ser cumprida a tarefa, os sensores voltam ao seu estado desorganizado. Os dispositivos são desprovidos de qualquer informação global, atendendo a um esquema de auto-organização local. Os sensores também são desprovidos de qualquer informação inicial sobre localização.

O mecanismo de localização descrito neste trabalho consiste em classificar os sensores segundo um sistema de coordenadas polares discreto com origem no ator. A distância do nó ao ator é classificada segundo coroas circulares enquanto os ângulos são discretizados formando setores. A Figura 2.3 mostra um exemplo de sub-rede formada segundo o sistema de coordenadas descrito. Assume-se que o ator possua duas antenas: uma isotrópica, capaz de transmitir em vários níveis de potência, e outra direcional. Desta forma, o treinamento é realizado através de *beacons* transmitidos pelo ator. A transmissão isotrópica permite ao sensor determinar a coroa ao qual pertence e a transmissão

---

<sup>4</sup>Do inglês *actor-centric*

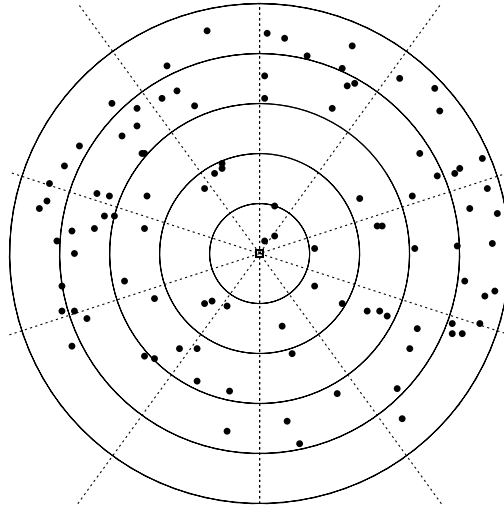


Figura 2.3: RSCA dividida em 5 coroa e 10 setores.

direcional determina o seu respectivo setor. Adicionalmente, caso o ator saiba sua localização (por exemplo via GPS), é possível uma informação absoluta sobre a localização dos sensores.

Assim como os trabalhos descritos em [Barsi et al. 2009, Barsi et al. 2011], aqui serão tratados apenas o treinamento dos sensores em coroa, visto que o treinamento em setores é menos complexo e pode ser realizado de maneira semelhante ao treinamento em coroa [Barsi et al. 2011]. Para funcionamento dos algoritmos são feitas as seguintes considerações. Um ator transmite a  $k$  níveis de potência, onde  $k$  é conhecido pelo ator e pelos sensores. Logo,  $k$  determina a quantidade de coroa que divide a sub-rede. Em geral, quando o ator transmite um *beacon*  $b_i$  o alcance máximo é um sensor localizado na coroa  $C_i$ . Em outras palavras, o *beacon*  $b_i$  transmitido à potência  $p_i$  identifica a coroa  $C_i$ . A Figura 2.4 ilustra um exemplo para  $k = 5$ . O tempo é dividido em *slots* de tamanho fixo. A cada *slot*, o ator transmite um *beacon*  $b_i$ , segundo um comportamento predefinido.

São considerados dois tipos de sensores. Os sensores periódicos possuem ciclos de funcionamento de tamanho fixo  $L_n$ , onde escutam por  $d_n$  *slots* e entram em inatividade até o próximo ciclo. Sensores livres escutam por  $d_n$  *slots*, mas podem controlar dinamicamente seu ciclo de trabalho e inatividade. A Figura 2.5 ilustra o esquema de funcionamento dos dois tipos de sensores. São considerados sensores não sincronizados, os quais utilizam relógios internos próprios e operam em ciclos de trabalho independentes. A rede é dita he-

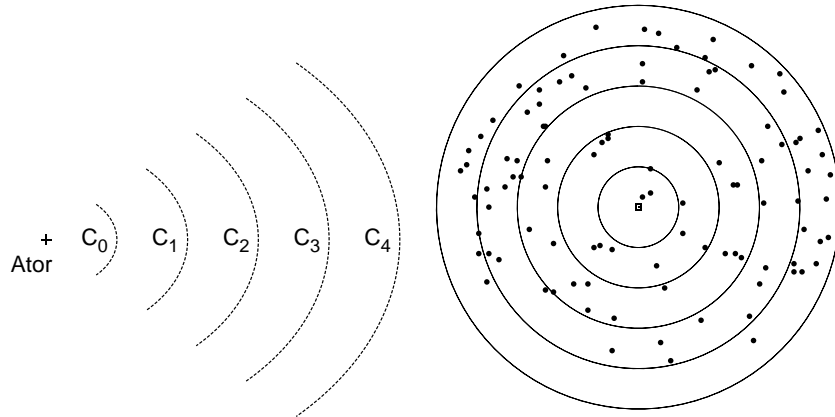
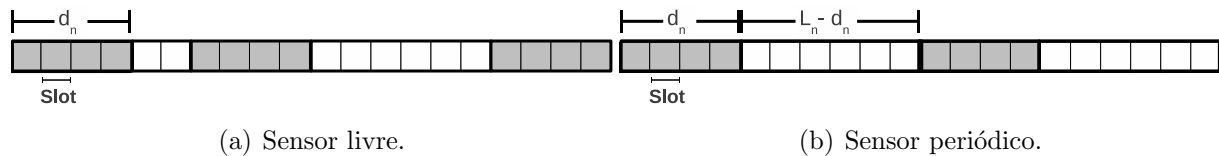


Figura 2.4: RSCA dividida em 5 coroas.

terogênea porque pode conter sensores de ambos tipos e com ciclos de trabalho distintos. Por fim, o sensor conhece o comportamento do ator, embora não necessariamente saiba quais *beacons* foram transmitidos enquanto o sensor estava inativo. O comportamento de cada ator será apresentado propriamente no Capítulo 4.



(a) Sensor livre.

(b) Sensor periódico.

Figura 2.5: Tipos de sensores na rede.

O objetivo do treinamento é possibilitar ao sensor estimar sua localização em relação ao ator apenas escutando *beacons* transmitidos. Os protocolos assíncronos de treinamento devem suportar redes heterogêneas, compostas por sensores livres e periódicos, com distintos ciclos de trabalho  $d_n$ . O treinamento de localização deve ser realizado de maneira eficiente, visando o baixo consumo energético por parte dos sensores.

## RESUMO

Este capítulo apresentou os conceitos sobre treinamento de localização em redes de sensores centradas em ator. As redes de sensores em geral possuem requisitos fortemente ligados à aplicação e às restrições dos dispositivos. Isso deve ser considerado no fornecimento de serviços de rede, tais como localização relativa a um ponto. Em especial, as

redes centradas em ator podem tomar como referência de localização o próprio ator. O conceito de treinamento de localização em RSCA deriva da ideia de permitir ao sensor estimar sua localização através da escuta de *beacons* transmitidos por um ator. Neste capítulo também foi definido o modelo de rede utilizado neste trabalho.

## CAPÍTULO 3

# TRABALHOS RELACIONADOS

Pesquisas em RSSF já vêm sendo conduzidas há cerca de duas décadas, resultando em certo amadurecimento na área. Entretanto, ainda há muitas questões pertinentes no estudo das RSSF, neste caso, especificamente, o problema de localização. Este capítulo apresenta alguns trabalhos recentes com o intuito de dar uma visão geral ao leitor sobre o atual estado da arte. Este capítulo também serve como ponto de partida para a compreensão do problema discutido neste trabalho.

### 3.1 O PROBLEMA DE LOCALIZAÇÃO EM REDES DE SENSORES

Tratando do problema de localização, [Mao et al. 2007b] apresenta um estudo geral sobre técnicas utilizadas nos protocolos de localização para RSSF, sobretudo lidando com técnicas para medida de distância euclidiana. É importante salientar que tais técnicas de medição são utilizadas para medidas a um salto, ou seja, para calcular a distância a um nó dentro do alcance de transmissão. Das técnicas de medição existentes, a mais utilizada é RSSI (medida da força do sinal) por demandar *hardware* menos específico, por dispensar condições de sincronização (ao contrário de ToA e TDoA) e ser menos suscetível a variações do meio se comparado às demais. O trabalho também disserta sobre os efeitos de comunicação NLoS (*Non Line of Sight*, comunicação sem visada direta) e comunicação a múltiplos saltos sobre o erro de localização e a propagação desse erro de acordo com cada salto. Uma modelagem proposta por [Huang et al. 2010] verifica o comportamento da propagação do erro em mecanismos de localização a múltiplos saltos. Através de simulações foi obtida uma equação onde o quadrado do erro médio cresce de acordo com o cubo da distância em saltos do sensor ao nó âncora.

Protocolos que utilizam âncoras dependem da distribuição dos sensores e âncoras no espaço [Zheng and Jamalipour 2009, Cheng et al. 2007]. Muitas vezes uma distri-

buição planejada não é possível, resultando em áreas com distribuição irregular e densidade insuficiente de âncoras. Esse é um grande obstáculo para o uso de âncoras em protocolos de localização. [Cheng et al. 2007] propõe uma abordagem mista onde são combinados dois protocolos. O primeiro possui alta complexidade e funciona bem mesmo com baixa densidade de âncoras. O outro, mais simples, funciona apenas sob boa distribuição dos nós. Simulações mostram que a precisão obtida ao usar os protocolos combinados é superior ao uso de cada protocolo individualmente. Contudo, a combinação de protocolos apenas ameniza os efeitos da má distribuição dos nós. Alguns trabalhos utilizam a ideia de âncoras móveis, também chamados de *beacons* móveis [Kim and Lee 2007, Lee et al. 2009], onde os nós referência podem eventualmente se mover para regiões de baixa densidade de âncoras. Outra possibilidade é instanciar âncoras secundários a partir de nós sensores que já estimaram suas posições ou utilizar a estimativa de distância a múltiplos saltos, embora essas abordagens resultem na propagação dos erros de estimativa [Zheng and Jamalipour 2009, Huang et al. 2010].

Em [Lin et al. 2009] é feito um estudo sobre o impacto da perda de *beacons* sobre um protocolo de localização *range-based* que usa RSSI e um algoritmo KNN para refinamento<sup>1</sup>. Nos experimentos descritos no trabalho, alguns dispositivos disparam *beacons* frequentemente, enquanto um outro armazena um vetor de assinaturas RSSI. É utilizado um algoritmo KNN sobre o vetor e estimada a posição através da média ponderada das  $K$  melhores posições. Os experimentos conduzidos utilizam 14 nós espalhados em ambiente interno. No cenário descrito, a taxa de perda de *beacons* depende da disponibilidade do canal, variando entre 42% e 83%. É constatada uma diferença de 300% entre os erros de localização obtidos nas medidas com canal livre e ocupado. Esse resultado é importante, uma vez que muitos protocolos de localização existentes na literatura negligenciam condições de perda de *beacons*.

Em [Chang and Liao 2009] é tratado o problema da localização através de um modelo probabilístico *range-based*, levando em consideração fenômenos de desvanecimento no canal de comunicação. Desvanecimento prejudica ainda mais na acurácia de medidas RSSI por causar variação na força do sinal recebido. O trabalho limita sua aplicação a situações onde a distribuição dos nós âncora é conhecida, uma vez que o planejamento

---

<sup>1</sup>Um algoritmo KNN utiliza informações de  $k$  colaboradores vizinhos na classificação de alguma propriedade.

na distribuição dos nós resulta em melhor desempenho. Esse tipo de distribuição pode ser aplicado geralmente em ambientes internos, por exemplo, em aplicações domésticas. Para redes de larga escala, isso nem sempre é possível. O protocolo proposto parte de um conjunto de medidas  $D = d_1, d_2, \dots, d_n$  de distância a  $n$  nós âncora e estima a posição  $(x', y')$  do nó através da probabilidade *a posteriori*, calculada via lógica *bayesiana*. Os experimentos constataam que o uso de cálculo probabilístico compensa os efeitos do desvanecimento sobre RSSI, acarretando em melhor precisão do protocolo proposto em comparação a duas outras soluções presentes na literatura.

### 3.2 PROTOCOLOS *RANGE-BASED*

Na literatura, pode ser encontrada uma grande diversidade de protocolos *range-based* para determinação de localização [Mao et al. 2007b, Zheng and Jamalipour 2009]. Em geral, os protocolos propostos funcionam seguindo dois passos: medição da distância euclidiana e estimativa da distância através da combinação das informações. Há trabalhos que estudam outras características da localização *range-based*, como comportamento das medidas RSSI, propagação de erros e distribuição de âncoras.

Em [Al-Dhalaan and Lambadaris 2010] o problema de localização é tratado como um problema de otimização com múltiplas soluções. O trabalho utiliza um modelo probabilístico para representar os efeitos das imperfeições do canal, verificando correlação espacial de perfis RSSI, isto é, determinando como a força do sinal recebido varia de acordo com a posição dos nós. O modelo é utilizado para estimar com mais precisão a distância transmissor/receptor a um salto.

Em [Zhang et al. 2009] foram conduzidos experimentos com o intuito de verificar a precisão da estimativa de distância via RSSI. Dois transceptores próprios para dispositivos sensores foram testados em cenários distintos. Nos cenários internos, há baixa relação entre a distância e as medidas RSSI. Para os cenários externos, o trabalho propõe um modelo de correlação baseado em ajuste não linear. Os experimentos mostraram a relação entre o erro de estimativa obtido por trilateração e a distância transmissor/receptor. O modelo considera baixa presença de interferência no meio de comunicação.

[Ameer and Jacob 2010] aborda o problema de localização para redes de sensores subaquáticas onde a comunicação é realizada por ondas sonoras. O protocolo utiliza técnicas



de *ray tracing* para determinar a localização de nós âncora, realizar uma estimativa inicial sobre a posição do sensor e refinar a estimativa através de técnicas de otimização baseadas em *simplex*. Por sua vez, em [Zhou et al. 2010] é proposta uma solução baseada em hierarquia para localização em redes de sensores sub-aquáticas. Recentemente, cresceu o interesse nos estudos em redes de sensores sub-aquáticas, em geral redes de larga escala. Protocolos de localização também são necessários nesse tipo de aplicação.

### 3.3 PROTOCOLOS *RANGE-FREE*

Dos protocolos *range-free*, as soluções mais conhecidas são o *Centroid* [Bulusu et al. 2000] e *DV-Hop* [Nagpal et al. 2003]. *Centroid* baseia-se no cálculo do centro de massa de um polígono formado por âncoras ao alcance do sensor. A precisão está diretamente relacionada à quantidade de âncoras e a sua distribuição. Extensões ao protocolo *Centroid* podem ser encontradas na literatura [Pivato et al. 2011, Wang et al. 2011]. A abordagem *DV-Hop* estima a posição do nó baseando-se na distância em saltos do nó até nós âncora. Essa abordagem apresenta limitações quando consideradas redes em larga escala por demandar muita troca de mensagens, sobretudo sob escassez de nós âncora. Em [Wang et al. 2010] o consumo energético do *DV-Hop* é reduzido através do limite de saltos permitidos aos *beacons*. Por sua vez, [Ying et al. 2010] sugere outras três alterações no *DV-Hop* com o intuito de melhorar a precisão.

Na literatura existe uma diversidade de propostas de protocolos de localização *range-free*, inclusive com o recente aumento de interesse em mecanismos desse tipo por apresentarem melhor custo-benefício. [He et al. 2005] propõe um esquema de localização onde o sensor determina sua posição segundo triângulos formados por trios de âncoras. Simulações mostram melhorias na precisão em comparação com o *Centroid* e com o *DV-Hop*. Em [Chen et al. 2010] é proposto um protocolo onde o sensor estima sua posição segundo cordas que cortam as circunferências formadas pelos nós âncora ao seu alcance. Através de simulações é demonstrada uma melhoria de precisão de até 30% em relação ao *Centroid*. Por sua vez, [Xiao et al. 2010] considera o problema de localização em redes com transmissão anisotrópica e propõe um protocolo *range-free* baseado em saltos, levando em consideração padrões de transmissão.

Em [Wang and Zhu 2009] é proposto um mecanismo *range-free* baseado em modela-

gem probabilística *Monte Carlo*<sup>2</sup>. São considerados cenários com âncoras de mobilidade imprevisível e irregularidades no canal de comunicação. No protocolo, o nó de posição desconhecida escuta *beacons* dos nós âncora, refinando sua localização contínua e iterativamente, como um problema de otimização. Resultados de simulação mostram a eficiência do protocolo em termos de precisão e tráfego, sobretudo sob condições de baixa densidade de âncoras. Em [Zhang et al. 2010a] é utilizada a abordagem *Monte Carlo* para melhoria de técnicas baseadas em *bounding-box*<sup>3</sup>.

Um estudo sobre erros de localização para mecanismos *range-free* é feito em [Zhang et al. 2010b]. O trabalho considera algoritmos baseados em regiões, tais como aqueles baseados em centroide e *bounding-box*. Através de análise teórica, é derivado um limite para a precisão de localização que depende do tamanho da região e na quantidade de partições da mesma. O trabalho também demonstra que utilizar partições com aproximadamente o mesmo tamanho melhora a precisão. Simulações são utilizadas para validar o modelo matemático, sendo obtida correlação satisfatória.

### 3.4 LOCALIZAÇÃO EM REDES DE SENSORES E ATORES

Na literatura também podem ser encontrados protocolos de localização próprios para redes de sensores e atores. Protocolos específicos para este ambiente são desejáveis, por exemplo, em aplicações que exijam apenas que os nós próximos ao ator descubram sua localização, inclusive resultando em menor custo energético por evitar que o protocolo de localização seja executado pela rede inteira. Esse caráter local também dispensa o emprego de alta densidade de âncoras. O trabalho descrito em [Peng et al. 2005] defendem tais argumentos, além de propor um protocolo de localização que só entra em funcionamento na ocorrência de um evento de interesse. A posição é calculada de maneira semelhante ao *DV-Hop* com limitação de saltos, onde os atores assumem o papel de âncoras de maneira cooperativa.

Em RSCA (redes de sensores centradas em ator), pode ser utilizado um sistema de coordenadas polares com o ator ao centro. [Ghidini et al. 2010] propõe um mecanismo semi-distribuído para localização em coroas circulares. O protocolo funciona em duas

---

<sup>2</sup>Em abordagens *Monte Carlo* são utilizadas grande quantidades de amostras aleatórias na solução de problemas de otimização

<sup>3</sup>*Bounding box* consiste em um retângulo que engloba um conjunto dado de pontos

etapas. A primeira define um treinamento de sensores próximos ao ator. Na segunda etapa, os sensores treinados enviam *beacons* para auxiliar na localização de sensores fora do alcance do ator. Em [Navarra et al. 2010] é proposto um protocolo semelhante, com uma terceira etapa adicional. A terceira etapa compreende um treinamento adicional, de precisão reduzida, a sensores que não foram treinados por conta de ruído (receberam *beacons* de mais de um sensor). Em [Navarra and Pinotti 2011] é proposto um protocolo de treinamento semelhante que considera a cooperação entre os sensores, distribuídos uniformemente, durante o processo de treinamento. Ambos protocolos citados neste parágrafo consideram sensores periódicos e o modelo de rede como descritos na Seção 2.3.

Ainda tratando de treinamento em coroas para RSCA, em [Barsi et al. 2009] são propostos protocolos de treinamento que dispensam a comunicação entre os sensores. O trabalho define, de maneira incremental, o protocolo *Flat+*. No mesmo trabalho, uma variação do *Flat+* chamada *Two Level* também é proposta. O *Two Level* divide o espaço de coroas em dois níveis, um externo e um interno, realizado o *Flat+* em cada nível. *Binary Training*, um protocolo baseado em busca binária é proposto em [Barsi et al. 2011]. Os protocolos *Flat+* e *Binary Training* serão analisados com mais detalhes nos Capítulos 4 e 5.

## RESUMO

Este capítulo apresentou uma revisão bibliográfica relacionada ao problema de localização em redes de sensores. Existe uma diversidade de protocolos de localização em RSSF, tanto *range-free* quanto *range-based*. Os problemas de localização em RSSF estão relacionados à distribuição de nós âncora, sobretudo em redes de larga escala, na comunicação adicional necessária a protocolos distribuídos e na propagação de erros, seja pela estimativa de distância ou por baixa acurácia (no caso dos protocolos *range-free*). Em redes de sensores e atores, o problema da localização pode ser tratado localmente, organizando os sensores em volta do ator. Recentemente, alguns protocolos foram propostos com este intuito, nesse caso dispensando o uso de múltiplas âncoras. Os protocolos utilizados neste trabalho foram escolhidos por dispensar troca de mensagem entre os sensores.

## PROTOCOLOS ASSÍNCRONOS DE TREINAMENTO

Como visto em capítulos anteriores, localização é um serviço importante para uma série de aplicações. Protocolos de treinamento de localização são alternativas *range-free* que possibilitam aos sensores a descoberta de sua localização de maneira independente, sem troca de mensagens. Os protocolos propostos em [Barsi et al. 2009, Barsi et al. 2011] são próprios para treinamento de RSCA de larga escala, além de serem assíncronos, isto é, os sensores possuem ciclos de trabalho distintos e a ativação do sensor é independente dos demais, não existindo um mecanismo de sincronização explícito. Assim, o treinamento assíncrono constitui uma solução escalável para o problema de localização. Este capítulo descreve esses protocolos de treinamento, assim como analisa suas características de maneira preliminar.

### 4.1 O PROTOCOLO FLAT+

Uma maneira simples para o sensor determinar a coroa onde está localizado é analisar os *beacons* recebidos de maneira sequencial e encontrar exatamente o menor *beacon* escutado na sequência. O protocolo *Flat+* funciona seguindo esse princípio.

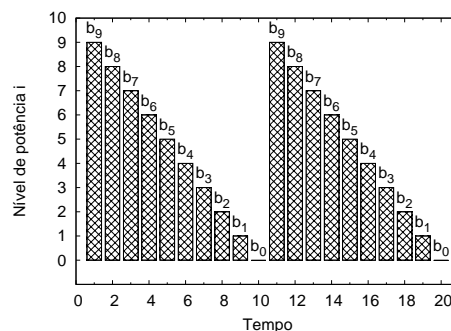


Figura 4.1: Comportamento do ator *Flat+*,  $k = 10$ .

Proposto em [Barsi et al. 2009], o protocolo *Flat+* mantém um vetor  $R_k[ ]$  que arma-

zena a informação se um *beacon* é ou não recebido pelo sensor. O sensor estará treinado na coroa  $C_i$ ,  $i > 0$ , se receber o *beacon*  $b_i$  e não receber  $b_{i-1}$ . No caso especial para  $i = 0$ , o sensor estará treinado na coroa  $C_0$  se receber o *beacon*  $b_0$ . A Figura 4.1 ilustra o comportamento do ator para o funcionamento do protocolo *Flat+*. O ator transmite ciclicamente *beacons*  $b_i$  diminuindo o nível de potência a cada *slot* de tempo.

---

**Algoritmo 4.1** Algoritmo Flat+.
 

---

```

1: heard ← false
2: trained ← false
3: min ←  $k - 1$ ; max ← 0; v ← 0
4: inicializa cada elemento de  $R_k[ ]$  ← 0
5: while ¬trained do
6:   v ← v + 1
7:   for  $i \leftarrow 0$  to  $d_n - 1$  do do
8:     if recebeu  $b_c$  then
9:       if ¬heard then
10:        heard ← true
11:         $t \leftarrow k - 1 - c$ 
12:        for  $j \leftarrow v - 1$  to 0 do
13:           $max \leftarrow \text{Max}(max, [c + j \cdot (L_n \bmod k) + v] \bmod k)$ 
14:        for  $h \leftarrow 0$  to max do
15:           $R_k[h] \leftarrow -1$ 
16:        for  $h \leftarrow c$  to min do
17:           $R_k[h] \leftarrow 1$ 
18:        if  $c = 0$  or ( $R_k[c] = 1$  and  $R[c - 1] = -1$ ) then
19:          trained ← true
20:          corona ←  $C_c$ 
21:          min ←  $c$ 
22:           $t \leftarrow t + 1$ 
23:        else
24:          if heard then
25:             $c \leftarrow k - 1 - t \bmod k$ 
26:            for  $h \leftarrow max$  to  $c$  do
27:               $R_k[h] \leftarrow -1$ 
28:            max ←  $c$ 
29:            if  $R_k[c + 1] = 1$  then
30:              trained ← true
31:              corona ←  $C_{c+1}$ 
32:             $t \leftarrow t + 1$ 
33:        if ¬trained then
34:          if ¬heard then
35:            espera  $L_n - d_n$  slots
36:          else
37:            espera Wait() slots

```

---

O Algoritmo 4.1 detalha como os sensores realizam o treinamento. Os registradores *min* e *max* armazenam, respectivamente, o menor *beacon* ouvido e o maior não ouvido. Após a recepção do primeiro *beacon* (linha 9), o sensor se sincroniza com o ator (linhas 10-11), determina quais *beacons* previamente transmitidos não foram escutados (linhas

12-13) e ajusta *max* e *min* (linhas 13 e 21). A partir da primeira recepção, *min* será ajustado apenas quando um *beacon* é recebido, enquanto *max* é ajustado quando um *beacon* não é recebido. A sincronização com o ator permite ao sensor determinar qualquer *beacon* não escutado (linha 22). Os elementos do vetor  $R_k[\ ]$  são preenchidos de acordo com os valores de *min* (linhas 16-17) e *max* (linhas 14-15 e 26-27). O treinamento é alcançado quando o sensor atinge as condições descritas nas linhas 18-20 ou 29-31.

---

**Algoritmo 4.2** Procedimento *Wait()* do Algoritmo *Flat+*.

---

```

1:  $z \leftarrow 0$ 
2: while  $R_k[j] \neq 0, \forall b_j$  transmitido entre  $(t + z \cdot L_n)$  e  $(t + z \cdot L_n + d_n - 1)$  do
3:    $z \leftarrow z + 1$ 
4:  $wait = L_n - d_n + z \cdot L_n \text{ slots}$ 
5: return  $wait$ 

```

---

O protocolo assume que se o sensor recebe um *beacon*  $b_i$  então receberá qualquer *beacon*  $b_j$ ,  $j > i$ . De modo análogo, se o sensor não recebe  $b_i$ , então não receberá qualquer  $b_j$ ,  $j < i$ . Como o comportamento do ator é conhecido, é possível prever e evitar ciclos onde a escuta se torna desnecessária, através do procedimento *Wait()* (linha 37) descrito no Algoritmo 4.2, resultando em maior eficiência energética. A cada iteração (ciclo de funcionamento), caso o sensor não esteja treinado, é necessário calcular o tempo de espera do sensor. O Algoritmo 4.2 descreve o procedimento que calcula a quantidade de *slots* em espera, o qual ignora o próximo ciclo  $L_n$  se  $\forall b_i$  a ser transmitido em  $L_n$  tem-se  $R_k[i] \neq 0$ .

Note que o *Flat+* treina apenas sensores periódicos. O treinamento é possível caso  $d_n \geq mdc(L_n, k)$ . No pior caso, são realizadas  $\psi_{max} = \frac{k}{mdc(L_n, k)}$  iterações, o tempo de atividade do sensor é de  $\omega_{max} = d_n \cdot \psi_{max}$  e o tempo total decorrido é  $\tau_{max} = L_n \cdot \psi_{max}$  [Barsi et al. 2009]. A função  $mdc(x, y)$  indica o máximo divisor comum de  $x$  e  $y$ .

De antemão, o estudo do protocolo *Two Level*, também proposto em [Barsi et al. 2009], foge ao escopo deste trabalho, por se tratar de uma extensão do *Flat+* baseada em níveis, também restrita a sensores periódicos.

## 4.2 O PROTOCOLO BINARY TRAINING

O protocolo *Binary Training* [Barsi et al. 2011] possui algumas vantagens em relação ao *Flat+*, como suporte a sensores livres, melhor desempenho e maior resistência a desvios nos relógios dos dispositivos. A proposta utiliza busca binária para determinar um *beacon*

de interesse e ativar o sensor quando esse for transmitido.

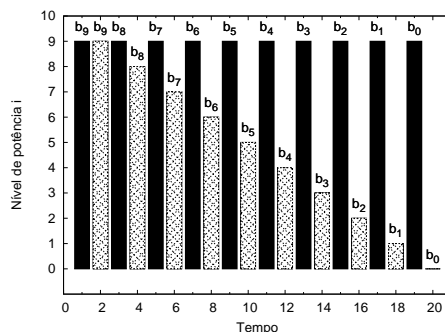


Figura 4.2: Comportamento do ator *Binary Training*,  $k = 10$ .

A Figura 4.2 ilustra o comportamento do ator com o protocolo *Binary Training*. Pares de *beacons* sucessivos  $b_i$  são transmitidos em *slots* consecutivos. O primeiro *beacon* do par é transmitido com potência máxima e é chamado de *beacon control*. O segundo *beacon* do par é transmitido com potência  $p_i$  e é conhecido por *beacon data*. À medida que os pares de *beacons* vão sendo transmitidos, a potência de transmissão dos *beacons data* é reduzida sequencialmente até o menor nível. A justificativa para esta modificação é reduzir os efeitos causados por irregularidades nos relógios do sensor e do ator (em outras palavras, um mecanismo implícito para resincronização).

O Algoritmo 4.3 detalha o algoritmo de treinamento executado pelo sensor. O protocolo utiliza os registradores *min* e *max* para determinar a coroa  $C_i$ ,  $0 \leq i < k$ , na qual o sensor está localizado. A coroa  $C_i$  estimada estará no intervalo  $[min, max]$ . Os *beacons* são analisados a cada par de *slots* (linhas 6-9 e 11-31) para determinar qual deles foi *control* e *data*. O registrador *max* determina o maior *beacon control* menor que  $C_i$  e é atualizado pela escuta do par de *beacons control* e *data* (linhas 17-21 ou 22-26). O registrador *min* determina o menor *beacon data* maior ou igual a  $C_i$  e é atualizado quando o sensor escuta o *beacon control* mas não escuta seu respectivo *data* (linhas 11-15 ou 27-31). O objetivo da busca binária é escutar o *beacon*  $b_{guess}$ ,  $guess = \lceil \frac{min+max}{2} \rceil$ , a fim de reduzir o tamanho do intervalo  $[min, max]$ . Quando satisfeita a condição  $max - min = 1$ , o sensor estará treinado e pertencerá à coroa  $C_{max}$  (linhas 33-35). Caso contrário, o sensor entrará em inatividade, através da execução do procedimento *Wait()*, por um período determinado de *slots* (linha 38).

Em cada iteração, o procedimento *Wait()* calcula quantos *slots* são necessários para a

**Algoritmo 4.3** Algoritmo *Binary Training*.

---

```

1: trained ← false
2: min ← -1; max ← k - 1; v ← 0; t ← 0
3: while ¬trained do
4:   for i ← 0 to dn - 1 do
5:     if i é par then
6:       if recebe bc then
7:         first ← c
8:       else
9:         first ← k
10:    else
11:      if não recebe bc then
12:        if min ≤ first then
13:          min ← first
14:          update ← left
15:          control ← t + i - 1
16:        else
17:          if c = first then
18:            if max ≥ c then
19:              max ← c
20:              update ← right
21:              control ← t + i - 1
22:          if first ≠ k and c = (first - 1) mod k then
23:            if max ≥ first then
24:              max ← c
25:              update ← right
26:              control ← t + i
27:          if first = k then
28:            if min ≤ (c + 1) mod k then
29:              min ← (c + 1) mod k
30:              update ← left
31:              control ← t + i
32:    t ← t + dn
33: if max - min = 1 then
34:   trained ← true
35:   corona ← Cmax
36: else
37:   guess = ⌈ $\frac{\textit{min} + \textit{max}}{2}$ ⌉
38:   espera Wait() slots

```

---

transmissão do *beacon*  $b_{guess}$  e depende do tipo do sensor. O resultado também depende de qual registrador, *min* ou *max*, foi ajustado na recepção do último par de *beacons*, armazenado pelo registrador *update*. Os Algoritmos 4.4 e 4.5 descrevem, respectivamente, os procedimentos *Wait*() utilizados no sensor livre e no sensor periódico. O procedimento *Wait*() para sensores livres calcula apenas o tempo necessário para a próxima transmissão de  $b_{guess}$ . Já no caso dos sensores periódicos o procedimento *Wait*() ignora o número necessário de *slots* até o ciclo mais próximo onde  $b_{guess}$  será transmitido.

Como demonstrado em [Barsi et al. 2011], a única condição necessária para o treinamento é que  $d_n$  seja par. No caso dos sensores periódicos, também deve ser satisfeita a



---

**Algoritmo 4.4** *Wait()* para o sensor livre.

---

```

if update = right then
  wait  $\leftarrow 2 \cdot \lfloor \frac{\text{max}-\text{min}}{2} \rfloor$ 
else
  wait  $\leftarrow 2 \cdot (k - \lfloor \frac{\text{max}-\text{min}}{2} \rfloor)$ 
return wait

```

---



---

**Algoritmo 4.5** *Wait()* para o sensor periódico.

---

```

if update = right then
  firstcorona  $\leftarrow (\text{max} + \frac{d_n}{2}) \bmod k$ 
else
  firstcorona  $\leftarrow \text{min}$ 
z  $\leftarrow 1$ 
while guess  $\notin [(\text{firstcorona} - z \cdot \frac{L_n}{2} - \frac{d_n}{2} + 1) \bmod k, (\text{firstcorona} - z \cdot \frac{L_n}{2}) \bmod k]$  do
  z  $\leftarrow z + 1$ 
return  $z \cdot L_n - d_n + 1$ 

```

---

condição  $\frac{d_n}{2} \geq \text{mdc}(\frac{L_n}{2}, k)$ . Para os sensores livres, são obtidos os valores máximos de  $\psi_{max} = 1 + \lceil \log_2 k \rceil$  iterações,  $\omega_{max} = \psi_{max} \cdot d_n$  slots de atividade e  $\tau_{max} = 2 \cdot k \cdot \psi_{max}$  slots de tempo total. Para os sensores periódicos, o desempenho depende dos valores de  $L_n$ ,  $d_n$  e  $k$ :

- Se  $\text{mdc}(\frac{L_n}{2}, k) \leq \frac{d_n}{2} < \frac{L_n}{2} \bmod k$  então  $\psi_{max} = 1 + \lceil \log_2 \frac{2k}{d_n} \rceil$ ,  $\omega_{max} = \psi_{max} \cdot d_n$  e  $\tau_{max} = \frac{2k}{d_n} \cdot L_n \cdot \psi_{max}$
- Se  $\frac{L_n}{2} \bmod k \leq \frac{d_n}{2}$  então  $\psi_{max} = 1 + \lceil \log_2 \frac{2k}{d_n} \rceil$ ,  $\omega_{max} = \psi_{max} \cdot d_n$  e  $\tau_{max} = \lceil \frac{2k}{d_n} \rceil \cdot L_n \cdot \psi_{max}$
- Se  $\frac{d_n}{2} = k$  então  $\psi_{max} = 1$  e  $\tau = \omega_{max} = d_n$

Os resultados obtidos em [Barsi et al. 2011] mostram que sensores livres são treinados em menor tempo, porém gastam mais energia. Em termos de consumo energético, o desempenho da busca binária é melhor se comparado ao *Flat+*. Entretanto o tempo total gasto é maior. Uma causa para isto é a transmissão dobrada dos *beacons*, exigida pela definição do comportamento do ator com a transmissão de *beacons control*.

## RESUMO

Este capítulo descreveu detalhadamente os protocolos de treinamento propostos em [Barsi et al. 2009, Barsi et al. 2011], tomados como base para este trabalho. *Flat+* e *Binary Training* são protocolos assíncronos de treinamento de localização *range-free* para

RSCA. Os nós se localizam em relação ao ator apenas escutando *beacons* transmitidos pelo mesmo. O protocolo *Binary Training* é mais eficiente, em termos de consumo energético, se comparado ao protocolo *Flat+*. Os trabalhos consideram apenas o treinamento dos sensores em coroas. Uma consideração importante é que o desempenho dos protocolos foi avaliado sob a condição de um canal de comunicação ideal, sem perda de *beacons*. Essa informação serve como ponto de partida para as avaliações de desempenho detalhadas no Capítulo 5.

# ANÁLISE EXPERIMENTAL DOS PROTOCOLOS FLAT+ E BINARY TRAINING

Em [Barsi et al. 2009] e em [Barsi et al. 2011] são realizadas análises de desempenho, respectivamente, dos protocolos *Flat+* e *Binary Training*. Entretanto, os trabalhos consideram canal de comunicação ideal, com transmissão isotrópica e sem perda de pacotes. Tais considerações não condizem com condições reais de funcionamento. Dessa forma, é necessária a avaliação dos protocolos considerando cenários mais realistas. Este capítulo descreve uma análise de desempenho dos protocolos *Flat+* e *Binary Training* sob condições mais realistas. Para isso, é modelado um canal de comunicação com desvanecimento, onde variações na força do sinal recebido causam erros de codificação e consequente perda de pacotes.

### 5.1 MODELO DE SIMULAÇÃO

Para avaliar o desempenho dos protocolos estudados, foi desenvolvido em *C++* um simulador para representar a interação entre um dispositivo sensor e um ator. É modelado o fenômeno de desvanecimento a fim de representar as imperfeições do canal, como será descrito na Seção 5.1.2.

#### 5.1.1 Descrição do Simulador

O simulador desenvolvido em linguagem *C++* utiliza duas classes principais. A primeira implementa o comportamento do ator, como descrito nas Seções 4.1 e 4.2. A outra classe implementa os algoritmos definidos pelos protocolos de treinamento estudados. Como os protocolos estudados dispensam a troca de mensagens entre os sensores, o treinamento pode ser realizado de maneira independente, não havendo distinção entre os

resultados da simulação do treinamento de  $n$  sensores distribuídos aleatoriamente e da simulação repetida  $n$  vezes de sensores posicionados aleatoriamente. A validação do simulador foi realizada através da reprodução dos resultados obtidos em [Barsi et al. 2009] e [Barsi et al. 2011].

A geração de números pseudoaleatórios e funções matemáticas mais complexas foram providos pela biblioteca *GNU Scientific Library* (GSL). Esses recursos são necessários na simulação das imperfeições do canal [Galassi et al. 2009].

### 5.1.2 Modelagem do Canal de Comunicação

Segundo modelos teóricos e medições reais para comunicação sem fio, a força do sinal decresce de maneira logarítmica com a distância entre o transmissor e o receptor [Rappaport 2001]. Tal perda é chamada de atenuação ou *path loss*. *Log Distance Path Loss* [Mao et al. 2007a, Wang et al. 2009] é um modelo matemático largamente utilizado na literatura para representação do fenômeno de atenuação. A força do sinal recebido é calculada pela equação

$$P_r = P_t - P_L(D_0) - 10 \cdot \alpha \cdot \log \left( \frac{D}{D_0} \right) \quad , \quad (5.1)$$

onde  $P_r$  e  $P_t$  são, respectivamente, a potência de recepção e de transmissão, medidas em decibéis,  $D$  é a distância transmissor/receptor em metros.  $P_L(D_0)$ , também em decibéis, é a perda de sinal percebida a uma distância de referência  $D_0$ . O parâmetro  $\alpha$  representa um expoente de atenuação cujo valores serão discutidos mais adiante.

Em situações reais, a força do sinal recebido sofre variações por conta do meio, fenômeno chamado de desvanecimento ou *fading*. A Figura 5.1 ilustra os efeitos do desvanecimento sobre a força do sinal recebido. Os efeitos do desvanecimento variam com a distância, tempo, frequência de transmissão e outras condições do ambiente. Geralmente, desvanecimento é modelado como um fenômeno aleatório [Rappaport 2001, Wang and Berger 2008]. Desta forma, o modelo *Log Distance Path Loss* é estendido pela adição de uma variável aleatória, como descreve a equação

$$P_r = P_t - P_L(D_0) - 10 \cdot \alpha \cdot \log \left( \frac{D}{D_0} \right) + X_\sigma \quad , \quad (5.2)$$

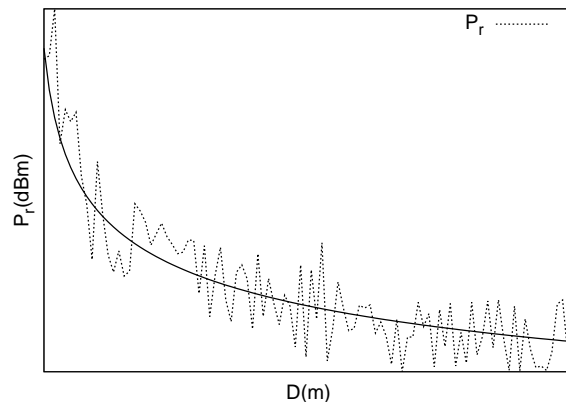


Figura 5.1: Efeitos do desvanecimento sobre a potência do sinal recebido. A linha contínua representa o valor da potência média recebida  $\overline{P_r}$ .

onde  $X_\sigma$  é uma variável aleatória de distribuição normal com média em zero e desvio padrão  $\sigma$  que representa uma variação na força do sinal recebido. O desvio  $\sigma$  depende das condições do meio.

Dada uma potência mínima  $\gamma$  necessária para o receptor decodificar o sinal, a probabilidade de recepção com sucesso é calculada segundo a probabilidade do sinal médio recebido exceder  $\gamma$ . Logo, a probabilidade de recepção do sinal,  $P_{suc}$ , é dada por

$$P_{suc} = Q\left(\frac{\gamma - \overline{P_r}}{\sigma}\right) \quad , \quad (5.3)$$

onde  $Q(x)$  é a função-Q da distribuição normal e  $\overline{P_r}$  é o valor médio da potência recebida, obtida da Equação 5.1, ou seja, removendo a variável aleatória  $X_\sigma$ .

### 5.1.3 Descrição dos Experimentos

Esta seção define valores para os parâmetros do modelo de propagação, define os cenários utilizados e lista as métricas consideradas neste trabalho. Os parâmetros foram escolhidos buscando se aproximar a transceptores realmente utilizados em dispositivos sensores. A escolha dos parâmetros também foi embasada em observações empíricas documentadas na literatura [Wang et al. 2004, Laselva et al. 2005, Mao et al. 2007a, Wang and Berger 2008].

**5.1.3.1 Parâmetros do Modelo** Neste trabalho são consideradas propriedades de transceptores *Chipcom CC2420* [Chipcom SmartRF<sup>©</sup> 2004]. Baseando-se nisso, é adotado um limiar de recepção  $\gamma = -90dBm$  e uma modulação OQPSK com codificação de 2 bits por símbolo. A taxa de erro de bit, *BER*, é calculada como  $ber = 1 - \sqrt{P_{suc}}$ . A taxa de erro de pacote, *PER*, é dada por  $ber = 1 - (1 - ber)^m$ , onde  $m$  é o tamanho do pacote. Os experimentos conduzidos assumem *beacons* de 128 bits e taxa de transmissão de 250Kbps, baseados na especificação *IEEE 802.15.4* [IEEE Standard 802.15.4 2001].

Este trabalho considera grandes quantidades de sensores dispersos em cenários externos abertos e sem obstruções de grande porte, tais como áreas rurais, campos e pequenas cidades. O modelo *log-distance path loss* adotado é amplamente utilizado para se estudar a propagação de sinais em ambientes externos [Wang et al. 2004, Laselva et al. 2005, Wang and Berger 2008]. O valor de  $\sigma$  nas avaliações deste trabalho é igual a 2dBm. A escolha deste valor é baseada em observações empíricas obtidas em [Wang et al. 2004, Laselva et al. 2005] para os tipos de cenários descritos.

Tabela 5.1: Valores do expoente de atenuação  $\alpha$ .

Ambiente	$\alpha$
Espaço livre	2
Área aberta	2.7 - 3.5
Área urbana	3 - 5
Construção com obstruções	4 - 6

A Tabela 5.1 contém alguns valores de  $\alpha$  normalmente utilizados [Rappaport 2001, Wang and Berger 2008], de acordo com cada tipo de cenário. De fato, existem trabalhos específicos no estudo dos valores de  $\alpha$  para RSSF [Mao et al. 2007a], onde resultados empíricos apresentam conformidade com os valores da Tabela 5.1. Desta forma, aqui é adotado expoente de atenuação  $\alpha = 3$ . A distância de referência  $D_0$  varia com transmissões de curta e larga escala. Neste trabalho é considerada  $D_0 = 1m$ . O valor  $P_L(D_0) = 20 \log \frac{4\pi}{\lambda}$  é obtido pela equação de *Friis* [Rappaport 2001].

**5.1.3.2 Descrição dos Cenários** São considerados sensores dispostos numa área circular com o ator posicionado ao centro. O espaço é dividido em 575 coroas disjuntas de raio fixo de 10m, ou seja  $k = 575$  níveis de potência. Nesse cenário, as coroas são ditas

disjuntas pois um sensor localizado na coroa  $C_i$  não recebe *beacons*  $b_j$ ,  $p_j < p_i$ , assim como considerado em [Barsi et al. 2009] e [Barsi et al. 2011].

A definição dos protocolos toma o tempo dividido em *slots*. A duração de cada *slot* é ajustada em  $2ms$ . Para cada valor de  $d_n$  (quantidade de *slots* onde o sensor escuta o canal de comunicação) são realizados  $N = 10.000$  treinamentos de sensores cujas posições são escolhidas aleatoriamente na área circular. Como os sensores são assíncronos, o instante de ativação deles também é escolhido aleatoriamente. Os valores de  $d_n$  estudados estão no conjunto  $4, 8, \dots, 52$ . O valor de  $L_n$  (duração do ciclo de trabalho em *slots*) é fixado em  $54$  *slots*.

É importante citar que os valores de  $k$ ,  $d_n$  e  $L_n$  aqui utilizados são os mesmos adotados em [Barsi et al. 2011], permitindo uma comparação direta com os resultados apresentados em tal trabalho. De fato,  $k = 575$  representa um valor de difícil compreensão para dispositivos reais. A escolha desse valor também é justificada pelo fato de que os problemas apresentados pelos protocolos de treinamento tornam-se mais evidentes com o crescimento de  $k$ . Esse fato foi constatado em nossos experimentos, porém tais resultados foram omitidos neste trabalho, uma vez que os resultados obtidos para  $k = 575$  são suficientemente conclusivos.

Tabela 5.2: Valores dos parâmetros de simulação.

Parâmetro	Valor
$\alpha$	3
$D_0$	$1m$
$\sigma$	$2dBm$
$\gamma$	$-90dBm$
<i>Beacon</i>	$128$ bits
<i>Slot</i>	$2ms$
$k$	575
Sensores	10000
$L_n$	54
$d_n$	$[4, 8, \dots, 52]$

A Tabela 5.2 resume os parâmetros utilizados nas simulações realizadas.

**5.1.3.3 Métricas Utilizadas** As métricas consideradas neste trabalho são definidas a seguir:

- **Erro ( $\Delta$ ):** A diferença entre a coroa estimada pelo sensor e a coroa real na qual o mesmo está posicionado;
- **Índice de Convergência ( $IC$ ):** a razão do número de sensores treinados (o algoritmo terminou normalmente) sobre o número total de sensores.
- **Índice de Treinamento ( $I_\Delta$ ):** a razão entre o número de sensores treinados com erro máximo de  $\Delta$  coroas sobre o número total de sensores;
- **Iterações ( $\psi$ ):** a quantidade de iterações necessárias para a convergência do algoritmo;
- **Tempo em Escuta ( $\omega$ ):** a quantidade de *slots* que o sensor permanece com o rádio em modo de escuta;
- **Tempo Total ( $\tau$ ):** a quantidade de *slots* decorridos desde a ativação do sensor até o final do treinamento.

Nos experimentos, as métricas são calculadas em função do tempo em escuta  $d_n$  e da posição do sensor  $C_n$ . Os intervalos de confiança foram calculados para os valores médios de  $\Delta$ ,  $\omega$  e  $\tau$  com níveis de confiança de 99%, o que nos permite estimar com precisão a média populacional para essas métricas. As barras que representam estes intervalos são algumas vezes imperceptíveis devido ao grande número de amostras coletadas (10000 sensores para cada  $d_n$ ).

## 5.2 DESEMPENHO DOS PROTOCOLOS

Para melhor apresentação foi adotada a convenção *BinF* para *Binary Training Free* e *BinP* para *Binary Training Periodic*, indicando, respectivamente, a versão do *Binary Training* para treinamento de sensores livres e periódicos.

A Figura 5.2(a) mostra a relação entre os problemas de convergência e a duração do tempo em escuta  $d_n$ . Se não tratados, os erros de convergência fazem o sensor realizar o treinamento por tempo indefinido, com conseqüente exaustão de energia. O protocolo *Binary Training*, para ambos tipos de sensores, apresenta Índice de Convergência próximo a 40% para  $d_n = 4$ . À medida que  $d_n$  aumenta, o Índice de Convergência decai, tendendo



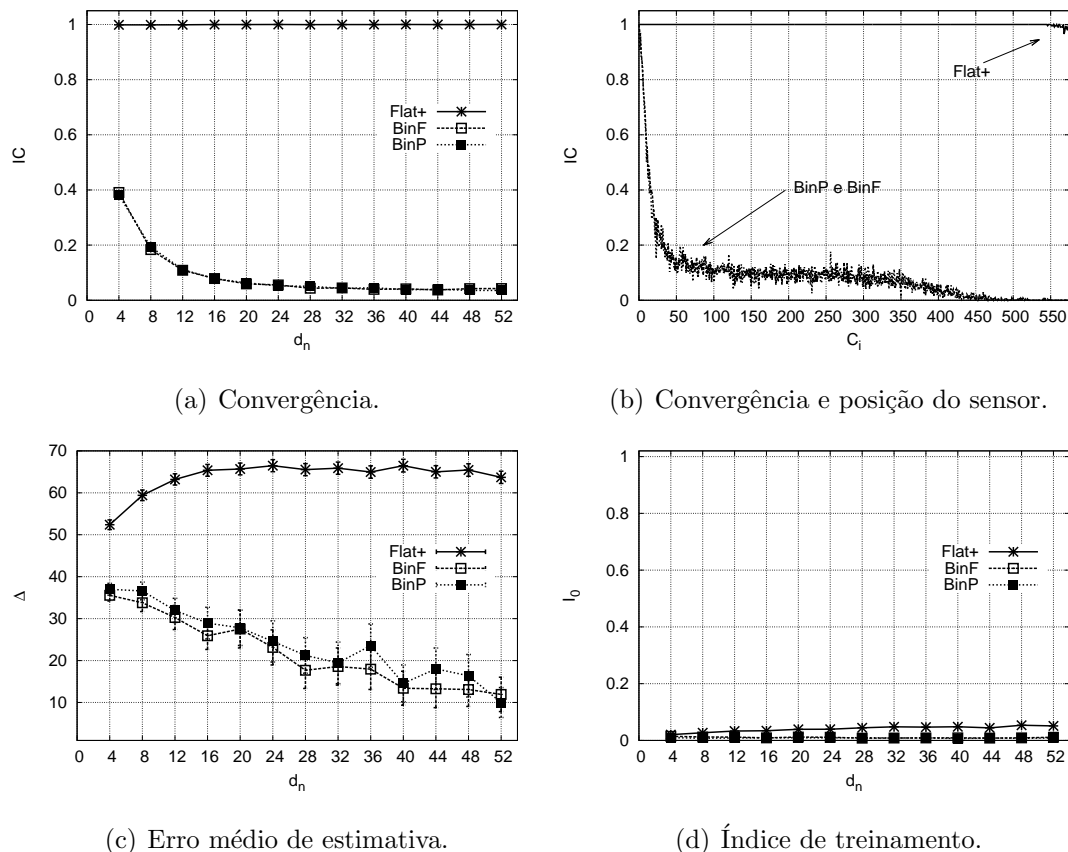


Figura 5.2: Convergência e precisão dos protocolos.

a valores próximos de 5%. Em outras palavras, a partir de  $d_n = 4$  aproximadamente 95% dos sensores executando *Binary Training* não completam o treinamento. Devido à perda de *beacons control*, o funcionamento do algoritmo *Binary Training* é prejudicado pela atualização indevida dos registradores *min* e *max* descritos na Seção 4.2, sobretudo com o aumento de  $d_n$ . Já o protocolo *Flat+*, apresenta um Índice de Convergência em torno de 99,98%. A Figura 5.2(b) mostra a relação entre o Índice de Convergência e o posicionamento real do sensor. Os problemas de convergência do *Flat+* ocorrem nas coroas mais externas ( $C_i \in [550, 574]$ ) devido ao alto *PER* experimentado pelos sensores lá posicionados. Um sensor localizado nas últimas coroas está quase no limite de transmissão do ator, ou seja, ele recebe menos *beacons*. Para o *Binary Training*, o Índice de Convergência cai rapidamente à medida que o sensor é posicionado mais distante do ator. Isso ocorre porque quanto maior for a distância entre o sensor e o ator, maior será a probabilidade do sensor perder *beacons control*.

A Figura 5.2(c) mostra como o Erro varia em função de  $d_n$ . Note que o *Flat+* obtém um erro superior a 50 coroas. Embora o *Binary Training* apresente um erro menor, esse resultado compreende apenas sensores cujo treinamento convergiu (vide Figura 5.2(a)). A Figura 5.2(d) mostra que a quantidade de sensores treinados corretamente é sempre inferior a 8%, para ambos os protocolos. *Flat+* treina mais sensores, porém o faz com erro considerável. *Binary Training* mostra-se inviável devido à baixíssima taxa de convergência, além do erro apresentado nas poucas vezes que o algoritmo converge.

### 5.2.1 Justificativas para os Erros Encontrados

Em ambientes reais não há garantia de um canal de comunicação perfeito [Rappaport 2001, Wang et al. 2009] e, do ponto de vista da aplicação, negligenciar essas perdas pode causar um mau funcionamento ou queda de desempenho [Lin et al. 2009, Wang et al. 2009]. No caso dos protocolos de treinamento, a perda de um *beacon* pode alterar significativamente o resultado, como mostram os resultados exibidos na Seção 5.2.

A Figura 5.3 mostra exemplos onde a perda de *beacons* causa erros de localização para os protocolos *Flat+* e *Binary Training*. A Figura 5.3(a) ilustra o funcionamento esperado do *Flat+*, com o sensor treinado na coroa  $C_6$ . Na Figura 5.3(b) a perda do *beacon*  $b_6$  faz com que o sensor encerre o treinamento e se localize na coroa  $C_7$ . Já na Figura 5.3(c), o erro é causado pela perda do *beacon*  $b_7$  e o sensor localizando-se na coroa  $C_8$ . A Figura 5.3(d) ilustra o funcionamento correto para o *Binary Training*, com o sensor treinado na coroa  $C_6$ . Na Figura 5.3(e) a perda do *beacon data*  $b_6$  faz com que o sensor se localize na coroa  $C_7$ . Já na Figura 5.3(f), a perda do *beacon control*  $b_1$  causaria uma situação inesperada no algoritmo, resultado em falhas de convergência.

Segundo [Barsi et al. 2011], a adição de *beacons control* serve apenas para tornar o treinamento resistente a desvios de relógio. Contudo, para os transceptores atualmente utilizados em sensores esse efeito é praticamente desprezível. Por exemplo, a especificação do rádio *CC2420* define um relógio de  $16MHz$ , cujo atraso de relógio fica na ordem dos picosegundos [Chipcom SmartRF<sup>©</sup> 2004]. De qualquer forma, o tratamento de desvio de relógio foge ao escopo deste trabalho.

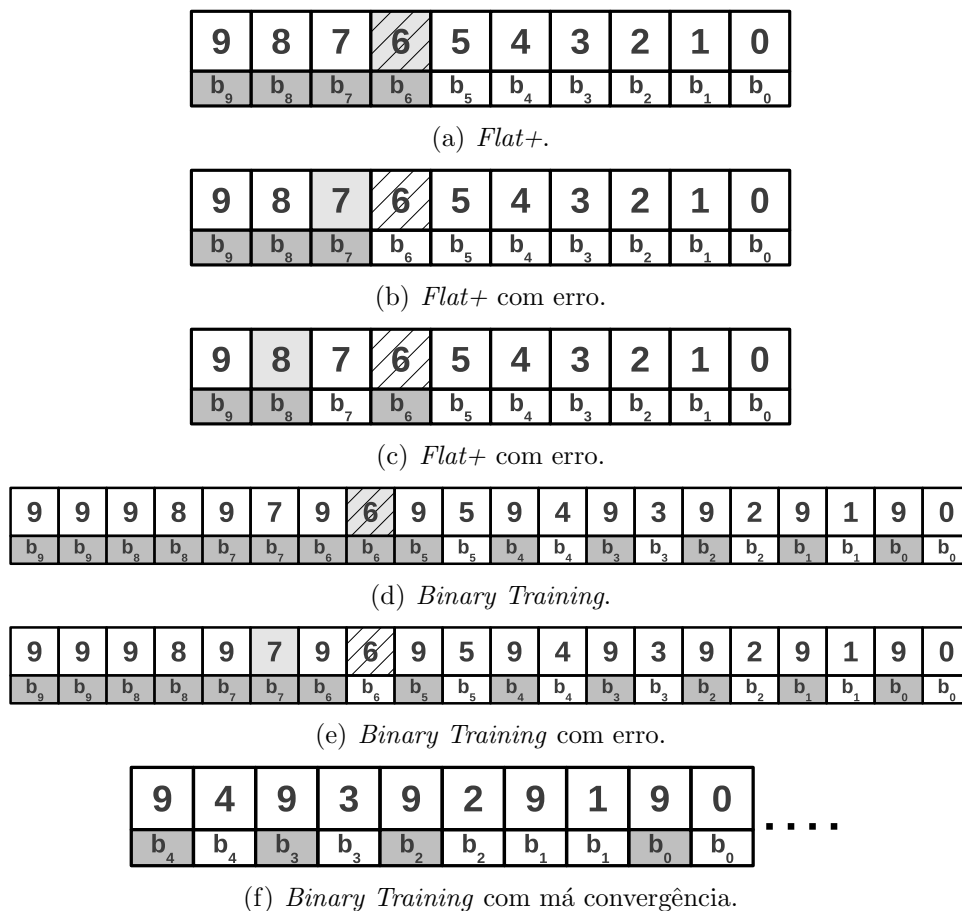


Figura 5.3: Exemplos de erros causados pela perda de *beacons*. Cada célula representa um *slot* (tempo fluindo da esquerda para direita). Acima está o nível de potência que *beacon* é transmitido. As células hachuradas abaixo simbolizam que o *beacon* foi escutado. O sensor está posicionado na coroa  $C_6$ .

## RESUMO

Este capítulo apresentou uma análise de desempenho dos protocolos *Flat+* e *Binary Training*. Através de um simulador propriamente desenvolvido e validado através da reprodução de experimentos, foram realizados experimentos para verificar o desempenho dos protocolos em condições mais realistas. Foi considerado um modelo de propagação com desvanecimento. Os resultados da simulação mostraram que os protocolos apresentam problemas quando considerada perda de *beacons*. O protocolo *Flat+* apresenta altos erros de estimativa de localização. O protocolo *Binary Training* enfrenta problemas de convergência devido à perda de *beacons control*. Pode-se concluir que o *Binary Training*

é inviável para cenários reais, enquanto *Flat+* inviabiliza uma série de aplicações devido à pouca precisão obtida pelo treinamento.

## CAPÍTULO 6

# PROTOCOLO STRONG FLAT

Um problema nos algoritmos descritos no Capítulo 5 está nas suposições sobre o recebimento dos pacotes. Em condições reais, a recepção de um *beacon* não é garantida mesmo o receptor estando dentro do alcance de transmissão. Para um melhor funcionamento, é necessário considerar probabilidades de recepção proporcionais ao nível de potência utilizado para transmissão. Logo, para verificar se o nó recebe ou não o *beacon* com certa garantia pode ser necessário obter mais de uma amostra. Esse é o princípio básico do protocolo proposto aqui, chamado *Strong Flat*. Este capítulo descreve detalhadamente o protocolo além de fazer uma breve análise sobre seu funcionamento.

### 6.1 DESCRIÇÃO DO ALGORITMO

O princípio do protocolo *Strong Flat* é armazenar informações sobre quantas vezes o *beacon*  $b_i$  foi recebido e quantas vezes o mesmo não foi escutado. Esse registro é feito através dos vetores  $H_k[ ]$  e  $M_k[ ]$ . Assim sendo,  $H_k[j] = x$  significa que o *beacon*  $b_j$  foi escutado  $x$  vezes, enquanto  $M_k[j] = y$  significa que o *beacon*  $b_j$  não foi escutado  $y$  vezes<sup>1</sup>. O algoritmo assume a recepção do *beacon*  $b_j$  apenas se  $H_k[j] \geq \beta_{hit}$  (*beacon* audível). Analogamente, o sensor assume que não recebe o *beacon*  $b_j$  se  $M_k[j] \geq \beta_{miss}$  (*beacon* inaudível). O algoritmo armazena dois registradores: *min* armazena o menor *beacon* audível e *max* armazena o maior *beacon* inaudível. Então, o sensor estará treinado e pertencerá à coroa  $C_i$  caso  $b_i$  seja audível e  $b_{i-1}$  seja inaudível. O comportamento do ator é o mesmo descrito para o protocolo *Flat+*, explicado na Seção 4.1.

O detalhamento do protocolo proposto é apresentado no Algoritmo 6.1. Na primeira recepção (condição na linha 8) é realizada a sincronização (linha 10) que permitirá ao sensor determinar os *beacons* transmitidos pelo ator. Em seguida, os *beacons* previamente

---

<sup>1</sup>Assim como nos demais protocolos, o sensor conhece o comportamento do ator, podendo determinar qual *beacon* está sendo transmitido naquele momento

**Algoritmo 6.1** Algoritmo Strong Flat.

---

```

1: heard ← false; trained ← false
2: min ←  $k - 1$ ; max ← 0;
3:  $v \leftarrow 0$ 
4: inicializa cada elemento  $H_k[\ ] \leftarrow 0$  e  $M_k[\ ] \leftarrow 0$ 
5: while  $\neg$ trained do
6:   for  $i \leftarrow 0$  to  $d_n - 1$  do do
7:     if recebeu  $b_c$  then
8:       if  $\neg$ heard then
9:         heard ← true
10:         $t \leftarrow k - 1 - c$ 
11:        for ciclo ← 0 to  $v - 1$  do
12:           $\forall b_j$  transmitido no ciclo,  $M_k[j] = M_k[j] + 1$ 
13:          for  $h \leftarrow 0$  to  $c - 1$  do
14:            if ( $M_k[h] \geq \beta_{miss}$ ) then
15:              max ←  $h$ 
16:              for  $h \leftarrow 0$  to max do
17:                 $M_k[h] = \beta_{miss}$ 
18:                 $H_k[c] \leftarrow H_k[c] + 1$ 
19:                if  $H_k[c] = \beta_{hit}$  then
20:                  for  $h \leftarrow c + 1$  to  $k - 1$  do
21:                     $H_k[h] \leftarrow \beta_{hit}$ 
22:                  if  $c < min$  then
23:                    min ←  $c$ 
24:                  if ( $min = 0$ ) or ( $M_k[min - 1] = \beta_{miss}$ ) then
25:                    trained ← true;
26:                    corona ←  $C_{min}$ 
27:                else
28:                  if heard then
29:                     $c \leftarrow t - 1 - t \bmod k$ 
30:                     $M_k[c] \leftarrow M_k[c] + 1$ 
31:                    if  $M_k[c] = \beta_{miss}$  then
32:                      for  $h \leftarrow max$  to  $c - 1$  do
33:                         $M_k[h] \leftarrow \beta_{miss}$ 
34:                      if  $c > max$  then
35:                        max ←  $c$ 
36:                      if  $H_k[max + 1] = \beta_{hit}$  then
37:                        trained ← true
38:                        corona ←  $C_{max+1}$ 
39:                    if  $\neg$ trained then
40:                      if  $\forall j \in [0; k], M_k[j] = \beta_{miss}$  then
41:                        trained ← true
42:                        corona ←  $C_{k-1}$ 
43:                    if  $\neg$ trained then
44:                      if  $\neg$ heard then
45:                        espera  $L_n - d_n$  slots
46:                      else
47:                        espera Wait() slots
48:                     $v \leftarrow v + 1$ 

```

---

transmitidos e não ouvidos são identificados (linhas 11-12) para que  $max$  e  $M_k[ ]$  sejam atualizados (linhas 13-17). A cada recepção do *beacon*  $b_c$  o elemento  $H_k[c]$  é atualizado (linha 18), enquanto o elemento  $M_k[c]$  é atualizado cada vez que *beacon*  $b_c$  não é ouvido (linhas 29-30). Quando é garantido um *beacon* audível (linha 19), os valores de  $H_k[ ]$  e  $min$  são atualizados (linhas 19-23). De forma análoga, quando um *beacon* é inaudível (linha 31), os valores de  $M_k[ ]$  e  $max$  são atualizados (linhas 31-35). Desta forma, a partir da primeira atualização de  $min$  e  $max$ , é garantido que  $H_k[min] = \beta_{hit}$  e  $M_k[max] = \beta_{miss}$ . O sensor estará treinado na coroa  $C_i$  se  $H_k[i] = \beta_{hit}$  e  $M_k[i-1] = \beta_{miss}$  (linhas 24-26 e 36-38). Ao fim de cada iteração, o sensor entra em inatividade segundo seu funcionamento próprio, representado pelo procedimento  $Wait()$  (linhas 44-47).

De fato, o protocolo proposto estende o *Flat+*. Logo, é necessário resolver os problemas de convergência relatados no Capítulo 5 para os sensores periódicos localizados nas coroas mais externas. Com este intuito, a cada ciclo é testado se todos os *beacons* foram considerados inaudíveis. Nesse caso, é considerada uma situação especial na qual o sensor é treinado na coroa mais externa  $k-1$  (linha 39-42).

---

**Algoritmo 6.2** Procedimento  $Wait()$  para o sensor livre.

---

```

guess =  $\lceil \frac{min+max}{2} \rceil$ 
if guess < c then
    wait  $\leftarrow c - guess - \frac{d_n}{2}$ 
else
    wait  $\leftarrow k + c - guess - \frac{d_n}{2}$ 
return max(0, wait)

```

---



---

**Algoritmo 6.3** Procedimento  $Wait()$  para o sensor periódico.

---

```

wait =  $L_n - d_n$ 
z = 0
while  $\forall b_i$  a ser transmitido,  $i \notin [min, max]$  do
    z  $\leftarrow z + 1$ 
return wait + z  $\cdot L_n$ 

```

---

O tempo de espera calculado pelo procedimento  $Wait()$  é determinado pelo tipo de sensor. Os algoritmos 4.4 e 4.5 descrevem o procedimento  $Wait()$ , respectivamente, para os sensores livre e periódico. Os sensores livres programam seu próximo ciclo para escuta do *beacon*  $b_{\lceil \frac{min+max}{2} \rceil}$ . Note que é utilizada busca binária para o sensor determinar o *beacon* de interesse programar sua ativação para recebê-lo exatamente no meio do período  $d_n$ . Já o sensor periódico ignora os ciclos onde todos os *beacons* a serem transmitidos são

audíveis ou inaudíveis (*beacons*  $b_i$ , com  $i \notin [max, min]$ ).

## 6.2 ANÁLISE TEÓRICA DE DESEMPENHO

Como nos algoritmo *Flat+*, o treinamento é possível se  $d \geq mdc(L_n, k)$ , com  $d_n$  par. Por depender de mais amostras, é necessário um maior tempo em escuta. O ajuste dos valores de  $\beta_{hit}$  e  $\beta_{miss}$  possibilita um equilíbrio entre tempo e precisão e será verificado na Seção 7.1.

São tomadas as seguintes considerações. Sejam  $k$ ,  $d_n$  e  $L_n$ , respectivamente, a quantidade de coroas, a duração do tempo em escuta e o tamanho do período de trabalho, como definidos na Seção 2.3. Seja  $p_{n,j}$  a probabilidade do sensor  $n$  receber o *beacon*  $b_j$ . Essa probabilidade depende da distância entre o sensor e o ator e varia com o tempo. É considerada uma probabilidade média por conta das variações temporais da força de sinal causadas por questões do meio de transmissão, como descrito na Seção 5.1.2. Para cada vez que um *beacon*  $b_j$  é testado, é obtida a informação que diz se o *beacon*  $b_j$  foi escutado ou não. Seja  $m_j$  a quantidade de vezes que um *beacon*  $b_j$  é testado. É definido como  $E(m_j)$  o valor esperado para a quantidade de testes (ver Apêndice A). Define-se  $\bar{E} = \frac{1}{k} \cdot \sum_{j=0}^k E(m_j)$  como a média geral da quantidade de testes realizados para todo *beacon*  $b_j$ .

Em termos de pior caso, a quantidade de iterações, o tempo em escuta e o tempo de treinamento, respectivamente representados por  $\psi_{max}$ ,  $\omega_{max}$  e  $\tau_{max}$ , são estimados como segue.

### 6.2.1 Sensor Periódico

Segundo o funcionamento do protocolo, o sensor ignora uma iteração quando detecta que todos os *beacons* a serem transmitidos na janela  $d_n$  já foram classificados como audíveis ou inaudíveis. O pior caso ocorre quando o sensor não ignora nenhuma iteração ( $d_n = L_n \bmod k$ ). A Figura 6.1 mostra um exemplo desse pior caso. O sensor, ativado durante a transmissão do *beacon*  $b_{15}$ , escutará os *beacons* ( $b_{15}, b_{14}, b_{13}, b_{12}, b_{11}, b_{10}, b_9, b_8, b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0$ ) nessa ordem. Um sensor localizado na coroa  $C_2$  só conseguirá concluir o treinamento após escutar a sequência aproximada-



mente  $\bar{E}$  vezes. Um padrão de escuta sequencial como esse impossibilita que o sensor ignore *beacons*. Generalizando, o pior caso do *Strong Flat* é semelhante ao pior caso do *Flat+* [Barsi et al. 2009]: se  $d_n = L_n \bmod k$ , um sensor que pertence à coroa  $C_i$  ativado durante a transmissão do *beacon*  $b_{i-1}$  só escutará a transmissão do *beacon*  $b_i$  após decorridas  $\frac{k}{L_n \bmod k}$  iterações.

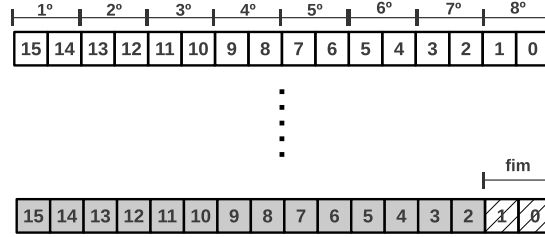


Figura 6.1: Exemplo do pior caso para sensor periódico *Strong Flat*, com  $k = 16$ ,  $d_n = 2$ ,  $L_n = 18$  e o sensor na coroa  $C_2$  ativado durante a transmissão do *beacon*  $b_{15}$ .

Então, para os sensores periódicos, a quantidade média de iterações,  $\bar{\psi}$ , depende da relação entre  $L_n$  e  $k$ . No pior caso, o sensor testará cada *beacon*, em  $\frac{k}{L_n \bmod k}$  blocos de  $L_n$  *slots*, por  $\bar{E}$  vezes. Em outras palavras,  $\bar{\psi} = \bar{E} \cdot \frac{k}{L_n \bmod k}$ . Consequentemente, o tempo médio de escuta, em *slots*, para este pior caso será  $\bar{\omega} = \bar{\psi} \cdot d_n$ . O tempo total em escuta será de  $\bar{\tau} = \bar{\psi} \cdot L_n$ .

### 6.2.2 Sensor Livre

Como o sensor controla seu ciclo de trabalho livremente, o procedimento de espera realiza busca binária para determinar o *beacon* de interesse. O pior caso assemelha-se com o pior caso obtido por uma busca binária em vetor.

A Figura 6.2 mostra um sensor localizado na coroa  $C_{k-1}$ , cuja ativação ocorreu durante a transmissão do *beacon*  $b_{\frac{k}{2}}$ , nesse caso o primeiro *beacon* de interesse. A partir da primeira recepção, o *beacon* de interesse  $b_{\frac{k}{2}}$  só será transmitido novamente após decorridos  $k$  *slots*. Após determinada a audibilidade do *beacon*  $b_{\frac{k}{2}}$ , o próximo *beacon* de interesse  $b_{\frac{k}{4}}$  será transmitido após  $k - \frac{k}{4}$  *slots*. Logo, a quantidade total de *slots* decorridos nessa primeira busca,  $\tau_1$  será de

$$\tau_1 = k \cdot [\bar{E} - 1] + k - \frac{k}{4} \quad , \quad (6.1)$$

com  $\bar{E}$  como definido anteriormente. O *beacon* de interesse passará a ser  $b_{\frac{k}{4}}$ . Analoga-

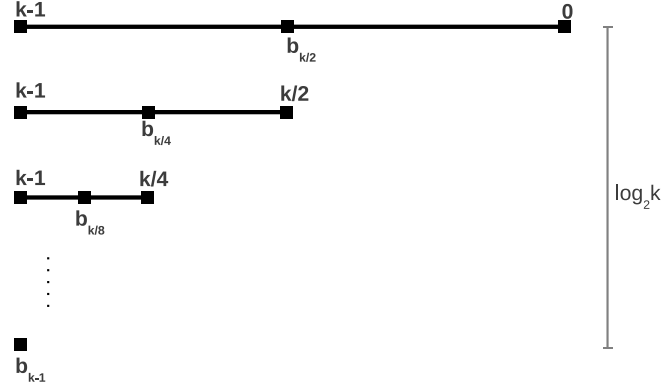


Figura 6.2: Exemplo do pior caso para sensor livre *Strong Flat*, um sensor localizado na coroa  $C_{k-1}$ .

mente, a quantidade total de *slots* decorridos para a determinação do *beacon*  $b_{\frac{k}{4}}$  e para a busca do *beacon*  $b_{\frac{k}{8}}$  será

$$\tau_2 = k \cdot [\bar{E} - 1] + k - \frac{k}{8} \quad , \quad (6.2)$$

e assim sucessivamente. No pior caso, são realizadas  $h = \lceil \log_2 k \rceil$  buscas. O tempo total decorrido, em *slots*, é determinado pelo somatório  $\tau = \sum_{i=1}^h \tau_i$ , definido como

$$\tau = \sum_{i=1}^h \left[ k \cdot (\bar{E} - 1) + k - \frac{k}{2^{i+1}} \right] \quad . \quad (6.3)$$

A Equação 6.3 pode ser simplificada agrupando os termos em  $k$  e removendo a constante do somatório. Procedendo assim, temos

$$\tau = k \cdot \left( h \cdot \bar{E} - \sum_{i=1}^h \frac{1}{2^{i+1}} \right) \quad . \quad (6.4)$$

A sequência  $\frac{1}{2^{i+1}}, i = \{1, 2, \dots, h\}$  representa uma progressão geométrica, com primeiro termo  $a_1 = \frac{1}{4}$  e razão  $q = \frac{1}{2}$ . O somatório na Equação 6.4 é a soma dos  $h = \lceil \log_2 k \rceil$  termos dessa progressão geométrica, ou seja:

$$\sum_{i=1}^h \frac{1}{2^{i+1}} = \frac{a_1 \cdot (q^h - 1)}{q - 1} = \frac{\frac{1}{4} \cdot (\frac{1}{2}^{\lceil \log_2 k \rceil} - 1)}{\frac{1}{2} - 1} = \frac{k - 1}{2k} \quad . \quad (6.5)$$

Por fim, agrupando os termos da Equação 6.4 e substituindo a expressão na

Equação 6.5, a equação pode ser mais uma vez simplificada para

$$\tau = k \cdot \lceil \log_2 k \rceil \cdot \bar{E} - \frac{k-1}{2k} \quad . \quad (6.6)$$

A quantidade de iterações é dada por  $\bar{\psi} = \bar{E} \cdot \lceil \log_2 k \rceil$ , diga-se a quantidade média de amostras para cada *beacon* de interesse vezes a quantidade de *beacons* de interesse utilizados durante a busca. Consequentemente, o tempo médio de escuta será  $\bar{\omega} = \bar{\psi} \cdot d_n$ . Já o tempo total em escuta  $\bar{\tau}$  é dado pela Equação 6.6.

## RESUMO

Este capítulo descreveu um novo protocolo assíncrono para treinamento de localização em RSCA, chamado *Strong Flat*. O protocolo proposto considera a perda de *beacons* durante o treinamento, testando *beacons* de interesse múltiplas vezes a fim de determinar se um *beacon* é ou não audível. O protocolo estende o *Flat+*, embora seja aplicável tanto a sensores periódicos quanto livres. O protocolo possui dois parâmetros de configuração,  $\beta_{hit}$  e  $\beta_{miss}$ , cujo ajuste influencia na precisão e, por consequência, no tempo em escuta. Este capítulo também definiu os valores estimados para quantidade de iterações, tempo em escuta e tempo de treinamento, considerando o pior caso para ambos tipos de sensores.

# ANÁLISE COMPARATIVA DOS PROTOCOLOS

Este capítulo descreve uma análise do protocolo proposto, focando na comparação com os protocolos *Flat+* e *Binary Training*. O capítulo descreve simulações realizadas segundo os mesmos cenários e métricas definidos na Seção 5.1.3. O protocolo proposto apresenta duas versões: *Strong Flat Free* treina sensores livres enquanto *Strong Flat Periodic* treina sensores periódicos. Além da convenção apresentada na Seção 5.2, foi adotada a notação *SFF* para *Strong Flat Free* e *SFP* para *Strong Flat Periodic*. Nesta seção também é apresentada uma análise teórica comparativa envolvendo o protocolo proposto e os demais estudados neste trabalho.

## 7.1 ANÁLISE DE DESEMPENHO DO PROTOCOLO STRONG FLAT

Esta seção verifica os efeitos dos valores de  $\beta_{hit}$  e  $\beta_{miss}$  no desempenho do protocolo *Strong Flat*. É avaliado o desempenho do protocolo para os pares  $(\beta_{hit}, \beta_{miss})$  iguais a  $(1, 1)$ ,  $(1, 2)$  e  $(1, 3)$ . Casos onde  $\beta_{hit} > 1$  são necessários quando temos sobreposição entre as coroas, ou seja, um sensor posicionado na coroa  $C_i$  pode receber *beacons*  $b_j$  com  $p_j < p_i$  e serão discutidos mais adiante.

Nas Figuras 7.1(a) e 7.1(b) é observado que os pares  $(1, 2)$  e  $(1, 3)$  obtêm um erro de localização significativamente menor em comparação ao obtido para o caso  $(1, 1)$ . Assim, testar apenas duas vezes se um *beacon* de interesse é inaudível, par  $(1, 2)$ , já permite obter melhorias significativas no erro de localização em comparação ao teste único, isto é, o par  $(1, 1)$ . Pode ser observada pouca melhoria no erro de localização ao se passar  $\beta_{hit}$  de 2 para 3. Em geral, os sensores livres apresentam menor erro de localização se comparados aos periódicos. As Figuras 7.1(c) e 7.1(d) mostram o impacto do valor de  $\beta_{miss}$  no tempo em escuta. Nesse caso, à medida que o valor de  $\beta_{miss}$  aumenta, o tempo em escuta aumenta. Observa-se também que os sensores livres apresentam um menor

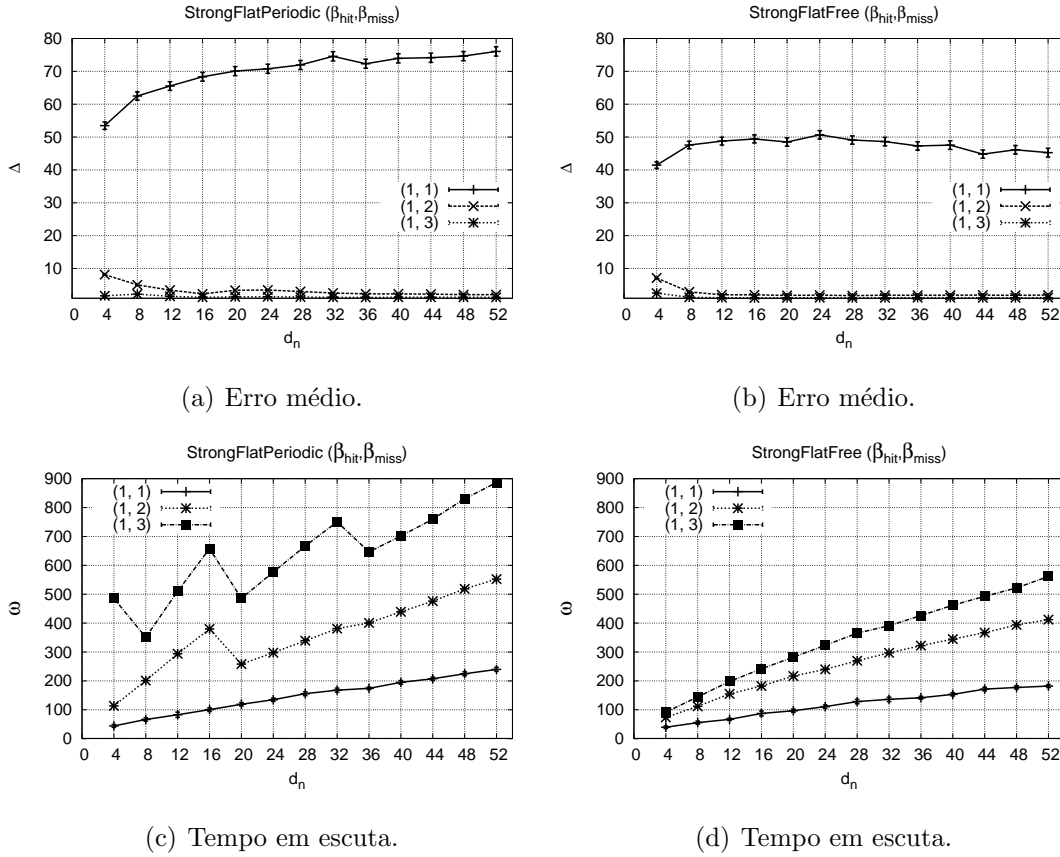


Figura 7.1: Efeito dos valores de  $\beta_{miss}$ .

crescimento no tempo em escuta em função do acréscimo de  $\beta_{miss}$ . Isso ocorre devido à capacidade que eles possuem de controlar dinamicamente o ciclo de trabalho.

As Figuras 7.2 mostram a relação entre o comportamento do tempo em escuta  $\omega$  e da quantidade de iterações  $\psi$  necessárias à convergência do algoritmo *Strong Flat* para sensores periódicos. O tempo em escuta total é proporcional ao tempo em escuta por ciclo  $d_n$  e à quantidade de iterações necessárias para a convergência do algoritmo. Por outro lado, a quantidade de iterações decresce com o aumento de  $d_n$ , como mostra a Figura 7.2(b). Nos valores de  $d_n$  onde  $\psi$  decai bruscamente,  $\omega$  também diminui. Por exemplo, o par (1, 3) apresenta quedas bruscas para  $d_n$  igual a 8, 20 e 36. Isso justifica o comportamento dente de serra nas curvas que representam os tempos em escuta para os pares (1, 2) e (1, 3) dos sensores periódicos.

A Figura 7.3 mostra como índice de treinamento de sensores é afetado por  $\beta_{miss}$ . Nota-se que, em relação ao caso mais simples (1, 1), o aumento de  $\beta_{miss}$  permite uma melhoria

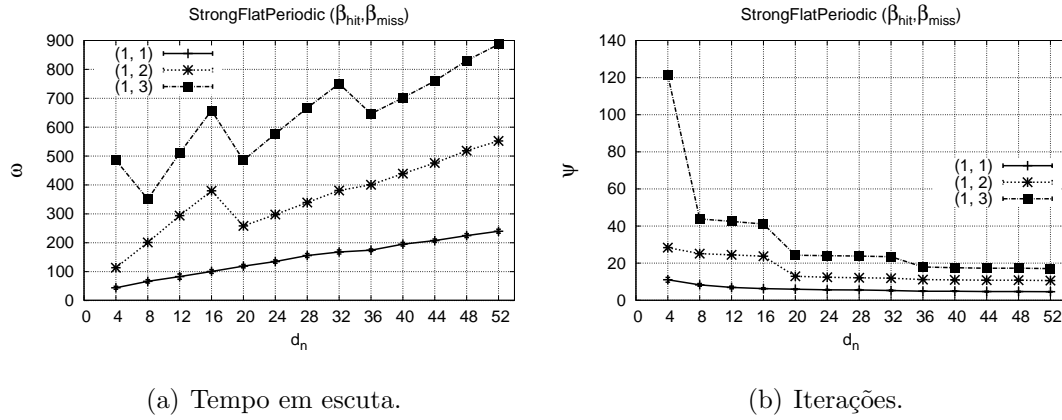


Figura 7.2: Análise do tempo em escuta do *Strong Flat Periodic*.

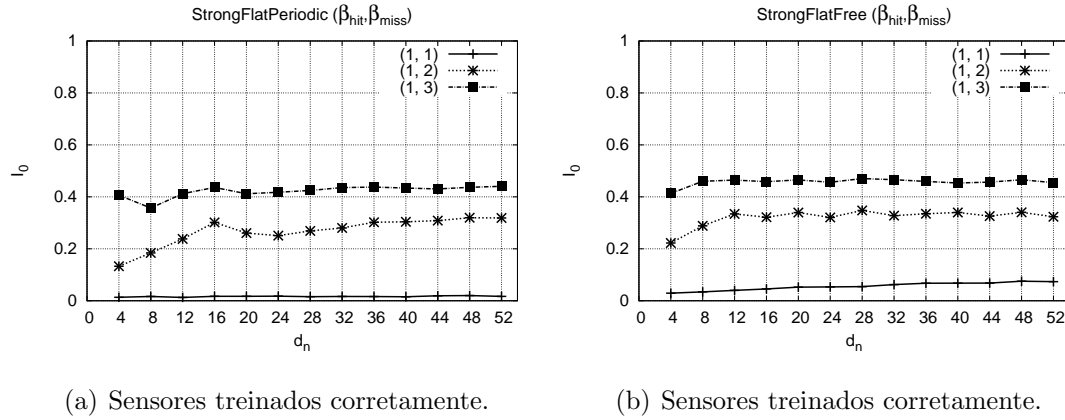


Figura 7.3: Efeito dos valores de  $\beta_{miss}$  sobre o índice de treinamento.

significativa na quantidade de sensores treinados corretamente na coroa em que estão fisicamente posicionados ( $I_0$ ). A precisão desejada no índice de treinamento depende da aplicação, ou seja, algumas aplicações podem tolerar um maior erro.

A Figura 7.4 mostra o índice de convergência obtido pelo protocolo proposto. As figuras mostram que não houve nenhum problema de convergência, isto é, todos os sensores, livres e periódicos, executando o *Strong Flat* concluíram o treinamento.

Nas Figuras 7.5(a) e 7.5(b) é verificada uma alta correlação, respectivamente, entre o tempo em escuta e o tempo de treinamento na comparação entre os protocolos. Pode ser observado que o par (1,1) para sensores periódicos resulta em comportamento similar ao protocolo *Flat+*. Desta forma, podemos utilizar aqui o par (1,1) como referência ao *Flat+*.

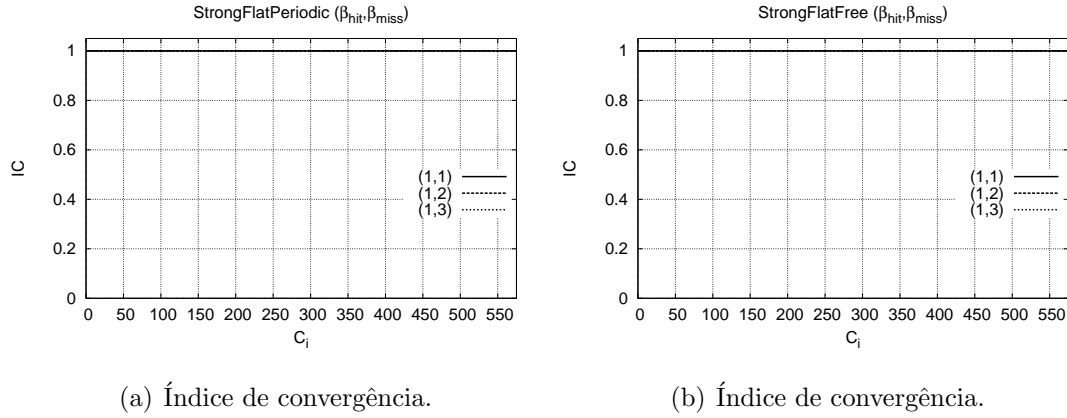


Figura 7.4: Efeito dos valores de  $\beta_{miss}$  sobre o índice de convergência.

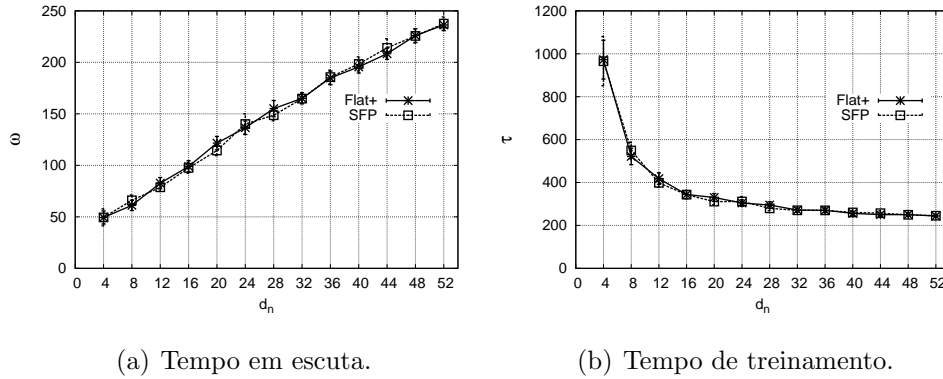


Figura 7.5: Comparação entre *Flat+* e *Strong Flat* (1, 1) para sensores periódicos.

Os resultados apresentados nesta seção também mostram que o par (1, 2) leva a um custo-benefício interessante em relação ao tempo de escuta e o erro de localização para ambos os tipos de sensores.

### 7.1.1 Cenário com $\beta_{hit} > 1$

Esta seção considera o caso onde um sensor localizado na coroa  $C_i$  recebe *beacons*  $b_j$  transmitidos para uma coroa  $C_j$ , com  $j < i$ , caracterizando coroas não disjuntas que se sobrepõem. Em outras palavras, o sensor pode receber *beacons* destinados a coroas mais internas<sup>1</sup>. Contudo, a probabilidade do sensor receber o *beacon*  $b_j$  é bem menor que a probabilidade de receber  $b_i$  ou qualquer outro de maior potência. Esse caso especial

<sup>1</sup>Essa propriedade será chamada de resíduo, por efeito de simplificação de notação.

abrange cenários onde existem obstáculos no caminho transmissor/receptor, como construções, ou quando considerado um alcance de transmissão que superestima o limiar de recepção do sensores.

Nesse cenário de simulação, foram admitidas coroas de raio fixado em 10 metros, sensores com limiar de recepção em  $-90dBm$  e a potência de transmissão do ator ajustada de modo a garantir que o sensor receba sinal médio de  $-88dBm$ . Dessa forma, um sensor localizado além dos 10 metros que delimitam uma coroa tem condições de receber o sinal transmitido para a coroa adjacente mais interna.

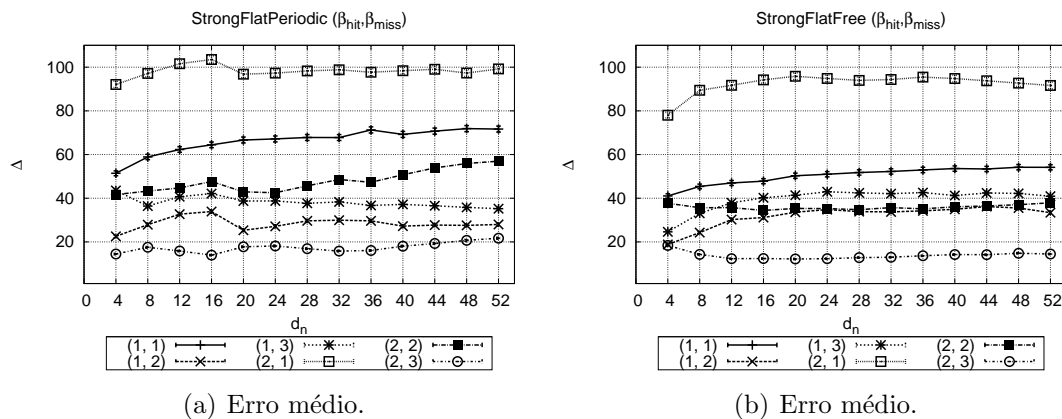


Figura 7.6: Efeito de  $\beta_{hit}$  e  $\beta_{miss}$ , com resíduo.

A Figura 7.6 mostra o erro obtido para o treinamento no cenário com resíduo. Note que o erro apresenta um aumento, se compararmos com o resultado para os pares da Seção 7.1, (1, 1), (1, 2) e (1, 3). Para ambos tipos de sensores, o menor erro foi obtido pelo par (2, 3). Para este cenário, o acréscimo de  $\beta_{hit}$  influencia significativamente no resultado. Por exemplo, entre (1, 3) e (2, 3) o erro médio é reduzido em até 20 coroas.

Note que o efeito da redução do erro é dado pela combinação de  $\beta_{hit}$  e  $\beta_{miss}$ . Ainda na Figura 7.6, se tomada como exemplo a comparação entre os pares (1, 2) e (2, 2), o acréscimo de  $\beta_{hit}$  não resulta na redução do erro. O mesmo ocorre com o par (2, 1). Nesses casos, isso ocorre porque o incremento de  $\beta_{hit}$  aumenta as chances do sensor ser treinado numa coroa mais externa, visto que a probabilidade de recepção de um *beacon* diminui com a distância mas aumenta com a potência (vide modelo de propagação, Seção 5.1.2).

Por efeito de comparação, a Figura 7.7 mostra o cenário sem resíduo, exibidos na Seção 7.1, com a adição dos pares (2, 1), (2, 2) e (2, 3). Quando é garantido que nenhum



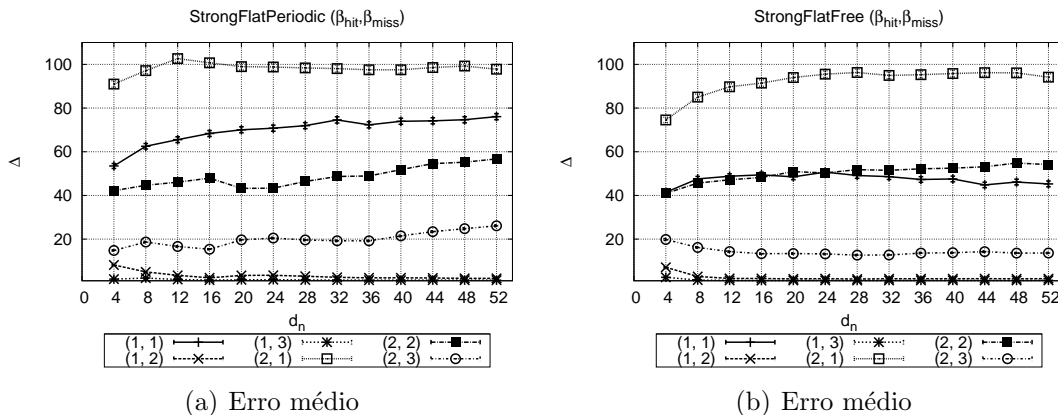


Figura 7.7: Efeito de  $\beta_{hit}$  e  $\beta_{miss}$ , sem resíduo.

sensor receberá *beacons* destinados as coroas mais internas, o uso de  $\beta_{hit} = 2$  não influencia positivamente no erro de localização, para ambos tipos de sensores. Os pares (1, 2) e (1, 3) ainda apresentam os menores erros, se comparados aos demais. O par (2, 3) é menos preciso que (1, 2) e (1, 3), embora obtenha menor erros que os pares (2, 1) e (2, 2) (cujo motivo é o mesmo explicado no parágrafo anterior).

## 7.2 COMPARAÇÃO ENTRE OS PROTOCOLOS ESTUDADOS

Esta seção realiza uma comparação entre o protocolo proposto e os demais protocolos estudados neste trabalho. A análise realizada nesta seção verifica e compara o desempenho dos protocolos *Flat+*, *Binary Training* e *Strong Flat* segundo três aspectos: análise teórica de desempenho, resultados de simulação e análise de complexidade dos algoritmos.

### 7.2.1 Análise Teórica de Desempenho

Esta seção compara o desempenho dos protocolos segundo seus piores casos como descritos nas Seções 4.1, 4.2 e 6.2.

A Tabela 7.1 faz uma comparação sobre o desempenho dos protocolos. É importante citar que a análise feita para os protocolos *Flat+* e *Binary Training* não considera imperfeições nos canal de comunicação. Como demonstrado, *Strong Flat* depende do número médio de amostras  $\bar{E}$  para cada *beacon*. Tais fatos limitam as possibilidades de comparações envolvendo os protocolos estudados aos casos onde não é observada perda de

Tabela 7.1: Comparação entre os protocolos estudados.

Protocolo		<i>Flat+</i>	<i>Binary Training</i>	<i>Strong Flat</i>
Livre	$\psi$	-	$1 + \lceil \log_2 k \rceil$	$\bar{E} \cdot \lceil \log_2 k \rceil$
	$\omega$	-	$d_n \cdot \psi$	$d_n \cdot \psi$
	$\tau$	-	$2k \cdot \psi$	$k \cdot \psi - \frac{k-1}{2k}$
Periódico	$\psi$	$\frac{k}{L_n \bmod k}$	$1 + \lceil \log_2 \frac{2k}{d_n} \rceil$	$\bar{E} \cdot \frac{k}{L_n \bmod k}$
	$\omega$	$d_n \cdot \psi$	$d_n \cdot \psi$	$d_n \cdot \psi$
	$\tau$	$L_n \cdot \psi$	$\frac{2k}{d_n} \cdot L_n \cdot \psi$	$L_n \cdot \psi$

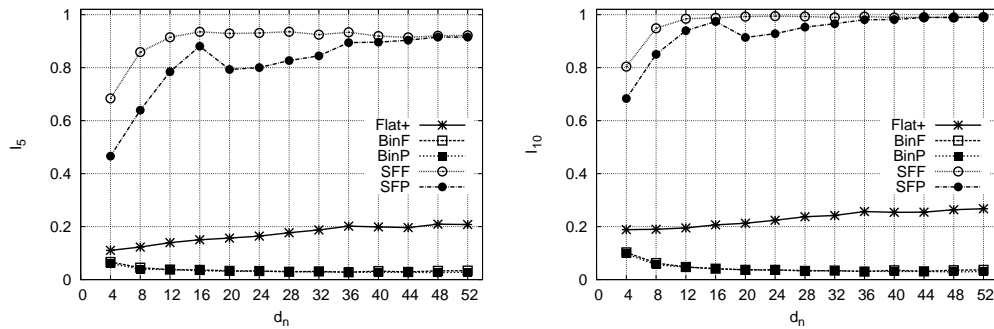
pacotes. Assim sendo, podem ser feitas certas observações para o caso sem perdas, com  $\bar{E} = 1$ , obtido quando  $\beta_{hit} = 1$  e  $\beta_{miss} = 1$ :

- Não há diferença significativa entre o *Strong Flat* e os demais protocolos em termos de iterações  $\psi$  e tempo de escuta  $\omega$ . De fato, o protocolo *Strong Flat* assemelha-se ao protocolo *Flat+* para os sensores periódicos e assemelha-se ao *Binary Training* quando considerados sensores livres;
- O tempo de convergência do *Strong Flat*, no pior caso, nunca ultrapassa o limite estipulado pelos demais protocolos, tanto para sensores livres quanto para periódicos;
- Em especial, o *Strong Flat* só é superado pelo *Binary Training* em termos de  $\psi$  e  $\omega$ , e isso quando considerados sensores periódicos. Nesse caso, porém, o tempo de convergência do *Binary Training* é superior ao dobro do *Strong Flat*.

### 7.2.2 Avaliações Experimentais

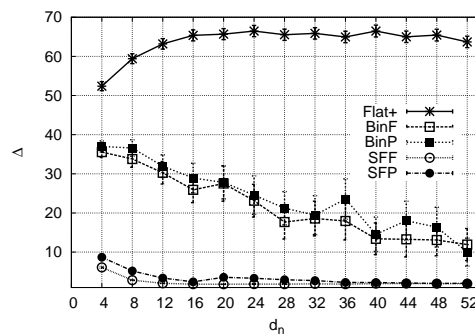
Esta seção apresenta o resultado de simulações realizadas com o intuito de comparar o desempenho entre o protocolo proposto e os demais protocolos estudados neste trabalho. É considerado o cenário de simulação sem resíduo. Como justificado na Seção 7.1, para os protocolos *Strong Flat* são utilizados  $\beta_{hit} = 1$  e  $\beta_{miss} = 2$  nas comparações seguintes.

As Figuras 7.8(a) e 7.8(b) mostram a quantidade de sensores treinados com erro, respectivamente, inferior a 5 e a 10 coroas. Para  $d_n > 12$ , mais de 80% dos sensores são treinados pelo protocolo *Strong Flat* com erro inferior a 5 coroas. Isso vale para ambos tipos de sensores. Como mostra a Figura 7.8(b), *Flat+* e *Binary Training* não treinam nem 30% dos sensores com erro inferior a 10 coroas. *Strong Flat*, por sua vez, o faz para



(a) Erro inferior a 5 coroas.

(b) Erro inferior a 10 coroas.



(c) Erro médio de estimativa.

Figura 7.8: Precisão dos protocolos.

mais de 80% dos sensores. Em outras palavras, o protocolo proposto garante que ao menos 80% dos sensores num raio de 5750 metros sejam localizados com erro máximo de 100 metros, sobretudo quando  $d_n > 8$ . Na Figura 7.8(c) é exibida uma comparação entre os erros médios de estimativa obtidos pelos protocolos de treinamento. *Strong Flat* reduz o erro em até 92,8% se comparado ao *Flat+*. O erro obtido pelo *Binary Training* é maior se comparado ao protocolo proposto, mesmo representando apenas 5% do total de amostras como discutido na Seção 5.2.

As Figuras 7.9(a) e 7.9(b) mostram, respectivamente, o tempo gasto para o treinamento e o tempo em escuta. Observa-se que o tempo total de treinamento gasto com o protocolo proposto é maior do que o observado para os demais protocolos na maioria dos casos. Mais uma vez, esse comportamento é esperado dado que o *Strong Flat* testa mais de uma vez se um *beacon* de interesse é audível.

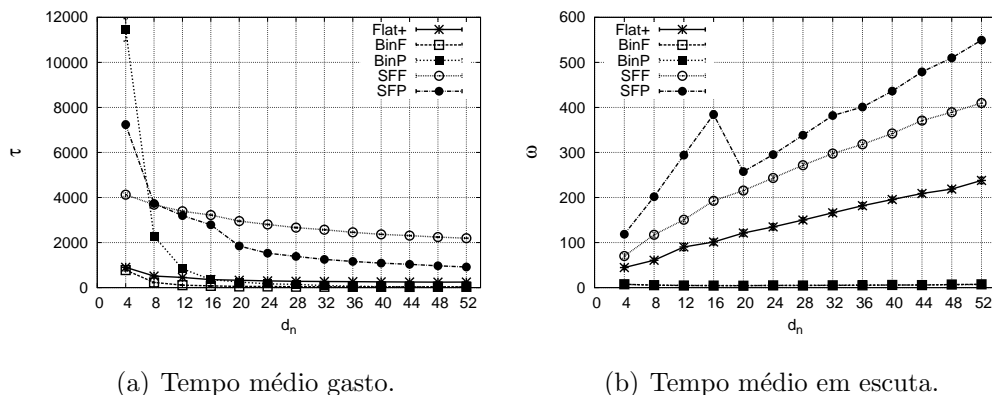


Figura 7.9: Desempenho dos protocolos.

### 7.2.3 Análise de Complexidade

As restrições dos dispositivos sensores exige um cuidado especial com a carga de processamento necessária para o funcionamento de um protocolo. A quantidade de instruções executadas pode ser estimada através da análise assintótica dos algoritmos [Cormen et al. 2009]. Dessa forma, a análise nos dará uma estimativa sobre a quantidade de instruções executadas pelo protocolo para uma única iteração, isto é, para um período  $d_n$ . Observando os algoritmos descritos nas Seções 4.1, 4.2 e 6.1, a análise assintótica revela o seguinte comportamento:

- *Flat+* é  $O(k)$ , devido à necessidade de atualização do vetor  $R_k[ ]$  (ver Algoritmo 4.1);
- *Binary Training* é  $O(1)$  para sensores livres (ver Algoritmo 4.3) e  $O(k)$  para sensores periódicos (devido ao procedimento *Wait()* descrito no Algoritmo 4.5);
- *Strong Flat* é  $O(k)$ , devido à atualização dos vetores  $H_k[ ]$  e  $M_k[ ]$ .

Tabela 7.2: Complexidade dos algoritmos estudados.

Protocolo	Sensor Livre		Sensor Periódico	
	Algoritmo	Treinamento	Algoritmo	Treinamento
<i>Flat+</i>	-	-	$O(k)$	$O(k^2)$
<i>Binary Training</i>	$O(1)$	$O(\log(k))$	$O(k)$	$O(k \cdot \log(k))$
<i>Strong Flat</i>	$O(k)$	$O(k \cdot \log(k))$	$O(k)$	$O(k^2)$

De fato, quanto maior a quantidade de iterações necessárias para o treinamento, mais instruções são realizadas pelo sensor para completar o processo de treinamento.

A Tabela 7.2 faz uma comparação da complexidade dos algoritmos. Na tabela é exibida a complexidade do algoritmo e do treinamento (com base na quantidade de iterações  $\psi$ , como mostra a Tabela 7.1). No geral, o pior caso para o treinamento é obtido para os sensores periódicos, com complexidade  $O(k^2)$ . Para os sensores livres, o pior caso obtido é  $O(k \cdot \log(k))$ . Para ambos tipos de sensores, *Binary Training* apresenta menor complexidade, refletindo numa menor quantidade de instruções executadas. Nota-se que o protocolo proposto *Strong Flat* apresenta comportamento similar ao *Flat+* para sensores periódicos. Para sensores livres, o protocolo proposto é menos eficiente se comparado ao *Binary Training*. De qualquer forma, todos protocolos de treinamento estudados possuem comportamento, no máximo, polinomial grau 2.

## RESUMO

Este capítulo descreveu uma análise comparativa envolvendo os três protocolos assíncronos de treinamento de localização estudados neste trabalho. Foi realizada uma análise de desempenho envolvendo o protocolo proposto. Os resultados indicaram que *Strong Flat* não apresenta problemas de convergência. Além disto, um valor  $\beta_{miss} = 2$  é o suficiente para aumentar substancialmente a precisão do protocolo. Um caso onde  $\beta_{hit} > 1$  também foi analisado, resultando em melhor precisão quando consideradas co-roas com sobreposição. Uma análise comparativa foi realizada envolvendo o protocolo proposto e os dois protocolos estudados neste trabalho. Tal análise compreendeu três partes distintas. A comparação teórica mostrou que o protocolo proposto não é menos eficiente se comparado ao demais, considerado o caso sem perda de pacotes. Os resultados experimentais mostraram que o protocolo proposto treina mais sensores e com menor erro, se comparado ao *Flat+* e ao *Binary Training*. Através da análise de complexidade, ficou constatado que o novo protocolo não adiciona muito processamento, em relação aos demais protocolos.

# CONCLUSÕES

Localização em redes de sensores sem fio é um problema bastante abordado na literatura por tratar-se de um serviço importante, uma vez que a ausência de informações sobre a localização dos dados coletados pode inviabilizar algumas aplicações. Devido às restrições dos dispositivos sensores, os protocolos de localização devem considerar as limitações de *hardware* e comunicação dos dispositivos, além de atender aos requisitos da aplicação. Desta forma, são necessários protocolos de localização para RSSF específicos para a plataforma. Diversas propostas de protocolos para localização já foram elaboradas, sendo os protocolos classificados como *range-based* e *range-free*. Recentemente vêm sendo estudados protocolos *range-free* para treinamento de localização em redes de sensores, cuja estimativa da posição do nó sensor é realizada apenas através da escuta de *beacons* transmitidos por um dispositivo mais robusto, em geral de localização conhecida.

Este trabalho abordou redes de sensores e atores, onde os sensores são organizados em sub-redes ativadas sob a presença de um ator. Redes deste tipo são chamadas de redes de sensores centradas em ator, RSCA. O principal problema abordado aqui é possibilitar que os sensores estimem sua posição em relação a um ator. Esse processo de treinamento considera um sistema de referência polar discreto que divide o espaço em coroas concêntricas e setores circulares, com o ator como centro do sistema de referência.

Na literatura podem ser encontrados protocolos para treinamento em RSCA. Este trabalho estudou dois protocolos existentes na literatura, *Flat+* e *Binary Training*, concebidos para o treinamento em redes de sensores centradas em ator heterogêneas de larga escala. Esses protocolos dispensam sincronização entre os dispositivos e são capazes de treinar sensores com distintos ciclos de trabalho, alcançando alta escalabilidade. Entretanto, os protocolos foram desenvolvidos sob a consideração de um canal de comunicação ideal sem perda de pacotes, fato improvável em cenários reais. Por isso, fez-se necessária uma avaliação acerca do comportamento dos protocolos sob condições mais realistas.

Com o intuito de verificar o desempenho sob condições de perda de pacotes, foi modelado um canal de comunicação com desvanecimento, através do modelo de propagação *Log Distance Path Loss*. Com a realização de análise de desempenho, foi constatada a ineficiência dos protocolos estudados. Os resultados das simulações mostraram que o uso dos protocolos em tais condições acarreta em significativo erro de localização e problemas de convergência que impossibilitam o fim do treinamento. Os problemas de convergência, se não tratados, tendem à exaustão de recursos do dispositivo sensor. Erros de localização adicionam imprecisão às medidas coletadas, prejudicando as aplicações. Logo, em cenários com canal de comunicação não ideal, o uso dos protocolos *Flat+* e *Binary Training* mostrou-se inviável.

Com o intuito de possibilitar o treinamento em cenários mais realistas, este trabalho propôs um novo protocolo de treinamento assíncrono para redes heterogêneas de sensores e atores, chamado *Strong Flat*. O protocolo proposto estende o *Flat+*, porém considera a possibilidade da perda de *beacons* durante o processo de treinamento. O protocolo realiza a escuta dos *beacon* de interesse mais de uma vez, a fim de obter maior precisão na localização. O protocolo também permite a configuração do nível de precisão obtido, através do ajuste de dois parâmetros  $\beta_{hit}$  e  $\beta_{miss}$  que influenciam no tempo de escuta.

O protocolo proposto foi comparado aos demais protocolos de treinamento de localização, segundo estudo teórico e através de simulações. Os resultados das simulações mostraram uma considerável redução nos erros de localização, alcançando até 92,8% de redução sobre o erro de localização. O protocolo proposto não apresenta problemas de convergência e treina mais sensores que os demais estudados. Fixado  $\beta_{hit}$ , o ajuste de  $\beta_{miss}$  permite uma maior precisão ao custo de um maior consumo energético. O ajuste de  $\beta_{hit}$  permite o treinamento mais preciso em cenários onde não temos coroas disjuntas. Através da análise teórica foi constatado que o pior caso de desempenho do protocolo proposto assemelha-se àqueles obtidos pelos outros dois protocolos estudados. Em resumo, o protocolo proposto mostrou-se mais viável em cenários com perda de pacotes do que os protocolos *Flat+* e *Binary Training*.

Como trabalhos futuros, podem ser listados os seguintes pontos:

- Estudo mais detalhado do impacto dos demais parâmetros do modelo e do sistema, como  $\alpha$ ,  $\sigma$ ,  $L_n$ , entre outros;

- Análise teórica sobre o comportamento do protocolo através do uso de um modelo probabilístico. Tal modelo possibilitaria uma análise mais geral sobre a estimativa do erro de localização;
- Um estudo considerando a existência de mais de um ator realizando treinamento simultaneamente. A presença de mais de um ator pode ser interessante a alguma aplicação específica, por exemplo. Assim sendo, a interferência causada por treinamentos simultâneos deve ser verificada;
- Experimentos reais para verificar de fato a viabilidade dos protocolos de treinamento de localização. Dado que este trabalho considerou simulações, modelos de propagação probabilísticos não conseguem representar perfeitamente os fenômenos que influenciam na comunicação por rádio.

Este trabalho gerou uma publicação no XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, promovido pela Sociedade Brasileira de Computação e realizado em Maio de 2012 [Jesus and Gonçalves 2012].



## APÊNDICE A

### CÁLCULO DA QUANTIDADE MÉDIA DE TESTES

Dado um *beacon*  $b_j$ , é necessária a realização de vários testes para determinar se o *beacon* é audível ou inaudível. Para cada vez que um *beacon*  $b_j$  é testado, é obtida a informação que diz se o *beacon*  $b_j$  foi escutado ou não. Esta seção descreve como será calculada a quantidade média de testes,  $E(m_j)$ , necessário para determinar se um *beacon*  $b_j$  pode ou não ser de fato escutado.

Primeiramente definimos um teste como uma resposta binária **sucesso** ou **falha**. Seja o conjunto de testes realizados para determinar se um *beacon* é audível ou inaudível definido como  $S = s_1, s_2, \dots, s_m$ , onde cada teste  $s_i$  assume o valor *sucesso* ou *falha*. Um conjunto de testes conclui **positivo** ao serem constatados  $\beta_{hit}$  testes com *sucesso*. Analogamente, um conjunto de testes conclui **negativo** ao serem constatados  $\beta_{miss}$  testes com *falha*. Por exemplo, dados  $\beta_{hit} = 3$  e  $\beta_{miss} = 4$ , podemos listar como conjuntos de testes:

- $S = sucesso, sucesso, sucesso$ , conclui **positivo** após 3 testes
- $S = falha, falha, falha, falha$ , conclui **negativo** após 4 testes
- $S = sucesso, sucesso, falha, sucesso$ , conclui **positivo** após 4 testes
- $S = falha, falha, sucesso, sucesso, sucesso$ , conclui **positivo** após 5 testes
- $S = falha, falha, sucesso, sucesso, falha, falha$ , conclui **negativo** após 6 testes

É desejado calcular a quantidade média de testes, em outras palavras, o tamanho médio do conjunto  $S$ . A quantidade mínima de testes é dada por  $m_{min} = \min(\beta_{hit}, \beta_{miss})$  e a quantidade máxima de testes é dada por  $m_{max} = \beta_{hit} + \beta_{miss} - 1$  (respectivamente, o primeiro e o último exemplo). A quantidade média de testes decorre da média dos valores

possíveis ponderada pela probabilidade de cada valor ocorrer. Logo

$$E(m_j) = \sum_{m=m_{min}}^{m_{max}} m \cdot P(m_j = m) \quad , \quad (\text{A.1})$$

onde  $P(m_j = m)$  é a probabilidade do conjunto  $S$  possuir  $m$  elementos. A probabilidade  $P(m_j = m)$  é obtida por duas situações distintas: concluir **positivo** ou concluir **negativo**.

Seja  $p$  a probabilidade de um teste resultar em *sucesso*, isto é, a probabilidade do sensor receber o *beacon* em questão<sup>1</sup>. Um conjunto de testes de tamanho  $m$  que conclui **positivo** é uma sequência formada da seguinte forma:

- O início da sequência é uma combinação qualquer de  $m - 1$  elementos  $\{sucesso, falha\}$  contendo exatamente  $\beta_{hit} - 1$  sucessos. Assim, são possíveis  $\binom{m-1}{\beta_{hit}-1}$  combinações;
- O último elemento é *sucesso*.

Sendo assim, se  $q_m$  é a probabilidade de um conjunto de  $m$  testes concluir **positivo**, então temos

$$q_m = \sum_{k=\beta_{hit}}^m \left[ \binom{m-1}{k-1} \cdot p^{\beta_{hit}-1} \cdot (1-p)^{k-\beta_{hit}} \right] \cdot p \quad . \quad (\text{A.2})$$

Analogamente, o padrão que define um conjunto de testes cujo resultado é **negativo** pode ser definido seguindo o mesmo princípio. Logo, se  $q'_m$  é a probabilidade de conjunto de  $m$  testes concluir **negativo**, então temos

$$q'_m = \sum_{k=\beta_{miss}}^m \left[ \binom{m-1}{k-1} \cdot (1-p)^{\beta_{miss}-1} \cdot p^{k-\beta_{miss}} \right] \cdot (1-p) \quad . \quad (\text{A.3})$$

Como para um conjunto de testes  $S$  a conclusão em **positivo** ou **negativo** configurar eventos mutuamente exclusivos, temos que

$$P(m_j = m) = q_m + q'_m \quad . \quad (\text{A.4})$$

---

<sup>1</sup>Formalizando,  $p = p_{n,j}$  é a probabilidade do sensor  $n$  receber o *beacon*  $b_j$ .

Por fim, através da substituição da Equação A.4 na Equação A.1, obtemos

$$E(m_j) = \sum_{m=\min(\beta_{hit}, \beta_{miss})}^{\beta_{hit} + \beta_{miss} - 1} m \cdot (q_m + q'_m) \quad , \quad (\text{A.5})$$

onde os valores de  $q_m$  e  $q'_m$  são obtidos, respectivamente, nas Equações A.2 e A.3.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [Akyildiz and Kasimoglu 2004] Akyildiz, I. F. and Kasimoglu, I. H. (2004). Wireless Sensor and Actor Networks: Research Challenges. *Ad Hoc Networks*, 2(4):351 – 367.
- [Akyildiz et al. 2002] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless Sensor Networks: A Survey. *Computer Networks*, 38(4):393–422.
- [Al-Dhalaan and Lambadaris 2010] Al-Dhalaan, A. and Lambadaris, I. (2010). Wireless sensor network localization with spatially correlated shadowing. In *Proceedings of IEEE International Conference on Communications, ICC*, pages 1–6. IEEE.
- [Ameer and Jacob 2010] Ameer, P. M. and Jacob, L. (2010). Localization Using Ray Tracing for Underwater Acoustic Sensor Networks. *Communication Letters*, 14(10):930–932.
- [Barsi et al. 2009] Barsi, F., Bertossi, A. A., Betti Sorbelli, F., Ciotti, R., Olariu, S., and Pinotti, M. C. (2009). Asynchronous Corona Training Protocols in Wireless Sensor and Actor Networks. *IEEE Transactions on Parallel and Distributed Systems*, 20(8):1216–1230.
- [Barsi et al. 2011] Barsi, F., Bertossi, A. A., Lavault, C., Navarra, A., Olariu, S., Pinotti, M. C., and Ravelomanana, V. (2011). Efficient Location Training Protocols for Heterogeneous Sensor and Actor Networks. *IEEE Transactions on Mobile Computing*, 10(3):377–391.
- [Boukerche 2008] Boukerche, A., editor (2008). *Algorithms and Protocols for Wireless Sensor Networks*. Wiley-IEEE Press.

- [Bulusu et al. 2000] Bulusu, N., Heidemann, J., and Estrin, D. (2000). Gps-less Low Cost Outdoor Localization for Very Small Devices. *IEEE Personal Communications Magazine*, 7(5):28–34.
- [Chang and Liao 2009] Chang, C.-H. and Liao, W. (2009). Revisiting Relative Location Estimation in Wireless Sensor Networks. In *Proceedings of the 2009 IEEE International Conference on Communications, ICC*, pages 42–46, Piscataway, NJ, USA. IEEE Press.
- [Chaurasia and Payal 2011] Chaurasia, S. and Payal, A. (2011). Analysis of Range-Based Localization Schemes in Wireless Sensor Networks: A Statistical Approach. In *Proceedings of 13th International Conference on Advanced Communication Technology, ICACT*, pages 190–195.
- [Chen et al. 2010] Chen, H., Chan, Y., Poor, H., and Sezaki, K. (2010). Range-Free Localization with the Radical Line. In *Proceedings of IEEE International Conference on Communications, ICC*, pages 1–5.
- [Cheng et al. 2007] Cheng, K.-Y., Lui, K.-S., and Tam, V. (2007). Localization in Sensor Networks with Limited Number of Anchors and Clustered Placement. In *Proceedings of Wireless Communications and Networking Conference, WCNC*, pages 4425–4429. IEEE.
- [Chipcom SmartRF<sup>©</sup> 2004] Chipcom SmartRF<sup>©</sup> (2004). CC2420 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver.
- [Cormen et al. 2009] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms*. The MIT Press, 3rd edition.
- [Domingo and Prior 2008] Domingo, M. C. and Prior, R. (2008). Energy Analysis of Routing Protocols for Underwater Wireless Sensor Networks. *Computer Communications*, 31(6):1227–1238.
- [Galassi et al. 2009] Galassi, M., Davies, J., Theiler, J., Gough, B., Jungman, G., Alken, P., Booth, M., and Rossi, F. (2009). *GNU Scientific Library Reference Manual*. 3rd edition.

- [Ghidini et al. 2010] Ghidini, G., Pinotti, C., and Das, S. (2010). A Semi-Distributed Localization Protocol for Wireless Sensor and Actor Networks. In *Proceedings of 8th IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 438–443.
- [He et al. 2005] He, T., Huang, C., Blum, B. M., Stankovic, J. A., and Abdelzaher, T. F. (2005). Range-free Localization and its Impact on Large Scale Sensor Networks. *ACM Transactions Embedded Computing Systems*, 4(4):877–906.
- [Huang et al. 2010] Huang, B., Yu, C., and Anderson, B. D. (2010). On the Rate of Error Propagation in Multihop Range-Based Localization. In *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing, ICASSP*, pages 2850–2853.
- [IEEE Standard 802.15.4 2001] IEEE Standard 802.15.4 (2001). *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*. IEEE Standard.
- [Jesus and Gonçalves 2012] Jesus, J. S. and Gonçalves, P. A. S. (2012). Um Protocolo de Localização Assíncrono para RSSF Heterogêneas Centradas em Ator. In *Anais do XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, SBRC, Ouro Preto, MG.
- [Ke et al. 2009] Ke, Y., Qiang, L. R., and Quan, G. R. (2009). Bluetooth Sensor Network Communication. In *Proceedings of the 2009 International Conference on Industrial and Information Systems*, pages 203–206, Washington, DC, USA. IEEE Computer Society.
- [Kim and Lee 2007] Kim, K. and Lee, W. (2007). Mbal: A mobile beacon-assisted localization scheme for wireless sensor networks. In *Proceedings of 16th International Conference on Computer Communications and Networks, ICCCN*, pages 57–62. IEEE.
- [Laselva et al. 2005] Laselva, D., Zhao, X., Meinila, J., Jamsa, T., Nuutinen, J., Kyosti, P., and Hentila, L. (2005). Empirical Models and Parameters for Rural and Indoor Wideband Radio Channels at 2.45 and 5.25 GHz. In *Proceedings of IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, pages 654–658. IEEE.

- [Lee et al. 2009] Lee, S., Kim, E., Kim, C., and Kim, K. (2009). Localization with a Mobile Beacon Based on Geometric Constraints in Wireless Sensor Networks. *IEEE Transactions on Wireless Communications*, 8(12):5801–5805.
- [Lin et al. 2009] Lin, T.-H., Ng, I.-H., Lau, S.-Y., and Huang, P. (2009). Impact of Beacon Packet Losses to RSSI-Signature-Based Indoor Localization Sensor Networks. In *Proceedings of Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*, pages 389–390.
- [Mao et al. 2007a] Mao, G., Anderson, B. D. O., and Fidan, B. (2007a). Path Loss Exponent Estimation for Wireless Sensor Network Localization. *Computer Networks*, 51(10):2467–2483.
- [Mao et al. 2007b] Mao, G., Fidan, B., and Anderson, B. D. O. (2007b). Wireless Sensor Network Localization Techniques. *Computer Networks*, 51(10):2529–2553.
- [Nagpal et al. 2003] Nagpal, R., Shrobe, H., and Bachrach, J. (2003). Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network. In *Proceedings of the 2nd International Conference on Information Processing in Sensor Networks*, IPSN, pages 333–348, Berlin, Heidelberg. Springer-Verlag.
- [Navarra et al. 2010] Navarra, A., Pinotti, C., Ravelomanana, V., Betti Sorbelli, F., and Ciotti, R. (2010). Cooperative Training for High Density Sensor and Actor Networks. *IEEE Journal on Selected Areas in Communications*, 28(5):753–763.
- [Navarra and Pinotti 2011] Navarra, A. and Pinotti, C. M. (2011). Collision-free Routing in Sink-centric Sensor Networks with Coarse-grain Coordinates. In *Proceedings of the 21st International Conference on Combinatorial Algorithms*, IWOCA, pages 140–153, Berlin, Heidelberg. Springer-Verlag.
- [Patwari et al. 2005] Patwari, N., Ash, J. N., Kyperountas, S., Hero, A. A., Moses, R. L., and Correal, N. S. (2005). Locating the Nodes: Cooperative Localization in Wireless Sensor Networks. *IEEE Signal Processing Magazine*, 22(4):54–69.

- [Patwari and Kaserer 2011] Patwari, N. and Kaserer, S. K. (2011). Temporal Link Signature Measurements for Location Distinction. *IEEE Transactions on Mobile Computing*, 10(3):449–462.
- [Peng et al. 2005] Peng, H., Chuanshan, G., Min, Y., Dilin, M., and Bo, Y. (2005). ECLS: An Efficient Cooperative Localization Scheme For Wireless Sensor and Actor Networks. In *Proceedings of Fifth International Conference on Computer and Information Technology*, CIT, pages 396–400.
- [Pivato et al. 2011] Pivato, P., Palopoli, L., and Petri, D. (2011). Accuracy of RSS-Based Centroid Localization Algorithms in an Indoor Environment. *IEEE Transactions on Instrumentation and Measurement*, 60(10):3451–3460.
- [Rappaport 2001] Rappaport, T. S. (2001). *Wireless Communications: Principles and Practice, 2nd Edition*. Prentice Hall.
- [Wadaa et al. 2005] Wadaa, A., Olariu, S., Wilson, L., Eltoweissy, M., and Jones, K. (2005). Training a Wireless Sensor Network. *Mobile Networks and Applications*, 10(1-2):151–168.
- [Wang et al. 2011] Wang, J., Urriza, P., Han, Y., and Cabric, D. (2011). Weighted Centroid Localization Algorithm: Theoretical Analysis and Distributed Implementation. *IEEE Transactions on Wireless Communications*, 10(10):3403–3413.
- [Wang et al. 2009] Wang, Q., Fan, P., and Letaief, K. B. (2009). On the Log-Normal Fading Networks: Power Control and Spatial Reuse. In *Proceedings of the 2009 IEEE international conference on Communications*, ICC, pages 313–318, Piscataway, NJ, USA. IEEE Press.
- [Wang and Zhu 2009] Wang, W. and Zhu, Q. (2009). Sequential Monte Carlo Localization in Mobile Sensor Networks. *Wireless Networks*, 15(4):481–495.
- [Wang and Berger 2008] Wang, X. and Berger, T. (2008). Spatial Channel Reuse in Wireless Sensor Networks. *Wireless Networks*, 14(2):133–146.
- [Wang et al. 2010] Wang, Y., Lian, Y., and Nakao, A. (2010). LHDV-HOP: An Energy-Effective Range-Free Localization Scheme in Wireless Sensor Networks. In *Proceedings*



- of *12th IEEE International Conference on Communication Technology, ICCT*, pages 1007–1010. IEEE.
- [Wang et al. 2004] Wang, Z., Tameh., E. K., and Nix, A. R. (2004). Statistical Peer-to-Peer Channel Models for Outdoor Urban Environments at 2 GHz and 5 GHz. In *Proceedings of IEEE 60th Vehicular Technology Conference*, volume 7, pages 5101–5105. IEEE.
- [Xiao et al. 2010] Xiao, Q., Xiao, B., Cao, J., and Wang, J. (2010). Multihop Range-Free Localization in Anisotropic Wireless Sensor Networks: A Pattern-Driven Scheme. *IEEE Transactions on Mobile Computing*, 9(11):1592–1607.
- [Yick et al. 2008] Yick, J., Mukherjee, B., and Ghosal, D. (2008). Wireless Sensor Network Survey. *Computer Networks*, 52(12):2292–2330.
- [Ying et al. 2010] Ying, D., Jianping, W., and Chongwei, Z. (2010). Improvement of DV-Hop Localization Algorithms for Wireless Sensor Networks. In *Proceedings of 6th International Conference on Wireless Communications Networking and Mobile Computing, WiCOM*, pages 1–4. IEEE.
- [Zhang et al. 2010a] Zhang, S., Cao, J., Li-Jun, C., and Chen, D. (2010a). Accurate and Energy-Efficient Range-Free Localization for Mobile Sensor Networks. *IEEE Transactions on Mobile Computing*, 9(6):897–910.
- [Zhang et al. 2010b] Zhang, S., Cao, J., Zeng, Y., Li, Z., Chen, L., and Chen, D. (2010b). On Accuracy of Region-Based Localization Algorithms for Wireless Sensor Networks. *Computer Communications*, 33(12):1391–1403.
- [Zhang et al. 2009] Zhang, Y., Fang, Z., Li, R., and Hu, W. (2009). The Design and Implementation of a RSSI-Based Localization System. In *Proceedings of 5th International Conference on Wireless Communications, Networking and Mobile Computing, WiCom*, pages 1–4.
- [Zheng and Jamalipour 2009] Zheng, J. and Jamalipour, A., editors (2009). *Wireless Sensor Networks: A Networking Perspective*. Wiley-IEEE Press.

- [Zhou et al. 2010] Zhou, Z., Cui, J.-H., and Zhou, S. (2010). Efficient Localization for Large-scale Underwater Sensor Networks. *Ad Hoc Networks*, 8(3):267–279.