



INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Monitorização Wireless de Pessoas em Ambiente Doméstico

Márcio Luís Mendonça de Vasconcelos de Nóbrega

Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores

Júri

Presidente:

Orientador:

Co-Orientador:

Vogal:

Doutor Renato Jorge Caleira Nunes

Doutor António Manuel Raminhos Cordeiro Grilo

Outubro 2012

“Nunca é demasiado tarde para ser aquilo que sempre se quis ser.” — George Eliot

Agradecimentos

No culminar de uma longa etapa, são várias as pessoas que faço questão de mencionar neste trabalho, expressando os meus sinceros agradecimentos.

À minha família, em especial à minha mulher, Ana Bernardo pelo seu constante apoio e ao meu filho, Tomás, por me ter dado, ainda dentro da barriga da mãe, a força e motivação para concluir tão longa etapa.

Por ter aceite orientar a minha tese e por toda a ajuda proporcionada, ao meu Orientador, Professor Renato Nunes, pela disponibilidade demonstrada e pelas ideias e opiniões sugeridas que me proporcionaram reflexão e determinação para concluir este trabalho.

Ao meu Co-Orientador, Professor António Grilo, por me ter dado a conhecer nas suas aulas o mundo fantástico das redes móveis e um simulador tão poderoso como o OMNeT++.

Ao meu patrão e amigo, Pedro Dias, aos meus amigos Rodrigo Dias e Daniel Zacarias por me terem dado o tempo e a serenidade para terminar este trabalho.

Resumo

O aumento constante da população idosa mundial tem criado uma enorme quantidade de desafios ao desenvolvimento nacional, à sustentabilidade das famílias e à capacidade dos sistemas de saúde de darem suporte à população idosa. À medida que a tecnologia dos sensores wireless evolui, dispositivos de baixo consumo, reduzida largura de banda e capacidade de armazenamento médio, surgem no mercado, com custos de aquisição bastante reduzidos. A monitorização de ambientes domésticos baseada em sensores wireless, fornece um meio seguro e contido para pessoas idosas, permitindo que estas possam viver nas suas casas o máximo tempo possível. Este trabalho introduz o *Elder Monitoring System* (EMoS), um sistema desenvolvido no *Mixed Simulator* (MiXiM), onde foi implementado um protocolo de encaminhamento *Ad hoc On-demand Vector Routing* (AODV) e um sistema de localização baseado no HORUS, com a finalidade de monitorizar, num ambiente doméstico, pessoas idosas ou com necessidades especiais. Os resultados obtidos desta investigação demonstram a viabilidade de construir um sistema simulado para monitorização de pessoas num ambiente doméstico, onde aspectos de hardware comercialmente disponível foram também discutidos.

Palavras Chave

Redes de Sensores, Pessoas Idosas, Protocolos de Encaminhamento, Localização, MiXiM

Abstract

The consistent increase in the world's elder population has been putting a lot of challenges regarding national development, sustainability of families and the ability of health care systems to provide for ageing populations. As wireless sensing technology continues to evolve, devices integrating low-power, low-bandwidth radios and a modest amount of storage, emerge due to considerable reduced costs. Wireless sensors based home monitoring systems provide a safe, sound and secure environment for elder people, enabling them to live in their own home as long as possible. This work introduces the *Elder Monitoring System* (EMoS), a MiXiM based framework, in which an *Ad hoc On-demand Vector Routing* (AODV) protocol has been implemented together with a modified HORUS system, for tracking and monitoring, in a home environment, elder people or people with special needs. The results obtained from this research demonstrate the feasibility to build a monitoring system for elder care using a simulated environment in which several aspects of the hardware commercially available have been also discussed.

Keywords

Sensor Networks, Elder Care, Routing Protocols, Indoor Location, MiXiM

Conteúdo

1	Introdução	1
1.1	Motivação	2
1.2	Objectivos	3
1.3	Organização da Dissertação	3
2	Estado da Arte	5
2.1	Monitorização com Sinal Vídeo ou Áudio	6
2.2	Monitorização com Sensores <i>Wearable</i>	8
2.3	Monitorização com Sensores Domésticos	11
2.4	Discussão	13
3	Trabalho Relacionado	15
3.1	Monitorização Doméstica de Idosos	16
3.1.1	Necessidades nos Cuidados de Saúde	16
3.1.2	Necessidades na Monitorização	17
3.2	Localização em Redes de Sensores Wireless	19
3.2.1	Medidas de Localização	19
3.2.2	Sistemas de Localização	20
3.3	Encaminhamento em Redes de Sensores Wireless	22
3.3.1	Desafios e Decisões de Design	23
3.3.2	Protocolos de Encaminhamento	24
4	Ambiente de Trabalho	27
4.1	Objective Modular Network Test-bed (OMNeT++)	28
4.2	Mixed Simulator (MiXiM) para OMNeT++	31
4.3	Simulação de Obstáculos para MiXiM	34
4.4	Simulação da Propagação do Sinal	36
5	Arquitectura do Sistema	37
5.1	Sistema de Monitorização EMoS	38
5.1.1	Nó Móvel (MN)	40

5.1.2	Nó Fixo (SN)	41
5.1.3	Nó Base (BN)	41
5.2	Camada <i>Network</i> : <i>AODVRoute</i>	43
5.2.1	Tipos de Mensagens	43
5.2.2	Estruturas de Dados	44
5.2.3	Modo de Funcionamento	45
5.3	Camada <i>Application</i> : Localização	51
5.3.1	HORUS Modificado	51
5.3.2	Modelo Matemático	53
5.3.3	Estruturas de Dados e Ficheiros XML	56
5.3.4	Modo de Funcionamento	56
5.4	Mobilidade	61
6	Resultados	63
6.1	<i>Testbed</i>	64
6.2	Sistema de Localização	64
7	Conclusões e Trabalho Futuro	67
A	Apêndice 1 - Ficheiros XML de Exemplo	73
B	Apêndice 2 - Ficheiros XML de Configuração	76

Lista de Figuras

1.1	Pirâmides demográficas em Portugal entre 1950 e 2050 (estimativa) [1].	2
2.1	Arquitectura do sistema proposto em [2].	7
2.2	Arquitectura de fusão de decisão referida em [3].	7
2.3	Processo de detecção de quedas e alertas descrito no trabalho [6].	8
2.4	Exemplo de uma BSN [8].	9
2.5	Exemplo de uma WBAN [9].	10
2.6	Infraestrutura de rede <i>Mediacup</i> [15] que integra IR, CAN e LAN.	13
3.1	Classificação da localização em redes WSN [18].	20
3.2	Sistema de localização MoteTrack [20].	22
3.3	Protocolo SPIN [23].	24
3.4	Exemplo de funcionamento do protocolo LEACH.	25
4.1	Representação modular do ambiente de trabalho.	28
4.2	Estrutura modular do OMNeT++.	29
4.3	Arquitectura lógica de um executável OMNeT++ [31].	30
4.4	Ambiente de simulação Tkenv no OMNeT++ [31].	30
4.5	Simulação de uma rede no MiXiM [32].	31
4.6	Módulo de nó no MiXiM [32].	32
4.7	Divisão lógica da <i>framework</i> MiXiM.	32
4.8	Camada PHY do MiXiM [32].	33
4.9	Simulação de uma rede MiXiM com obstáculos.	34
5.1	Esquema modular do <i>Elder Monitorization System</i> (EMoS).	38
5.2	Estrutura interna de um nó no sistema EMoS.	39
5.3	Módulo <i>Wasmote</i> da <i>Libelium</i>	40
5.4	Rádio <i>XBee</i> IEEE 802.15.4 usado no módulo <i>Wasmote</i>	41
5.5	Módulo <i>Micaz</i> da <i>Crossbow</i>	42
5.6	Módulo <i>Wasmote Gateway</i> da <i>Libelium</i>	42
5.7	Tipos de mensagens do AODVRoute.	44
5.8	PktMap - estrutura para a gestão dos pacotes de aplicação em espera.	45

Lista de Figuras

5.9	RREQVector - estrutura que guarda os RREQs recebidos num nó.	45
5.10	RouteMap - estrutura que guarda os caminhos encontrados para nós destino. . .	45
5.11	Fluxograma de chegada da mensagem de aplicação à camada <i>Network</i>	47
5.12	Fluxograma de chegada de um <i>RREQ</i> vindo da camada <i>MAC</i>	48
5.13	Fluxograma de chegada de um <i>RREP</i> vindo da camada <i>MAC</i>	49
5.14	Fluxograma de chegada de uma mensagem de controlo com o pacote cuja entrega falhou no <i>MAC</i>	50
5.15	Fluxograma da chegada de um <i>RERR</i> vindo da camada <i>MAC</i>	50
5.16	Componentes do sistema HORUS modificado e fluxo de informação.	52
5.17	Exemplo de parametrização da distribuição de potência do sinal recebido [21]. . .	54
5.18	Estrutura de dados para armazenamento das posições do mapa rádio obtidas na fase <i>offline</i>	56
5.19	Estrutura de dados para armazenamento dos <i>clusters</i> obtidos na fase <i>offline</i>	57
5.20	Fluxograma do modo de funcionamento do nó móvel na localização.	58
5.21	Fluxograma do modo de funcionamento do nó base na localização.	60
5.22	Movimento do nó móvel durante a fase <i>offline</i> usando o modelo <i>TractorMobility</i> disponível no MiXiM.	61
6.1	Cenário da simulação criada para testar o EMoS.	64
6.2	Cenário de mobilidade para teste do sistema de localização.	65
6.3	Erros de estimativa no sistema HORUS sem técnicas de centro de massa e sem média.	65
6.4	Erros de estimativa no sistema HORUS com centro de massa e sem média. . . .	66
6.5	Erros de estimativa no sistema HORUS com centro de massa e com média. . . .	66
6.6	Erros de estimativa no sistema HORUS com centro de massa e sem média para um tempo de amostragem do nó móvel de $t=40s$	66

Lista de Tabelas

2.1	Aplicações para redes de sensores <i>In-body</i> e <i>On-body</i> [9].	11
2.2	Grau de intrusão por tecnologia usada em [12].	12
3.1	Índice de independência nas ADLs [17].	17
3.2	Uso de tecnologia pelos Idosos [16].	17
3.3	Classificação das ADLs [16].	18
4.1	Valores de atenuação por metro e por parede usados neste trabalho.	35
4.2	Atenuação de materiais de construção comuns para frequências de 5GHz e 2.4 GHz.	35
5.1	Parâmetros do rádio <i>XBee</i> da <i>Digi</i>	41
5.2	Parâmetros do rádio <i>Texas Instruments CC2420</i>	42

Lista de Acrónimos

BSN <i>Body Sensor Network</i>	9
BAN <i>Body Area Network</i>	9
WBAN <i>Wireless Body Area Network</i>	9
MAC <i>Medium Access Control</i>	9
GTS <i>Guaranteed Time Slots</i>	10
WLAN <i>Wireless Local Area Network</i>	11
WSN <i>Wireless Sensor Network</i>	11
IR <i>Infrared</i>	13
CAN <i>Car Area Network</i>	13
LAN <i>Local Area Network</i>	
RFID <i>Radio-frequency Identification</i>	
CM <i>Case Manager, profissionais de saúde do ramo da geriatria</i>	16
ADL <i>Activity of Daily Living</i>	16
PDA <i>Personal Digital Assistant</i>	
BS <i>Base Station</i>	21
QoS <i>Quality of Service</i>	23
SPIN <i>Sensor Protocols for Information via Negotiation</i>	24
DD <i>Direct Diffusion</i>	25
AODV <i>Ad hoc On-demand Vector Routing</i>	v
DSR <i>Dynamic Source Routing</i>	25
LEACH <i>Low-Energy Adaptive Clustering Hierarchy</i>	25
PEGASIS <i>Power-Efficient Gathering in Sensor Information Systems</i>	26
GEAR <i>Geographical and Energy Aware Routing</i>	26
TOA <i>Time of Arrival</i>	19
TDOA <i>Time Difference of Arrival</i>	19

Lista de Acrónimos

RSS	<i>Received Signal Strength</i>	19
POA	<i>Phase of Arrival</i>	19
AOA	<i>Angle of Arrival</i>	19
RM	<i>Radio Map</i>	20
RF	<i>Radio Frequency, rádio-frequência.</i>	
AP	<i>Access-Point</i>	21
OMNeT++	<i>Objective Modular Network Test-bed</i>	28
MiXiM	<i>Mixed Simulator</i>	v
NED	<i>Network Description</i>	29
MF	<i>Mobility Framework</i>	31
ChSim	<i>CHannel SIMulator</i>	31
NIC	<i>Network Interface Card</i>	
PHY	<i>Physical Layer</i>	
GSM	<i>Global System for Mobile Communications</i>	
EMoS	<i>Elder Monitoring System</i>	v
IEEE	<i>Institute of Electrical and Electronics Engineers</i>	
SD	<i>Secure Digital</i>	
MN	<i>Mobile Node</i>	38
SN	<i>Static Node</i>	38
BN	<i>Base Node</i>	38
RSSI	<i>Received Signal Strength Information</i>	
PC	<i>Personnal Computer</i>	
RREQ	<i>Route Request</i>	43
RREP	<i>Route Response</i>	43
RREP-ACK	<i>Route Response Acknowledgement</i>	43
RERR	<i>Route Error</i>	43
APP	<i>Application Layer</i>	
FIFO	<i>First In First Out</i>	
TTL	<i>Time To Live</i>	43
ACK	<i>Acknowledgement</i>	
PDF	<i>Probability Density Function</i>	56

1

Introdução

Conteúdo

1.1	Motivação	2
1.2	Objectivos	3
1.3	Organização da Dissertação	3

1.1 Motivação

O aumento da esperança de vida provoca actualmente um envelhecimento generalizado da população mundial o que coloca diversos desafios ao desenvolvimento nacional, à sustentabilidade das famílias e à capacidade dos sistemas de saúde. Durante anos recentes, o número de pessoas no mundo, com idades acima dos 60 anos aumentou de 200 milhões em 1950 para 670 milhões, sector etário que representa já cerca de 20% do total da população nos países desenvolvidos [1]. Com a deslocalização dos jovens para a periferia dos grandes centros e a baixa natalidade, aumenta cada vez mais o número de idosos que vivem sozinhos em suas casas. Esta situação cria ansiedade em todos os envolvidos, resultando muitas vezes num internamento precoce em lares, com um custo elevado e vagas limitadas.

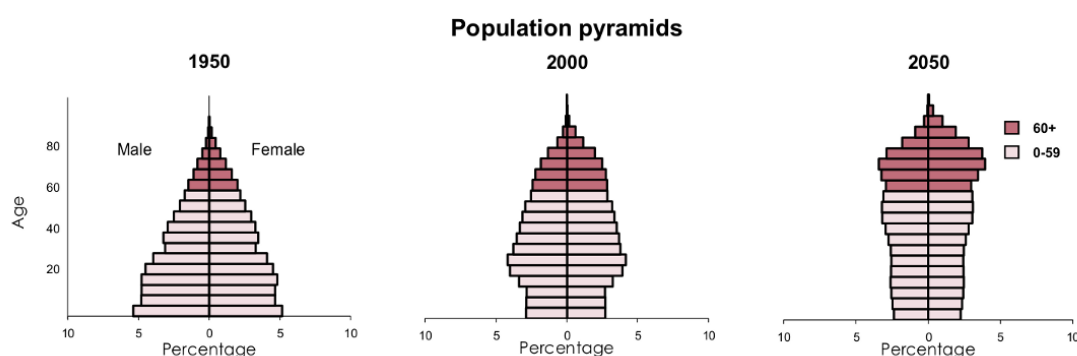


Figura 1.1: Pirâmides demográficas em Portugal entre 1950 e 2050 (estimativa) [1].

Pessoas com deficiências físicas ou mentais apresentam também uma idêntica necessidade de acompanhamento. Por exemplo, pessoas com deficiência mental média, normalmente têm capacidades sociais e funcionais para serem minimamente independentes, ainda que necessitem de alguma supervisão e assistência. Normalmente têm problemas tão básicos como, por exemplo, decidir quando se levantar ou deitar na cama, ou tomar medicamentos à hora certa.

A monitorização de ambos os casos descritos permitiria libertar mão-de-obra especializada para situações de maior dependência, reduzindo custos e aumentando a eficiência, notificando médicos ou hospitais da mudança de sinais vitais e comportamentos, que precedam situações de risco.

A evolução tecnológica dos sensores wireless tem vindo a introduzir no mercado sensores, rádios e processadores de baixa potência e baixo custo. Estes dispositivos, com o seu reduzido tamanho, têm um enorme potencial para o desenvolvimento de aplicações centradas no utilizador. Com um vasto tipo de sensores, as aplicações ubíquas¹ podem por isso surgir como alternativa de baixo custo e enorme valor acrescentado para monitorização de pessoas num ambiente doméstico, criando uma simbiose entre pessoa e máquina, que permitiria usufruir do

¹Aplicação que tem como objectivo tornar a interacção entre pessoa e máquina invisível, integrando a informática com acções e comportamentos naturais das pessoas.

direito de viver de forma independente, com privacidade e dignidade.

1.2 Objectivos

Nesta dissertação é proposto o desenvolvimento de uma solução onde uma ou mais pessoas, portadoras de um nó com comunicação wireless, se movimentam num ambiente onde existem outros nós wireless. Deverá ser possível localizar cada pessoa e estabelecer uma comunicação bidireccional entre esta e um servidor central.

Assim definem-se os seguintes objectivos:

- Identificar necessidades num ambiente doméstico e propor para estas, soluções de hardware existentes no mercado;
- Identificar uma plataforma de simulação existente que permita, de uma forma realista, simular o comportamento do sistema;
- Definir a arquitectura do sistema e os papeis de cada interveniente;
- Implementar a simulação de um protocolo de encaminhamento;
- Implementar a simulação de um sistema de localização;
- Simular casos de estudo que permitam validar o protocolo de encaminhamento e de localização implementados.

Esta dissertação apresenta as seguintes contribuições:

- Propõe um sistema simulado de monitorização de pessoas idosas;
- Propõe diversas soluções de hardware para a implementação de um sistema real;
- Implementa um módulo *Ad hoc On-demand Vector Routing* (AODV) para OMNeT++/MiXiM;
- Implementa um sistema de localização de nós móveis numa rede WSN, usando um sistema HORUS modificada para OMNeT++/MiXiM;

1.3 Organização da Dissertação

No Capítulo 1 é apresentada uma pequena introdução, referindo a motivação para o tema, bem como objectivos propostos e contribuições.

No Capítulo 2 é abordado o estado da arte. Neste capítulo faz-se uma análise da diversa bibliografia existente sobre o tema da monitorização. São abordados os sistemas de monitorização com áudio e vídeo, bem como os sensores *Wearable* e os sensores domésticos de uso comum..

1. Introdução

No Capítulo 3 é feita uma recolha de trabalho relacionado, considerado relevante para este trabalho. Temas como, a monitorização de idosos e a forma como estes aceitam a tecnologia, a visão dos prestadores de cuidados de saúde, são abordados, bem como as métricas necessárias em contexto de pessoas idosas. Seguidamente fala-se da localização em redes WSN, das possibilidades de localização, usando a potência do sinal recebido, o ângulo de chegada e são analisados três sistemas de localização. São ainda estudados os protocolos de encaminhamento, as vantagens e desvantagens de cada um.

No Capítulo 4 é feita uma introdução ao ambiente trabalho, ao processo de escolha e às decisões tomadas na simulação de obstáculos. É analisado ao pormenor o mecanismo de tratamento de sinal recebido no MiXiM e escolhido o modelo de propagação de sinal a usar.

No Capítulo 5 é descrita a arquitectura do sistema EMoS, dos seus componentes, da camada *network* onde é implementado o AODV e da camada de aplicação onde se implementa o sistema de localização HORUS. Faz-se uma descrição completa do funcionamento das duas camadas recorrendo para isso a fluxogramas exemplificativos.

No Capítulo 6 é feita uma pequena análise de resultados os efeitos produzidos pelas técnicas de centro de massa e média espaço-temporal usadas na estimação de espaço contínuo.

No Capítulo 7 faz-se a conclusão deste trabalho e lançam-se algumas ideias trabalho futuro.

2

Estado da Arte

Conteúdo

2.1	Monitorização com Sinal Vídeo ou Áudio	6
2.2	Monitorização com Sensores <i>Wearable</i>	8
2.3	Monitorização com Sensores Domésticos	11
2.4	Discussão	13

2. Estado da Arte

O conceito de casa inteligente actual consiste em vários electrodomésticos e outros dispositivos, com sensores, actuadores e/ou monitores biomédicos, usados pelos residentes numa base diária. Em alguns casos a casa é monitorizada recorrendo a tecnologias áudio e vídeo, sendo que estes sistemas apresentam uma excelente forma de monitorização mas têm algumas desvantagens:

- Custos elevados devido ao uso de sensores sofisticados e equipamentos áudio-visuais;
- Custos elevados de instalação devido à instalação individualizada;
- Elevada largura de banda necessária;
- Demasiada intrusão no quotidiano da pessoa criando um sentimento de falta de privacidade ou desconforto.

Três grupos de tecnologias emergem por entre a bibliografia revista:

- Monitorização com Sinal Vídeo ou Áudio;
- Monitorização com Sensores *Wearable*;
- Monitorização com Sensores Domésticos.

2.1 Monitorização com Sinal Vídeo ou Áudio

Em [2] através de um sensor wireless equipado com um acelerómetro e transportado pela pessoa, são detectadas possíveis quedas. Por forma a minimizar o número de falsos alarmes, são usadas câmaras que cobrem o espaço, que analisam a posição da pessoa e são activadas de acordo com a localização do nó móvel. Essa localização é obtida através de triangulação baseada nas posições conhecidas dos nós fixos e da potência recebida do nó móvel. É também apresentada a possibilidade de efectuar transmissão de voz utilizando o rádio IEEE 802.15.4, uma vez que já existem rádios com largura de banda necessária para efectuar transmissão de voz.

Em [3] e [4] é feita a combinação da informação fornecida por redes de sensores e sistemas de vídeo-vigilância. Através de uma inferência lógica que considera sequências de eventos são tomadas decisões tal como é possível observar em 2.2. O ocupante da casa usa um sensor não intrusivo para determinação da posição e comunicação por voz, mas não é necessária qualquer interacção com a tecnologia. À semelhança do trabalho anterior a privacidade é um tema fulcral e todo o tratamento de imagem é feito localmente usando *Smart Cameras* ¹.

No trabalho [5] é feita a aplicação de um sistema de monitorização num lar de idosos através de vídeo e áudio sem recurso a sensores portáteis. O trabalho referencia a insuficiência de

¹ câmaras que para além de captar imagem também podem tratar a imagem e obter resultados a partir desta

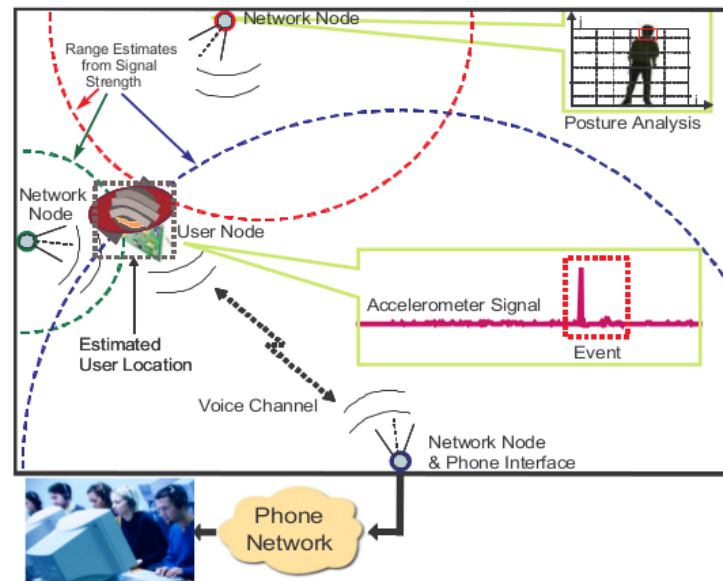


Figura 2.1: Arquitectura do sistema proposto em [2].

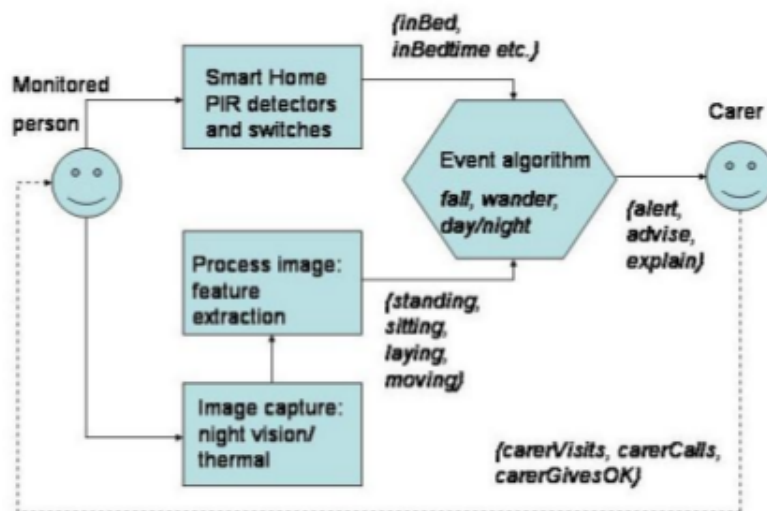


Figura 2.2: Arquitectura de fusão de decisão referida em [3].

profissionais em contraste com o rápido crescimento da população idosa e o pouco tempo que estes têm disponível para cada idoso. Emerge assim a necessidade de obter um conjunto de dados de forma autónoma e usado para detectar situações de perigo atempadamente, como por exemplo a instabilidade do andar ou registos comportamentais que favorecem a prescrição de medicamentos psicotrópicos. Os grandes desafios indicados são a localização por vídeo, a correcta identificação e marcação das pessoas no campo de visão e a análise das suas actividades individuais.

Partindo do conceito *aging in place*, onde idosos vivem de forma independente e segura nas suas próprias casas, o trabalho [6] apresenta, a monitorização de quedas mas também

2. Estado da Arte

funcionalidades utilitárias como a detecção de objectos, calendário, vídeo-conferência e livro de endereços. Recorrendo a câmaras e a técnicas de *machine learning* o sistema não necessita que o utilizador use um sensor. O sistema tem uma abordagem centralizada devido à forte exigência de processamento em tempo real e capacidade de memória. A detecção de objectos é feita verificando mudanças na imagem ou procurando objectos de acordo com as suas características.

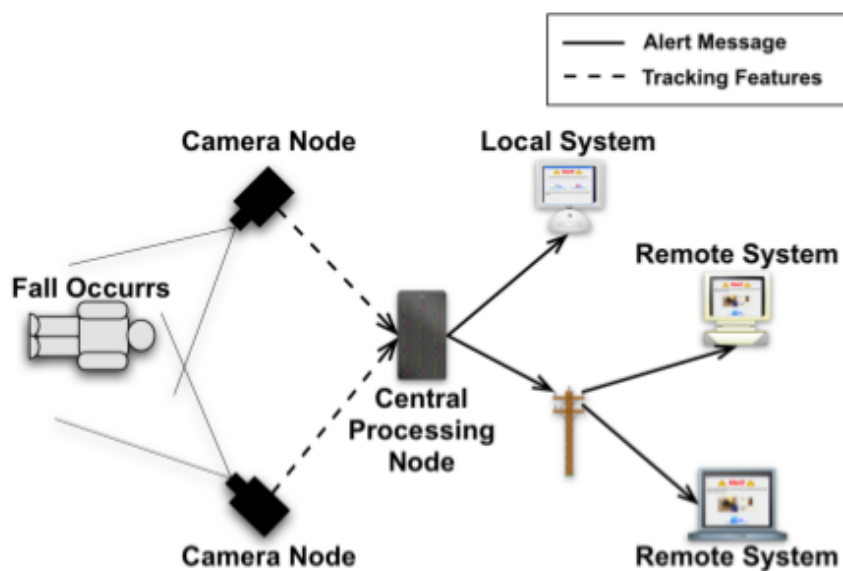


Figura 2.3: Processo de detecção de quedas e alertas descrito no trabalho [6].

Em [7] é utilizado o sinal áudio em conjunto com o vídeo para inferir acerca de uma possível queda. O sinal áudio torna-se essencial para distinguir entre uma pessoa que se sentou ou que caiu. Consideram-se processos de *Markov*², processos esses que permitem perceber se o comportamento do indivíduo está de acordo com o previsto ou não e assim tomar as medidas necessárias.

Embora cada aplicação tenha as suas mais-valias e a precisão dos sistemas onde o sinal vídeo é utilizado seja bastante elevada, existe a questão da privacidade que resulta numa baixa aceitação deste tipo de sistemas por parte de pessoas idosas.

A grande preocupação nos trabalhos identificados permanece na detecção de quedas e na fiabilidade dessa detecção.

2.2 Monitorização com Sensores *Wearable*

Com a evolução dos sensores wireless aparecem cada vez mais soluções que permitem fazer uma monitorização contínua do estado de saúde de uma pessoa, independentemente da sua localização ou actividade. A redução do tamanho dos sensores permite idealizar a criação

²Processo sem memória onde podem ser feitas previsões do futuro com base somente no estado presente, onde o futuro é independente do passado

de vestuário com sensores embutidos, suficiente leve e confortável para poder ser usado diariamente. Para além da monitorização há também a possibilidade de administrar medicamentos automaticamente (recorrendo a actuadores) ou de forma manual por um profissional de saúde de forma remota.

Em [8] é abordada a *Body Sensor Network* (BSN) como solução para a detecção precoce de problemas cardíacos, através de um conjunto de sensores equipados com medidor de temperatura, medidor de pulso, acelerómetro e até sensores capazes de obter um electrocardiograma³ (ECG), um electromiograma⁴ (EMG) ou um electroencefalograma⁵ (EEG). O sistema abordado tem um nó coordenador para onde todos os outros enviam informação e é usada a norma IEEE 802.15.4, que com suficiente largura de banda permite a transmissão da informação necessária.

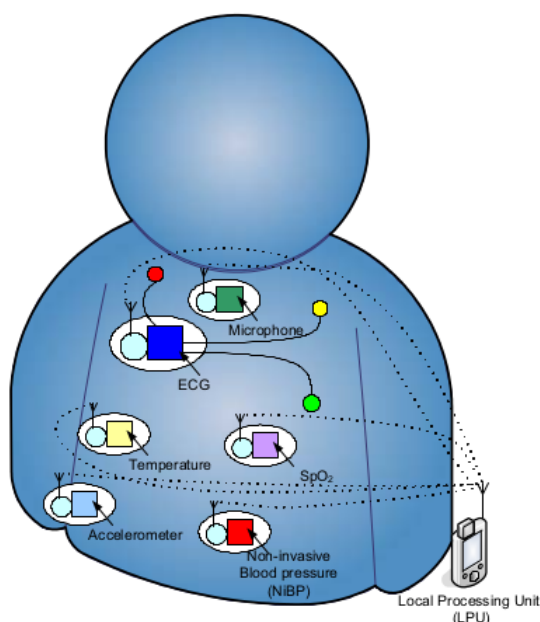


Figura 2.4: Exemplo de uma BSN [8].

A aplicação corre em TinyOS, open-source e com uma gestão de energia eficiente. É referida a estrutura modular do sistema operativo que permite escolher componentes conforme a sua aplicação, o que facilita bastante a utilização de diferentes tipos de sensores.

De referir o grupo de estudo IEEE 802.15 TG6⁶ que pretende estabelecer a norma para as *Body Area Networks* (BANs), que define um protocolo de comunicação para dispositivos de baixa potência que operem dentro, em ou à volta do corpo humano.

Em [9] é feita uma discussão sobre o tipo de antena e protocolo *Medium Access Control* (MAC) para *Wireless Body Area Networks* (WBANs), bem como sobre diversas aplicações para este tipo de redes. Na Figura 2.5 podemos observar que o tráfego é categorizado em 3

³Representação gráfica da actividade eléctrica do coração

⁴Representação do potencial eléctrico gerado pelas células dos músculos

⁵Representação da actividade do cérebro, obtida por pequenos sinais eléctricos chamados impulsos

⁶<http://www.ieee802.org/15/pub/TG6.html>

2. Estado da Arte

categorias: *On-demand* iniciado pelo médico ou nó coordenador para obter uma determinada informação de um ou mais sensores, *Emergency* iniciado pelos nós quando ultrapassam um determinado *threshold* e *Normal* que não apresenta qualquer elemento temporal crítico.

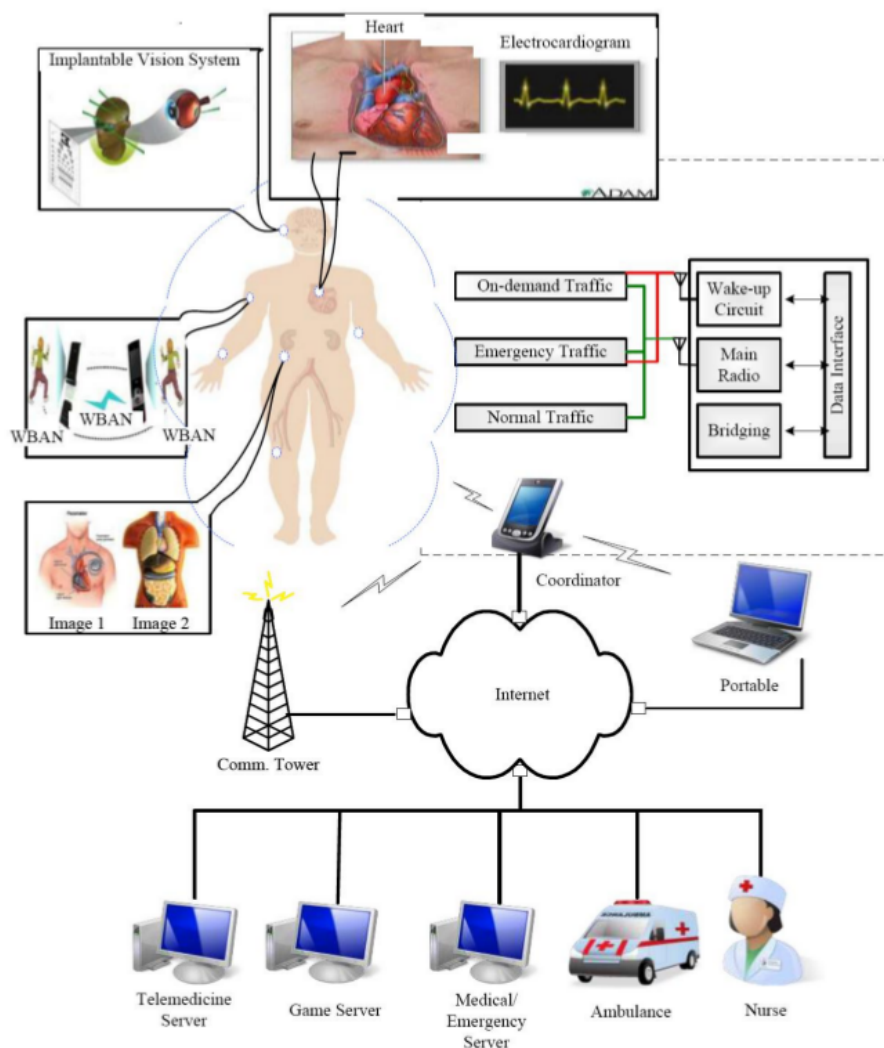


Figura 2.5: Exemplo de uma WBAN [9].

É referido o impacto do corpo na propagação do sinal através da constante dielétrica alta que este possui bem como através da condutividade parcial do tecido muscular que pode absorver parte do sinal, factores que se tornam ainda mais significativos quando as antenas são de muito pequena dimensão. Outro aspecto relevante referido neste trabalho é o facto de não existir no IEEE 802.15.4 um mecanismo fiável para o envio das mensagens *On-demand* e *Emergency*. Como possível solução para este problema é apontada a utilização das IEEE 802.15.4 *Guaranteed Time Slots* (GTS) para lidar com eventos críticos.

Por fim, o trabalho [9] indica através da Tabela 2.1 um conjunto de possíveis aplicações para sensores. Doenças cardiovasculares, detecção de doenças oncológicas, sistemas de tele-

2.3 Monitorização com Sensores Domésticos

Application Type	Sensor Node	Data Rate	Duty Cycle (per device)% time per	Power Consumption	QoS (Sensitive to Latency)	Privacy
In-body Applications	Glucose Sensor	Few Kbps	<1%	Extremely Low	Yes	High
	Pacemaker	Few Kbps	<1%	Low	Yes	High
	Endoscope Capsule	> 2Mbps	<50%	Low	Yes	Medium
On-body Medical Applications	ECG	3 Kbps	<10%	Low	Yes	High
	SpO2	32 bps	<1%	Low	Yes	High
	Blood Pressure	<10 bps	<1%	High	Yes	High
On-body Non-Medical Applications	Music for Headsets	1.4 Mbps	High	Relatively High	Yes	Low
	Forgotten Things Monitor	256 Kbps	Medium	Low	No	Low
	Social Networking	<200 Kbps	<1%	Low	No	High

Tabela 2.1: Aplicações para redes de sensores *In-body* e *On-body* [9].

medicina são algumas das aplicações mencionadas.

No trabalho [10] é analisada a coexistência entre *Wireless Local Area Networks* (WLANs) e *ZigBee* que operam na mesma frequência de 2.4GHz. A problemática de um número elevado de módulos WLAN, com potência de transmissão mais elevada, impossibilita a comunicação entre módulos *ZigBee*. É sugerida como solução a implementação de um algoritmo implementado na *Wireless Sensor Network* (WSN) que força a que, quando não existem frequências disponíveis, a WLAN seja obrigada a abandonar o canal deixando assim espaço para o sistema *ZigBee* comunicar.

[11] propõe um projecto que integra tecnologias WSN com redes públicas de comunicação por forma a construir um sistema eficiente de cuidados de saúde para idosos em casa. O sistema apresenta quatro funcionalidades principais: monitorização interior, monitorização exterior, actividade e decisão com base no estado de saúde. É feita a medição e colecção de parâmetros do corpo e da casa e enviada para um servidor central através de várias redes disponíveis.

Uma das principais desvantagens abordada na pesquisa efectuada é o facto de ter de existir de forma contínua um contacto com o corpo do idoso, de vários sensores, o que pode causar desconforto. Muitos idosos poderão não estar predispostos para usar uma BSN durante um tempo prolongado. Existe também a possibilidade de interferência com *pacemakers* ou outros equipamentos médicos que tenham sido colocados no idoso.

2.3 Monitorização com Sensores Domésticos

Neste tipo de monitorização recorre-se a sensores instalados em electrodomésticos e outros dispositivos utilizados pelos residentes. São obtidos padrões comportamentais através da correlação com a utilização dos diversos aparelhos numa casa. Uma das maiores vantagens deste tipo de monitorização é a privacidade, uma vez que a informação fornecida por cada sensor não contém qualquer identificação da pessoa que o acciona. Usam-se dispositivos do dia-a-dia o que não implica mudanças de comportamento, como por exemplo com a utilização dos sensores

2. Estado da Arte

Wearable e as BSN. Esta abordagem é chamada de *artefact computing model* e representa um paradigma de mudança na interação pessoa-máquina na sua forma explícita para uma forma implícita.

Identificam-se vários tipos de sensores aplicáveis a dispositivos domésticos:

- Sensores de pressão;
- Sensores de movimento e proximidade;
- Sensores de temperatura;
- Sensores RFID;
- Interruptores;
- Sensores de vibração;
- Sensores de caudal de água ou gás;
- Sensores de corrente;

No artigo [12] aborda-se a prestação de cuidados de saúde aos idosos num complexo construído pela *Elite Care*⁷. Com o objectivo de dar maior autonomia aos residentes são criados ambientes personalizados de sensores. O sistema permite identificar residentes que precisam de cuidados imediatos ou iluminar o caminho para um residente que vá durante à noite à casa-de-banho. Na Tabela 2.2, a partir de um estudo feito com questionários feitos aos residentes é obtido o grau de intrusão de cada uma das tecnologias implementadas.

Category	Function	Rating
Computing infrastructure and Web servers	Apartment consoles, personalized databases, database	High
Mobile computing devices	Personal badge locators with help functions used by all residents	Medium
Wireless networking	RF and IR locators	Medium
Device discovery	Static configuration	Low
Service discovery	Static configuration	Low
Multimodal user interfaces at apartment consoles	Environmental sensors for location, weight, and speech recognition	High
Pervasive databases	Personalized databases for status and history	High
User authentication	Statically configured	Low
Secure networking	Short range transmission of limited data, relying on the facility's physical security.	Low

Tabela 2.2: Grau de intrusão por tecnologia usada em [12].

[13] usa sensores de pressão para localização. É referida, a título de exemplo, a aplicação do sistema a uma pessoa com doença de Alzheimer num estágio médio e cuja detecção do

⁷<http://www.elite-care.com>

movimento permite activar ecrãs que se ligam quando a pessoa se aproxima e indicam as opções de percurso na casa. Os melhores sensores conseguem identificar a posição e direcção do utilizador, no entanto a \$10800 por metro quadrado não é uma alternativa viável. No projecto são usados os *Phidgets* 1.5 polegadas que para 32.5 metros quadrados custa \$4000. Sendo o custo uma desvantagem evidente, são propostas alternativas, por exemplo a redução de sensores às zonas previsíveis de passagem ou a utilização de modelos de previsão que preencham as secções sem sensores.

Em [14] é abordado o *PlaceLab*. Situado em Cambridge é um laboratório vivo para estudo das tecnologias ubíquas. Está optimizado para moradias unipessoais. Foram criadas para este laboratório 15 divisões e em cada foram colocadas redes de 25 a 30 sensores.

O projecto *Mediacup* [15] faz uma análise da adaptação de sensores, processamento e comunicação a dispositivos domésticos. Neste artigo uma caneca é adaptada com sensores de movimento e temperatura e ligada em rede com diversos outros dispositivos. Num cenário completo, todos os objectos de uso diário numa casa poderiam ser adaptados. É usado um processador de 1MHz para redução do consumo energético sendo o carregamento feito através de um campo electromagnético instalado num pires. É utilizada a tecnologia *Infrared* (IR) para a comunicação, através de mensagens, com transdutores que usam uma arquitectura *Car Area Network* (CAN) integrada por sua vez com uma LAN (Figura 2.6)

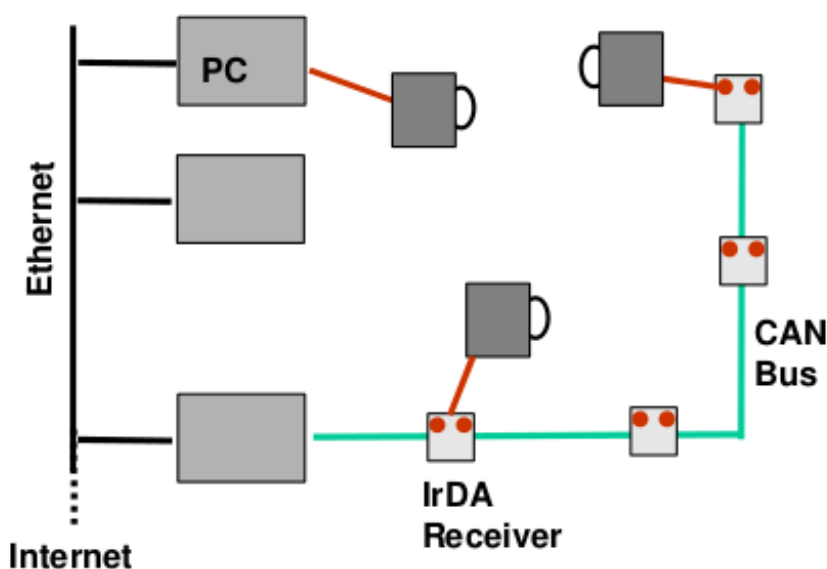


Figura 2.6: Infraestrutura de rede *Mediacup* [15] que integra IR, CAN e LAN.

2.4 Discussão

A bibliografia revista permitiu adquirir um maior conhecimento das soluções existentes no mercado, fornecendo pistas valiosas para o desenvolvimento desta dissertação.

2. Estado da Arte

O rápido crescimento das redes BSN e a diversidade de sensores existentes, tornam a integração das mesmas como uma rede WSN muito aliciante, traduzindo um valor acrescido na monitorização de pessoas, pela possibilidade de detectar situações anómalas antes de qualquer sinal evidente da mesma.

O grande foco nas quedas de idosos, encontrado em diversos trabalhos permite antever a necessidade de incluir num sistema de monitorização, sensores com câmaras embutidas, pese embora todas as questões de privacidade e custos associadas.

A utilização de sensores de uso doméstico comuns e tecnologicamente menos evoluídos, para uma monitorização não-intrusiva é também uma possibilidade a explorar.

Por último a possibilidade de tornar os sensores parte integrante dos objectos do dia-a-dia, revela a necessidade de tornar as aplicações mais ubíquas mesmo que para isso seja necessário fazer algumas concessões ao nível dos custos.

3

Trabalho Relacionado

Conteúdo

3.1	Monitorização Doméstica de Idosos	16
3.1.1	Necessidades nos Cuidados de Saúde	16
3.1.2	Necessidades na Monitorização	17
3.2	Localização em Redes de Sensores Wireless	19
3.2.1	Medidas de Localização	19
3.2.2	Sistemas de Localização	20
3.3	Encaminhamento em Redes de Sensores Wireless	22
3.3.1	Desafios e Decisões de Design	23
3.3.2	Protocolos de Encaminhamento	24

3.1 Monitorização Doméstica de Idosos

Em [16] faz-se uma análise de aspectos fundamentais na monitorização doméstica de idosos ouvindo os profissionais de cuidados de saúde. São também neste mesmo trabalho sugeridas diversas propriedades monitorizáveis e feita uma análise global da rede de cuidados de saúde.

3.1.1 Necessidades nos Cuidados de Saúde

No estudo intitulado “*The Activities of Daily Living Study*” em [16] são examinados questionários (91) feitos a profissionais de saúde que prestam cuidados de monitorização ao domicílio. Pretende-se determinar a forma como a tecnologia pode ajudar pessoas idosas a envelhecer em casa com a colaboração dos profissionais de saúde, a necessidade de autonomia do idoso e as necessidades da família e amigos.

Designam-se *Case Managers* (CMs) aos profissionais de saúde que prestam cuidados ao domicílio (ex:enfermeiros,médicos). Os CMs interagem de forma activa com as pessoas idosas, presencialmente ou por telefone.

Uma parte significativa da monitorização do CMs são as chamadas *Activities of Daily Living* (ADLs), uma lista de actividades que permite medir a função cognitiva e física do idoso (Tabela 3.1).

Esta lista permite definir numa escala de 0-muito dependente a 6-independente, o grau de dependência do idoso. São também apresentados desafios à introdução de novas tecnologias pelos CMs, nomeadamente:

- Receio de perda do emprego por parte dos CMs;
- Problemas de aceitação da nova tecnologia por parte dos idosos, visto que estes têm tendência a esconder informação com receio de irem para a um lar de idosos;
- Problemas de privacidade;

O processo de integração de um sistema de monitorização apenas pode ser um sucesso se os profissionais de saúde estiverem activamente envolvidos na sua implementação.

São identificadas diversos problemas de saúde nos idosos, sendo os mais comuns a fraqueza, diabetes, surdez, perda de visão, deficiência nutricional e demência moderada.

Na Tabela 3.2 o estudo identifica as tecnologias e a sua aceitação por parte dos idosos. A comunicação e a segurança são identificados claramente como muito valorizados, através dos botões de emergência e sistemas de segurança, enquanto que a tecnologia de monitorização coloca mais incerteza e desconfiança aos idosos.

ACTIVITIES POINTS (1 OR 0)	INDEPENDENCE: (1 POINT) NO supervision, direction or personal assistance	DEPENDENCE: (0 POINTS) WITH supervision, direction, personal assistance or total care
BATHING POINTS: _____	(1 POINT) Bathes self completely or needs help in bathing only a single part of the body such as the back, genital area or disabled extremity.	(0 POINTS) Needs help with bathing more than one part of the body, getting in or out of the tub or shower. Requires total bathing.
DRESSING POINTS: _____	(1 POINT) Gets clothes from closets and drawers and puts on clothes and outer garments complete with fasteners. May have help tying shoes.	(0 POINTS) Needs help with dressing self or needs to be completely dressed.
TOILETING POINTS: _____	(1 POINT) Goes to toilet, gets on and off, arranges clothes, cleans genital area without help.	(0 POINTS) Needs help transferring to the toilet, cleaning self or uses bedpan or commode.
TRANSFERRING POINTS: _____	(1 POINT) Moves in and out of bed or chair unassisted. Mechanical transferring aides are acceptable.	(0 POINTS) Needs help in moving from bed to chair or requires a complete transfer.
CONTINENCE POINTS: _____	(1 POINT) Exercises complete self control over urination and defecation.	(0 POINTS) Is partially or totally incontinent of bowel or bladder.
FEEDING POINTS: _____	(1 POINT) Gets food from plate into mouth without help. Preparation of food may be done by another person.	(0 POINTS) Needs partial or total help with feeding or requires parenteral feeding.

Tabela 3.1: Índice de independência nas ADLs [17].

Technology	Definitely	Maybe	Never	Unsure	What's that?	Already have
Emergency call buttons	77%	22%	0%	0%	0%	1%
Home security system	57%	39%	1%	2%	0%	1%
Cell phones	51%	42%	2%	1%	0%	3%
Internet	18%	66%	4%	8%	1%	3%
Computers / Laptops	16%	64%	6%	10%	1%	3%
Motion detectors	13%	58%	6%	22%	0%	1%
Microphones	8%	43%	18%	31%	0%	0%
Cameras	6%	51%	11%	31%	0%	1%
PDAs	5%	51%	12%	30%	2%	0%

Tabela 3.2: Uso de tecnologia pelos Idosos [16].

3.1.2 Necessidades na Monitorização

Com base nos resultados do estudo referenciado na Secção 3.2.1 o trabalho [16] faz uma análise de diversas temáticas de utilização de um sistema de monitorização.

3. Trabalho Relacionado

Localização doméstica. Determinar se o idoso se levantou pela manhã e os seus padrões de movimento são também apontadas como duas informações importantes. Uma granularidade menor que a divisão pode por isso ser importante sendo necessária uma maior precisão do sistema.

Agendamento de visitas ao domicílio. Saber se o idoso está ou não em casa é apontado pelos CMs como um factor de melhoria no agendamento de visitas ao domicílio.

Visitas ao Hospital e Socialização. A integração do sistema de monitorização doméstico com outro baseado em GPS e integrado no sistema de saúde permitiria para todos os intervenientes no sistema saber onde estava um idoso a qualquer momento para além do ambiente doméstico.

Número de ocupantes da casa. Sabendo os padrões de ocupação da casa é possível perceber quais os momentos em que os idosos têm menos apoio familiar. Em casos mais agudos esta monitorização permitiria determinar se era ou não seguro para o idoso continuar em sua casa durante uma determinada altura do dia.

Animais de estimação. A maioria dos clientes dos CMs entrevistados tinham animais de estimação o que pode ser um problema para sistemas baseados em sensores de pressão.

Oportunidades de instrumentação. São apresentadas diversas oportunidades de instrumentação sem grande necessidade da mudança de hábitos como a aplicação de sensores a bengalas, andarilhos ou cadeiras de rodas. O facto de existir uma baixa partilha deste tipo de objectos indica que seriam uma boa hipótese de monitorização não-intrusiva.

Privacidade. É bastante refutada a utilização de câmaras, microfones ou PDAs enquanto que os sensores de movimento poderiam ser alternativas viáveis.

Very Important to Know		Very Difficult to Collect		Top Ten
Falling	99%	Taking medication	11%	Taking medication
Taking medication	93%	Eating/nutrition	10%	Falling
Eating/nutrition	84%	Falling	10%	Eating/nutrition
Toileting	82%	Socializing	9%	Toileting
Bathing	80%	Getting out of bed	7%	Getting out of bed
Getting out of bed	78%	Cooking	4%	Bathing
Grooming/hygiene	74%	Toileting	4%	Cooking
Dressing	68%	Bathing	3%	Socializing
Cooking	61%	Grooming/hygiene	3%	Grooming/hygiene
Leaving the house	56%	Dressing	3%	Dressing

Tabela 3.3: Classificação das ADLs [16].

Escolha das ADLs a monitorizar. Na Tabela 3.3 é feita uma classificação das ADLs. O maior valor acrescentado está naquelas que são mais difíceis de obter mas mais importantes para serem conhecidas pelo CM.

Atenção à actividade das famílias ou assistentes. É importante perceber se existe de facto

um apoio real dos familiares ou outros assistentes ao idoso, para além de saber que o mesmo está acompanhado.

Monitorização do uso de equipamentos. A inclusão nos equipamentos de saúde de sensores que analisem o estado do equipamento ou a força exercida pelo idoso no mesmo, poderiam ajudar a determinar melhor o ponto em que é necessário passar de uma bengala para um andador ou de um andador para uma cadeira de rodas.

3.2 Localização em Redes de Sensores Wireless

A chave para obter uma localização fiável é representar de forma precisa os efeitos da degradação causada pelo canal de propagação no sinal. A propagação no mundo real sofre diversas perturbações causadas por obstruções, reflexões e pessoas ou objectos em movimento, o que torna esta representação um problema de elevada complexidade. Nesta secção enumeram-se os tipos de medições que permitem inferir uma localização e analisa-se bibliografia relacionada com o objectivo de identificar algoritmos de localização distintos, as suas vantagens e desvantagens na aplicação ao objecto deste trabalho.

3.2.1 Medidas de Localização

Vários tipos de medições permitem inferir uma localização, nomeadamente:

- *Time of Arrival* (TOA)
- *Time Difference of Arrival* (TDOA)
- *Received Signal Strength* (RSS)
- *Phase of Arrival* (POA)
- *Angle of Arrival* (AOA)

Na medida do **TOA** mede-se o tempo que um sinal demora a chegar ao nó de destino. A distância entre origem e destino é obtida multiplicando o atraso entre o momento da transmissão e o momento da recepção do sinal pela velocidade de propagação do sinal. O requisito mais importante é a sincronização entre nós que obriga à existência de hardware de maior complexidade e a troca de mensagens de sincronização. Ruído aditivo e efeitos multi-caminho são as maiores fontes de erro neste tipo de medição.

Utilizando o **TDOA** é medida a diferença entre os tempos de chegada em diversos nós dum mesmo sinal enviado pelo emissor. Um mínimo de dois nós é necessário para uma estimativa em duas dimensões da posição do do emissor. À semelhança do TOA é necessária sincronização entre os nós o que obriga uma vez mais a hardware complexo que aumenta o custo do nó.

3. Trabalho Relacionado

O RSS é a medida da potência do sinal recebido. Este método não necessita de qualquer hardware especial para sincronização. A potência do sinal é uma função da distância, cuja localização pode ser baseada num modelo, onde se admite que as características de propagação do sinal são bem conhecidas ou então baseada num mapa de medições, *Radio Map* (RM), onde é feita uma amostragem da potência em diversas localizações.

Com a medida POA o objecto de medição é o ângulo de chegada. Este método usa a diferença na fase do sinal para determinar a localização do nó emissor.

Por último a medida de AOA indica o ângulo a que o sinal chega ao receptor, medido com antenas direccionais ou um conjunto de antenas. São usadas relações geométricas simples para calcular a posição do nó emissor.

3.2.2 Sistemas de Localização

Os esquemas de localização são diversos e variam conforme o tipo de aplicação.

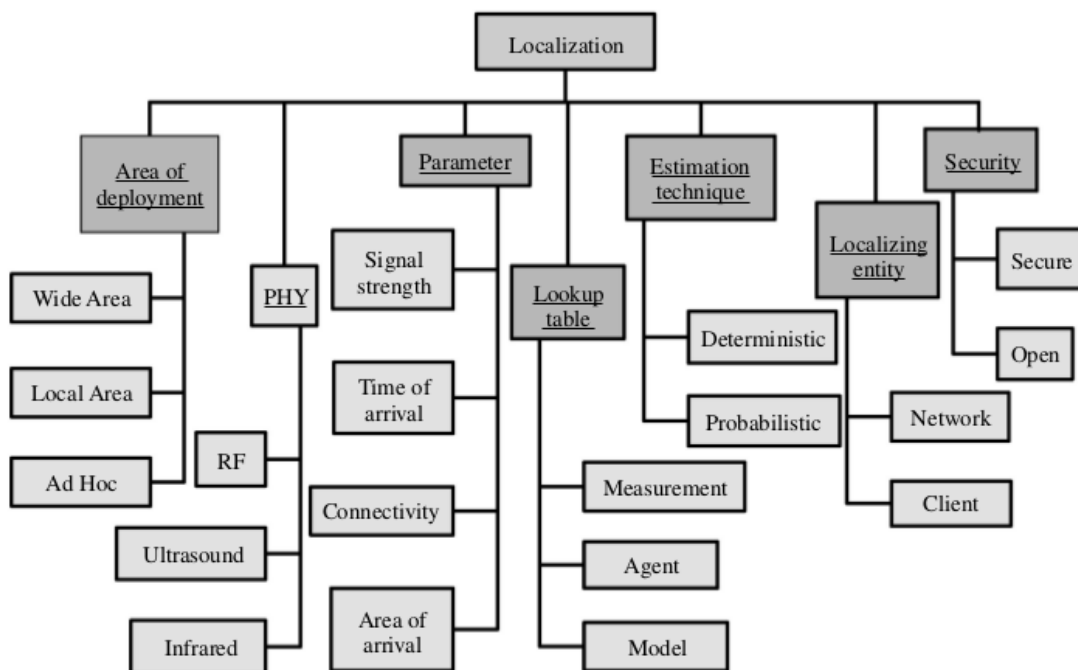


Figura 3.1: Classificação da localização em redes WSN [18].

A partir da Figura 3.1 é possível, desde já, fazer algumas observações relativas ao tema deste trabalho.

Área de Instalação: A área de instalação deverá ser local uma vez que estamos perante um ambiente doméstico.

PHY: O sinal rádio obtido deverá ser a RF pois o infravermelho necessita de estar na linha de visão o que não será sempre possível num ambiente doméstico e o ultra-som, embora tenha uma precisão bastante elevada é uma tecnologia cara.

Medida de Localização: Dada o baixo custo do hardware a utilizar a medida deverá ser a potência do sinal.

Tabela de Procura: A monitorização será feita no interior onde o percurso do sinal sofre muitas perturbações. Assim sendo a colecção de medidas prévia será a opção indicada uma vez que o modelo de propagação é altamente complexo e pode variar bastante e a utilização de agentes aumenta bastante o custo.

Técnica de Estimação: Neste caso a resposta não é tão evidente e terão de ser comparadas as precisões de cada uma das opções: Determinístico ou Probabilístico.

Entidade localizadora: As entidades a localizar são os nós móveis e como tal deverá ser a rede a efectuar a localização para efectuar uma poupança de energia no nó móvel.

Segurança: Embora se tenha optado por deixar esta questão fora do âmbito deste trabalho por limitações do simulador, um sistema de monitorização doméstica não poderia ser aberto pelas questões de privacidade abordadas na Secção 3.1.2.

Feitas algumas opções são analisados alguns trabalhos que se enquadram nas características escolhidas.

O RADAR [19], é um método determinístico onde é feita a localização recorrendo a diversas *Base Stations* (BSs) numa WLAN e apresenta segundo os autores uma precisão de 2.94 m em 50% dos casos. Este esquema de localização usa a potência do sinal RF de vários *Access-Points* (APs) para triangular a posição do nó móvel. Tem uma fase *offline* onde é construído um mapa rádio do local, em que o nó móvel envia pacotes (*beacons*) para os nós fixos que registam cada amostra, sendo o conjunto de amostras reunido numa tabela com tuplos da forma (x, y, d, ss_i, snr_i) para cada BS_i . Posteriormente na fase *online* é feita uma comparação entre os sinais recebidos e o mapa rádio que permite obter a coordenada mais próxima para a potência registada. Através do centróide das várias posições obtidas para cada AP obtém-se então a coordenada estimada do nó móvel.

Em [20] temos o MoteTrack que, com base no trabalho RADAR, sugere um esquema para uma rede de sensores wireless. É indicada uma precisão de 2m em 50% dos casos e de 3m em 80% dos casos. Este sistema de localização foca a robustez, a distribuição do algoritmo e a precisão da localização. Na Figura 3.2 podemos observar o sistema MoteTrack. B_1 , B_2 e B_3 são nós fixos e M um nó móvel. Na fase *offline*, para cada posição, é feita uma recolha pelo nó móvel de várias mensagens assinatura enviadas pelos nós fixos B_i , dando origem a uma tabela de tuplos $(x, y, B_i, P_i, RSSI_{medio})$ que constitui o mapa rádio. Na fase *online* o nó móvel recebe a assinatura do nó fixo durante um determinado tempo de amostragem. Após esse tempo devolve o resultado para o nó fixo de modo a que este determine posição com base nas amostras que detém. Cada nó fixo devolve a sua estimativa para o nó móvel que calcula o centróide das diversas posições recebidas.

O HORUS [21], um método probabilístico, apresenta uma precisão de 0.86 a 1.32m para 90%

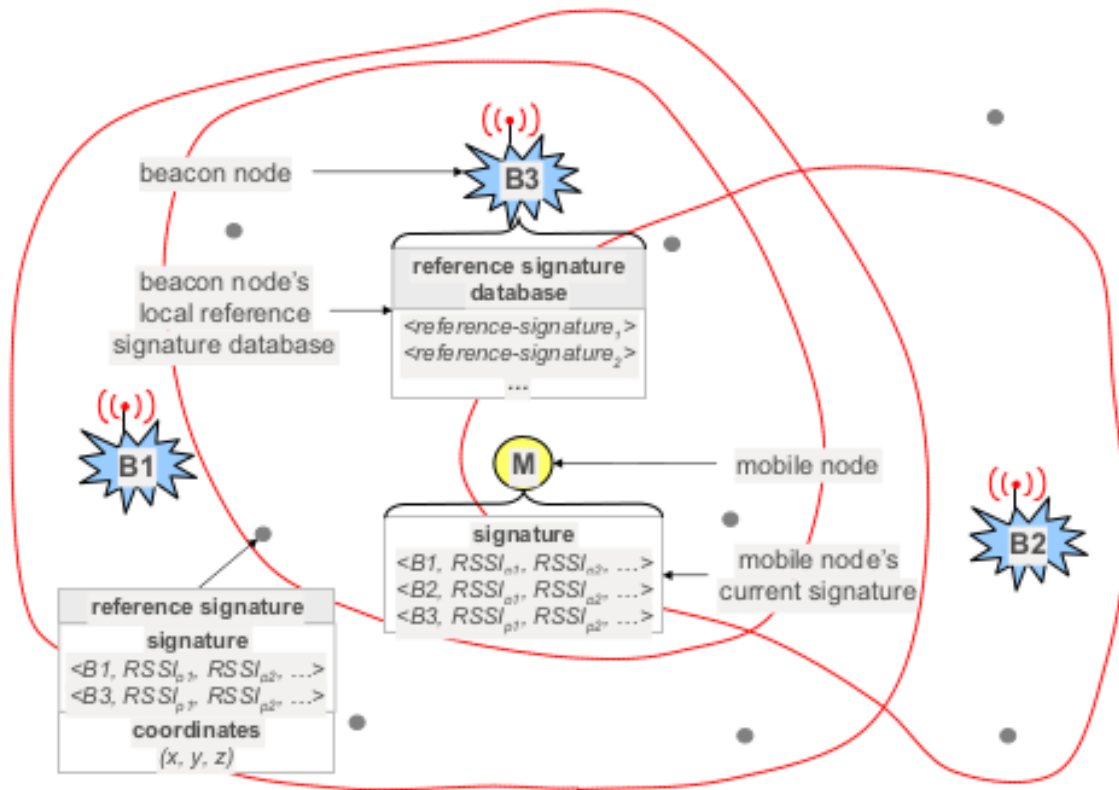


Figura 3.2: Sistema de localização MoteTrack [20].

dos casos. Num método probabilístico é guardada informação sobre as distribuições da potência do sinal em vez da média como num método determinístico. O objectivo do HORUS é atingir um alto nível de precisão e baixas necessidades de processamento. À semelhança do MoteTrack, este método usa técnicas de *clustering* para reduzir o espaço de procura quando tenta obter uma posição a partir do mapa rádio. O sistema é constituído também por uma fase *offline* onde é feita a construção de um mapa rádio que é depois dividido em *clusters*. Na fase *online* é feito no nó móvel o cálculo da posição recorrendo a várias filtros, que vão passo a passo melhorando a precisão da localização.

3.3 Encaminhamento em Redes de Sensores Wireless

Com a redução do custo dos sensores wireless tornou-se possível construir WSNs com centenas ou milhares de nós. A falta de um esquema global de endereçamento, as condicionantes energéticas ou a possibilidade de existirem nós que se movimentam provocando modificações na topologia da rede recorrentes, faz surgir a necessidade de encontrar um protocolo de encaminhamento adequado.

3.3.1 Desafios e Decisões de Design

Em [22] são abordados os diversos desafios no design de protocolos de encaminhamento.

Uma WSN apresenta diversas restrições tais como fornecimento de energia limitado pelo uso de bateria, processamento limitado ou largura de banda reduzida devido a rádios relativamente simples.

Instalação dos nós. A forma como os nós são instalados depende do tipo de aplicação e pode ser determinística ou aleatória. Se for aleatória a distribuição não é uniforme o que pode requerer *clustering*. A distância de transmissão é reduzida o que obriga a que a comunicação seja feita através de vários nós.

Tolerância a falhas. Alguns sensores podem falhar devido à falta de energia, dano físico ou interferência. Essas falhas não podem por isso condicionar ou impedir a comunicação e devem existir protocolos MAC e de encaminhamento que consigam detectar essa situação e reformular a topologia da rede.

Modelo de aquisição de dados. A forma como é feita a aquisição de dados é dependente da aplicação e pode ser orientada ao tempo, para aplicações de monitorização periódica ou ao evento para e à *query*, para nós que reagem a mudanças na medição de parâmetros ou a um pedido feito pela *Base Station* (BS).

Homogeneidade dos nós ou ligações. Os nós podem ter todos capacidades iguais sendo a rede homogénea ou então podem ter capacidades diferenciadas, havendo nós mais básicos e outros mais complexos.

Escalabilidade. Devido ao elevado número de nós possível numa WSN qualquer protocolo de encaminhamento deve ser escalável reagindo de forma automática à adição ou remoção de nós da rede.

Dinâmica da rede. A maior parte das arquitecturas assume que os nós estão fixos. No entanto para aplicações em que a topologia muda, a estabilidade dos caminhos torna-se um assunto importante, sendo necessário efectuar a actualização periódica ou redescoberta de novos caminhos.

Agregação de dados. Os dados de vários sensores podem ser agregados para que o número de transmissões sofra uma redução. A agregação pode ser feita com remoção de duplicados, valores mínimos, valores máximos e valores médios.

Quality of Service (QoS). Em algumas aplicações os dados têm de ser entregues com sucesso durante um determinado limite de tempo após a sua obtenção, caso contrário perdem significado ou introduzem erros desnecessários no sistema. Este limite de tempo pode ser gerido de forma dinâmica conforme a qualidade da transmissão.

3.3.2 Protocolos de Encaminhamento

Os protocolos nas WSNs podem ser classificados conforme a sua estrutura em *flat-routing* onde todos os nós têm as mesmas capacidades e papéis na rede, *hierarchical-routing* em que existem nós com capacidades diferenciadas e papéis diferentes e *location-based routing* onde a posição dos nós é parte integrante do protocolo de encaminhamento.

Flat-routing

O *Sensor Protocols for Information via Negotiation* (SPIN) [23] surge com a necessidade de resolver três problemas nos métodos clássicos de envio de mensagens (*Flooding* e *Gossiping*), a implosão causada pela recepção de várias mensagens repetidas vindas de vários nós diferentes, a sobreposição resultante dos dados obtidos por sensores próximos e a falta de adaptação aos recursos existentes no nó. Na Figura 3.3 está um exemplo onde são utilizadas os três tipos de mensagens ADV (*advertisement*), REQ (*request*) e DATA. O nó A pretende enviar uma mensagem para o nó B e envia um ADV (a). B está pronto para receber e envia para A um REQ (b). A recebe o REQ e envia uma mensagem DATA para B (c). B continua o processo da mesma forma para os seus nós vizinhos. Este protocolo permite poupar energia e reduzir o envio de informação redundante mas não dá garantias de entrega de dados.

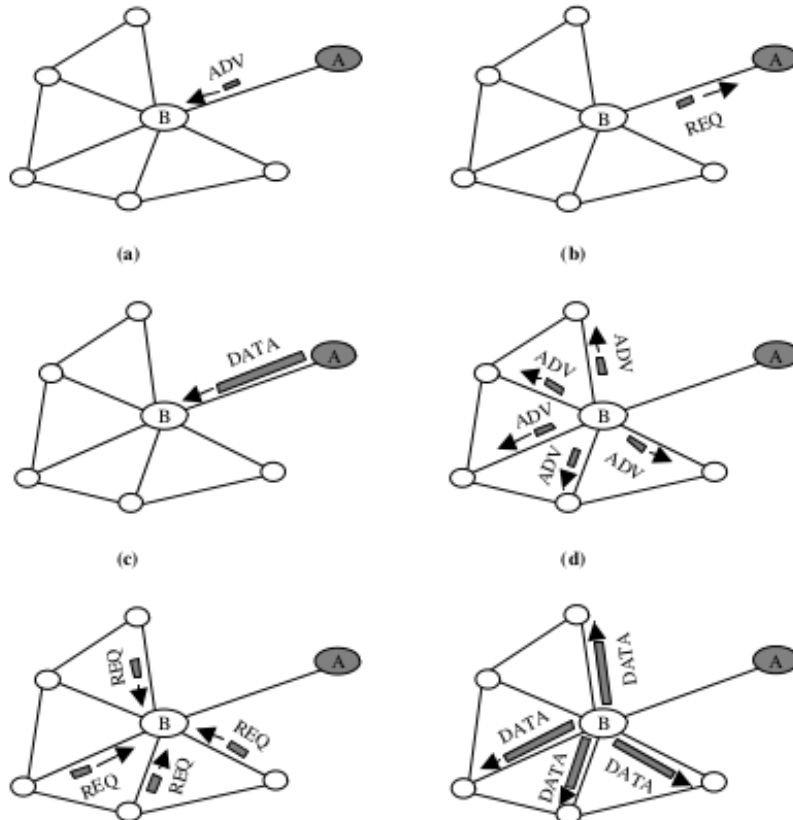


Figura 3.3: Protocolo SPIN [23].

O *Direct Diffusion* (DD) [24] introduz um método de procura através da propagação de interesses e criação de gradientes construídos à medida que um determinado percurso vai sendo utilizado cada vez mais utilizado.

O *Ad hoc On-demand Vector Routing* (AODV) [25] introduz o conceito da descoberta de caminhos e da persistência dos mesmos de forma distribuída por todos os nós. É um protocolo *On-demand* que só entra em acção quando é necessário enviar uma nova mensagem e com mecanismos de *Self-healing* que permitem recuperar um caminho quando por alguma razão existiu uma alteração de topologia. Apresenta duas fases, uma de descoberta de caminho e outra de utilização desse caminho.

Em [26] é abordado o *Dynamic Source Routing* (DSR) semelhante ao AODV tem como objectivo diminuir a largura de banda consumida pelas mensagens de controlo e necessidade de manutenção através de *beacons*. O percurso é guardado na mensagem e vai sendo actualizado à medida que, na fase de descoberta de caminho, esta vai passando em cada nó.

Hierarchical-routing

No trabalho [27] é proposto o *Low-Energy Adaptive Clustering Hierarchy* (LEACH), um protocolo baseado em *clusters*, que usa coordenação entre nós e através de uma mudança aleatória do *cluster-head* distribui de forma eficiente o consumo de energia por todos os nós. Este protocolo consegue reduzir o consumo de energia até oito vezes menos que outros protocolos hierárquicos. O facto dos nós estarem agrupados em *clusters* permite que a informação dos diversos nós não coordenadores possa ser agregada antes de ser enviada para uma BS. Como desvantagens tem o facto de não ser aplicável em redes de grande área, tem um *overhead* extra de mensagens controlo e assume que todos os nós iniciam o seu funcionamento com a mesma energia e que tanto um nó coordenador como um nó simples consomem a mesma energia. Na Figura 3.4 está um exemplo de aplicação.

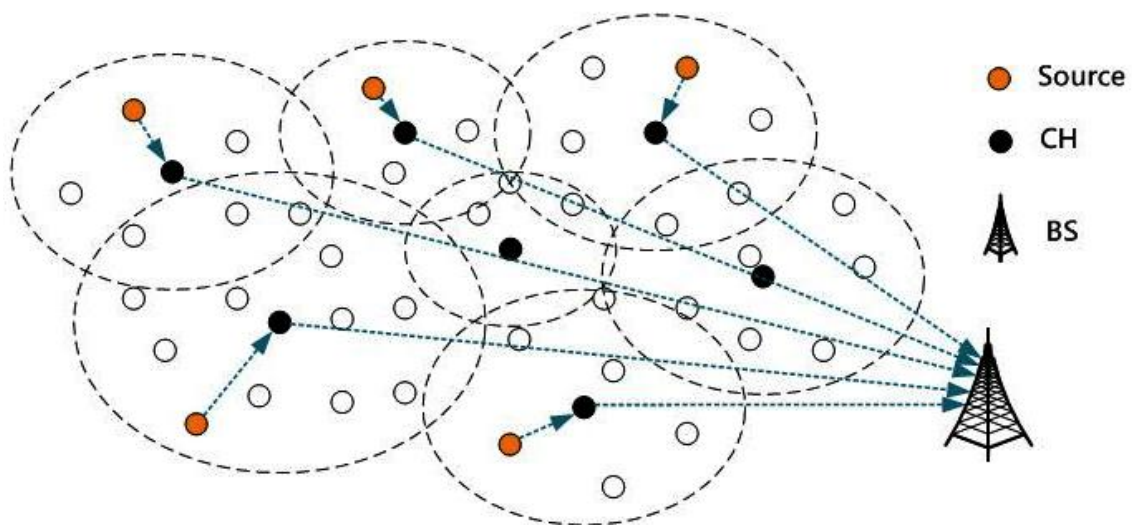


Figura 3.4: Exemplo de funcionamento do protocolo LEACH.

3. Trabalho Relacionado

Outro protocolo hierárquico é o *Power-Efficient Gathering in Sensor Information Systems* (PEGASIS) [28] que surge como um melhoramento do LEACH. Este protocolo aumenta o tempo de vida de cada nó usando técnicas colaborativas, onde cada nó fala apenas com o seu vizinho mais próximo e transmite alternadamente para a BS, eliminando assim a necessidade de formação de *clusters* de forma dinâmica e existência de vários nós coordenadores. Como desvantagens o facto de se assumir que todos os nós conseguem comunicar com a BS directamente, que os nós têm o mesmo nível de energia e podem desligar-se ao mesmo tempo ou a possibilidade do coordenador único se tornar um *bottleneck* no sistema.

Geographic-based Routing

Em [29] é abordado o *Geographical and Energy Aware Routing* (GEAR). Este protocolo surge em redes com um número elevado de sensores e onde poderão ser feitas consultas a determinadas zonas geográficas da rede, sem que tal seja feito com recurso a *flooding*. São utilizadas heurísticas baseadas na energia dos nós e informação sobre a sua posição para encaminhar um pacote para uma determinada região.

4

Ambiente de Trabalho

Conteúdo

4.1	Objective Modular Network Test-bed (OMNeT++)	28
4.2	Mixed Simulator (MiXiM) para OMNeT++	31
4.3	Simulação de Obstáculos para MiXiM	34
4.4	Simulação da Propagação do Sinal	36

4. Ambiente de Trabalho

As WSNs são compostas por inúmeros sensores wireless munidos de reduzidas capacidades de processamento, comunicação e armazenamento. Antes da implementação de aplicações que recorram a sensores wireless e respectiva arquitectura base (por exemplo, o TinyOS [30]) em aplicações reais, torna-se necessário avaliar a eficiência e robustez das mesmas, recorrendo a simulações que englobem tanto a componente aplicacional do nó como a rede no seu todo.

Nesta dissertação sugere-se a criação de um ambiente de trabalho que resulta da utilização conjunta de três sistemas: a *Objective Modular Network Test-bed* (OMNeT++) [31], uma *framework* base de simulação por módulos, o *Mixed Simulator* (MiXiM) [32], uma união de várias *frameworks* para OMNeT++, vocacionadas para a simulação de sensores wireless e um componente de simulação de obstáculos para o MiXiM [33].

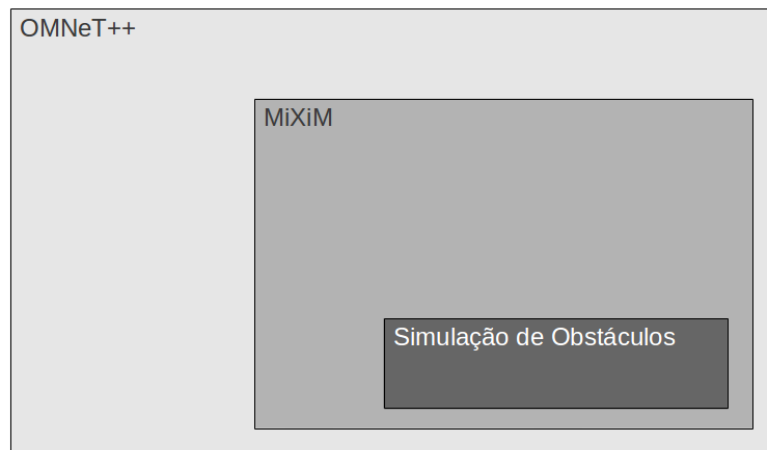


Figura 4.1: Representação modular do ambiente de trabalho.

4.1 Objective Modular Network Test-bed (OMNeT++)

O OMNeT++¹ é uma plataforma de simulação baseada em módulos, escrita em C++ e com um IDE baseado em Eclipse.

O OMNeT++ foi a plataforma base escolhida para este trabalho pelas seguintes razões:

- A partilha dos resultados deste trabalho com a comunidade OMNeT++, promovendo a continuidade do trabalho efectuado nesta dissertação;
- A reutilização e combinação de módulos já construídos;
- A orientação por objectos que permite uma flexível extensão das classes base;
- A existência de um ambiente gráfico automático para uma melhor visualização e *debug* da simulação;

¹<http://http://www.omnetpp.org/>

- A biblioteca extensa incluída que oferece suporte para estatística, colecção de dados, apresentação gráfica, números aleatórios e estruturas de dados;
- A possibilidade de simular vários cenários mudando apenas parâmetros num ficheiro de configuração, sem necessidade de nova compilação.

Cada módulo pode ser do tipo simples ou composto. Os módulos compostos são constituídos por módulos simples ou por outros módulos compostos criando assim uma estrutura hierárquica de dependência. Todos os módulos assentam sobre um módulo de sistema, responsável pela realização da simulação. A comunicação entre módulos é feita através do envio de mensagens, que podem ser tão especializadas quanto necessário e enviadas por canais de comunicação de entrada e saída. Na Figura 4.2 está um diagrama exemplificativo desta arquitectura.

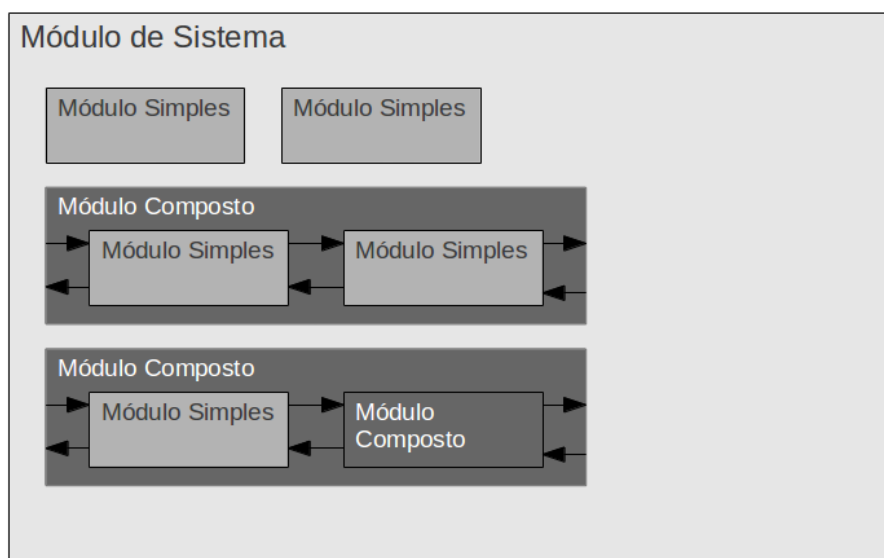


Figura 4.2: Estrutura modular do OMNeT++.

A topologia de cada módulo e a forma como interliga com outros, é descrita utilizando a linguagem *Network Description* (NED) sendo posteriormente a implementação feita em C++. É utilizado um ficheiro de configuração (ex: `omnetpp.ini`) que permite criar diversos cenários possíveis definindo para cada um, por exemplo, parâmetros dos módulos, tempo de simulação, *seed* para números aleatórios, etc. Esta solução permite a utilização de apenas um executável para diversas cenários.

Na Figura 4.3 é possível observar a estrutura interna de um executável OMNeT++.

A *Model Component Library* contém o código compilado dos módulos simples e compostos. Os módulos são instanciados e o *Simulation Model* é construído pelo *Simulation Kernel* (SIM) no início da execução. A simulação é então executada num ambiente definido pelo utilizador que pode ser um dos disponibilizados no OMNeT++ (*Tkenv* ou *Cmdenv*) ou outro (ambiente criado pelo utilizador ou embebido noutra aplicação). Para cada ambiente podem escolhidos fichei-

4. Ambiente de Trabalho

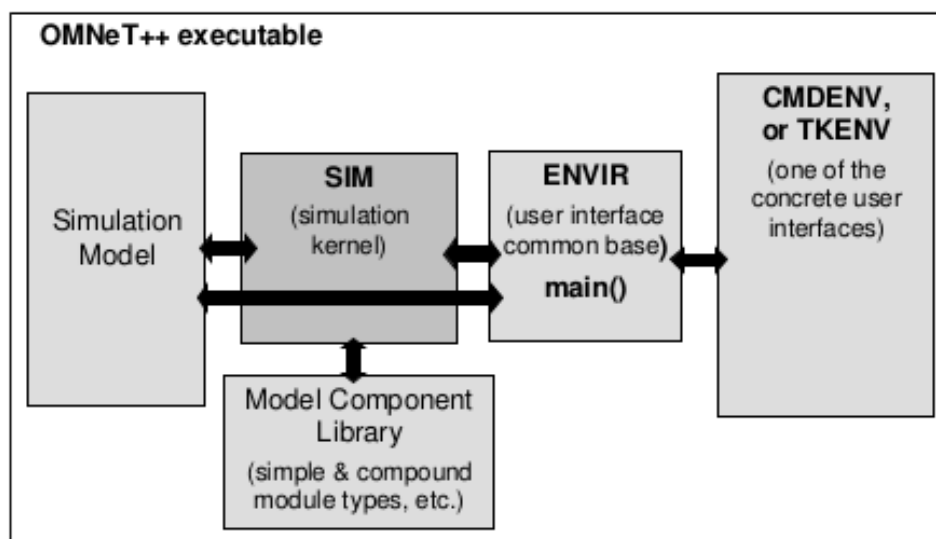


Figura 4.3: Arquitetura lógica de um executável OMNeT++ [31].

ros de configuração (*.ini) e cenários definidos em cada ficheiro de configuração. O ambiente *Cmdenv* corre na linha de comandos de forma rápida enquanto que o ambiente *Tkenv* fornece um ambiente gráfico capaz de animar de forma automática o percurso das mensagens ou as posições dos nós (Figura 4.4).

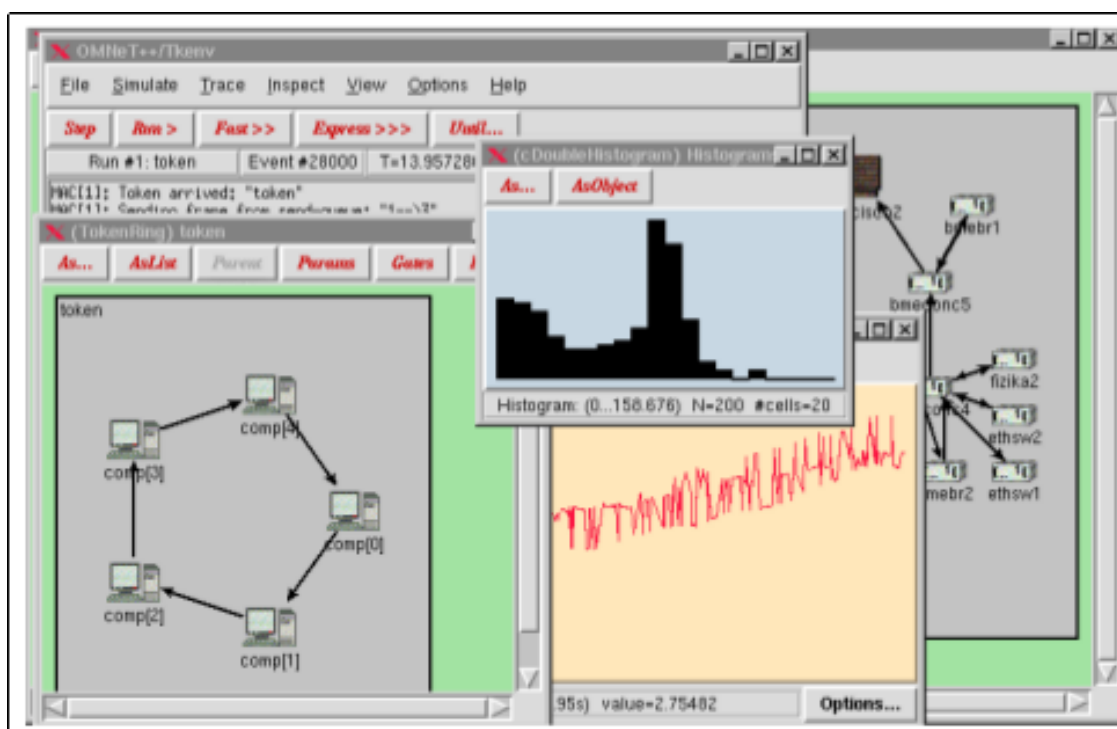


Figura 4.4: Ambiente de simulação Tkenv no OMNeT++ [31].

4.2 Mixed Simulator (MiXiM) para OMNeT++

O MiXiM² resulta da combinação de quatro *frameworks*: a *Mobility Framework* (MF) que introduz suporte à mobilidade, o *CHannel SIMulator* (ChSim) que adiciona modelos detalhados de propagação, o MAC Simulator e a Positif Framework que adicionam o MAC. Esta plataforma foi criada especificamente para simulação de redes wireless introduzindo várias novidades úteis na simulação de WSNs, tais como:

- Módulos para sensores wireless com diversas camadas e simulação de bateria;
- NICs de sensores wireless existentes no mercado (Texas Instruments CC1100 e CC2420);
- Novos modelos de propagação de sinal, como por exemplo o *Two-Ray Ground Path Loss* ou o *Log-normal Shadowing*;
- A possibilidade de ter na mesma simulação vários canais para diferentes frequências o que permite ter na mesma simulação comunicação WI-FI e GSM.
- A decisão da qualidade do sinal e sua recepção feita pelo nó receptor;
- Novos módulos de mobilidade.

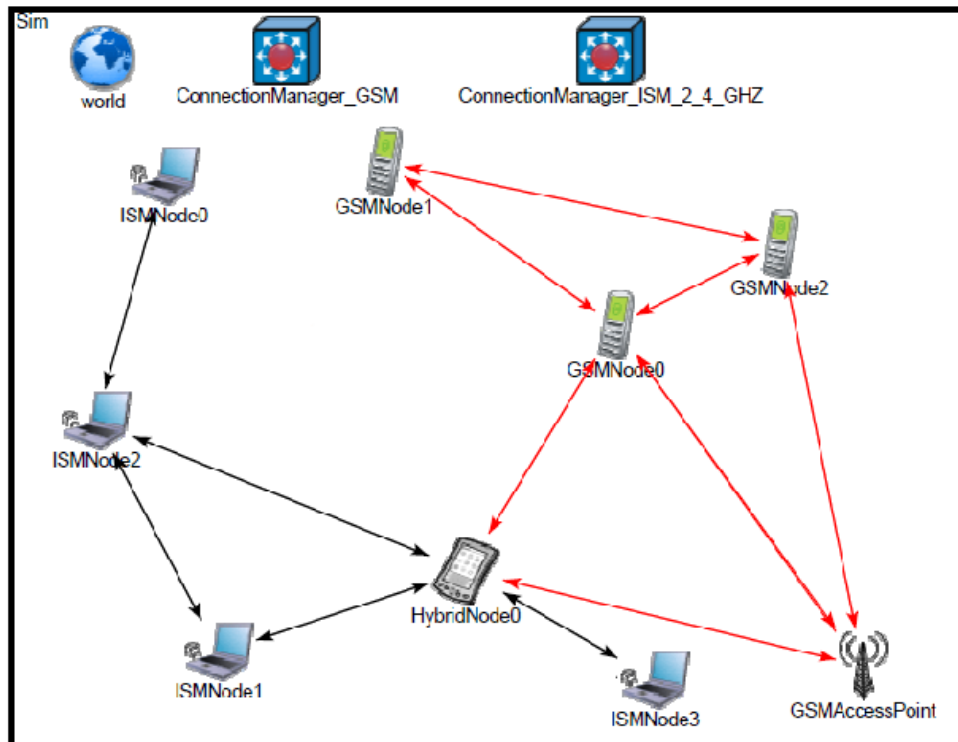


Figura 4.5: Simulação de uma rede no MiXiM [32].

Na Figura 4.5 temos a estrutura do MiXiM que pode ser dividido em dois tipos de módulos:

²<http://mixim.sourceforge.net/>

4. Ambiente de Trabalho

- Módulos de Simulação: módulo *world* responsável pela configuração do ambiente (dimensões da área de trabalho, gestão de parâmetros globais) e *ConnectionManager* responsável pela gestão das ligações entre nós. De notar que o MiXiM suporta vários nós de ligação, tantos como os canais de transmissão existentes;
- Módulos de Nó: módulos com vários sub-módulos que implementam cada uma das camadas lógicas e físicas presentes num nó de uma rede wireless.

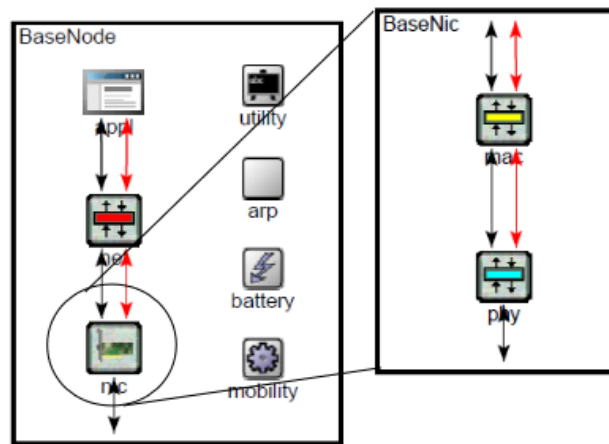


Figura 4.6: Módulo de nó no MiXiM [32].

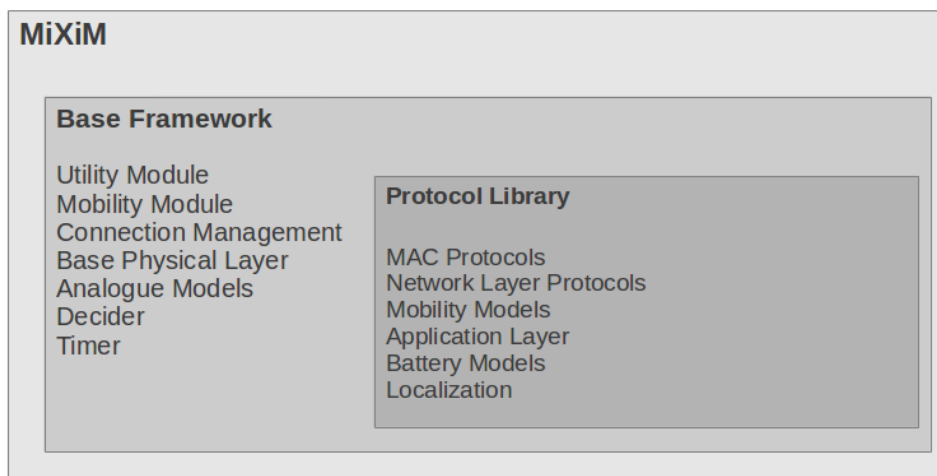


Figura 4.7: Divisão lógica da *framework* MiXiM.

Na Figura 4.6 observa-se em detalhe o módulo de nó onde estão presentes as camadas lógicas de um sensor wireless, o NIC constituído pelas PHY e MAC, a camada *Network* (Netw) e a camada de aplicação. Existem ainda paralelamente vários sub-módulos, nomeadamente o *mobility* que trata da posição e movimentação do nó na área de trabalho, o *battery* que simula o

consumo de energia, o *arp* que trata do endereçamento e o *utility* que serve para efeitos utilitários na partilha de informação durante a simulação.

O MiXiM pode ser dividido de forma lógica numa plataforma base e numa biblioteca de protocolos conforme se pode observar na Figura 4.7. A plataforma base tem todos componentes necessários para criar uma simulação. A biblioteca de protocolos tem diversas extensões da plataforma base que permitem diversificar a quantidade de protocolos e modelos existentes.

Importa analisar como funciona a camada PHY no MiXiM. A potência do sinal é influenciada pelo canal de propagação, influência que pode ser modelada por atenuações causadas por efeitos de *path loss*³, *shadowing*⁴ e *fading*⁵. Para além disso também a frequência do sinal, a potência de envio e o *bit-rate* (modulação e codificação) no tempo, espaço e frequência podem afectar a potência do sinal recebido.

Para modelar este complexo processo o MiXiM implementa uma classe especial de sinal que é associada a cada mensagem e implementa a camada PHY tal como esquematizado na Figura 4.8.

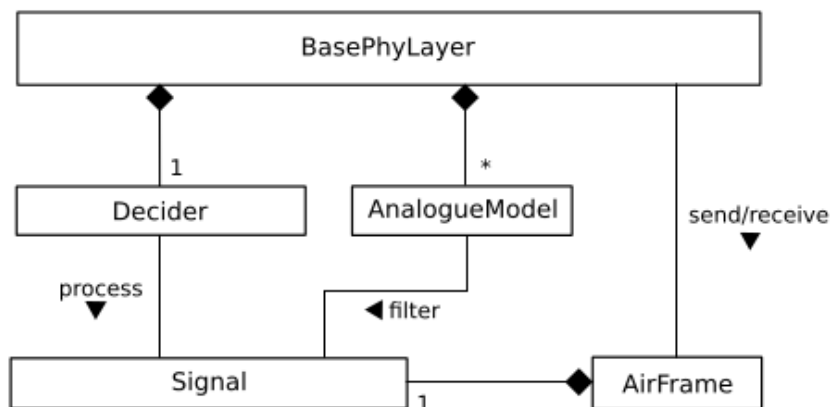


Figura 4.8: Camada PHY do MiXiM [32].

Quando recebe uma mensagem vinda do exterior (*AirFrame*), a camada PHY envia a mensagem para o modelo analógico que irá calcular a atenuação do sinal e para o *Decider* que verifica se o sinal é ruído (com base na potência recebida) e calcula os bit-errors. Depois deste momento a camada PHY calcula o atraso de propagação e de transmissão da mensagem cabendo à camada MAC determinar com base nos *bit errors* se a mensagem é válida ou não.

A título de curiosidade, no trabalho [34] é apresentada uma descrição mais pormenorizada sobre todo o funcionamento desta camada no MiXiM.

³Atenuação causada pelo ar.

⁴Atenuação causada por obstáculos.

⁵Atenuação causada pela multi-propagação de um sinal derivada de diversas reflexões.

4.3 Simulação de Obstáculos para MiXiM

Embora esteja referida em [32], a simulação de obstáculos, como parte integrante do MiXiM, nunca chegou a ser implementada. Assim foi necessário procurar uma solução que permitisse simular a existência de obstáculos no ambiente de simulação.

O trabalho [33] implementa a simulação de obstáculos no MiXiM e a sua representação no ambiente Tkenv. O modelo descrito não contempla efeitos de reflexão ou difracção e pretende ser computacionalmente rápido. A configuração dos obstáculos é feita através de um ficheiro XML. Na Listagem 4.1 está o código XML necessário para desenhar o obstáculo da Figura 4.9.

```
1 <obstacles>
2   <poly id="wall#0" type="brickWall20cm" color="#F00" shape="12.8,10 13,10 13,15 12.8,15" />
3 </obstacles>
```

Listagem 4.1: Exemplo de configuração XML de obstáculos.

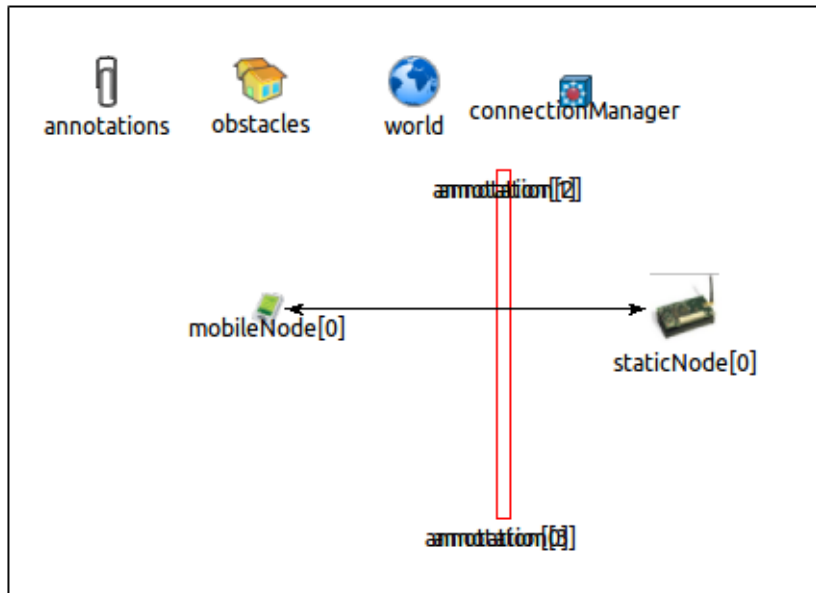


Figura 4.9: Simulação de uma rede MiXiM com obstáculos.

Para melhor perceber a solução encontrada analisa-se a base matemática do modelo.

A potência de um sinal recebido é dada por:

$$P_r[dBm] = P_t[dBm] + G_t[dB] + G_r[dB] - \sum L_x[dB] \quad (4.1)$$

onde P é potência, G é ganho e L_x são os termos que traduzem as perdas. No MiXiM as perdas já são calculadas como descrito na secção anterior pela camada PHY mas não contemplam as perdas decorrentes da existência de obstáculos.

É então sugerido um termo para este tipo de perdas da seguinte forma:

$$L_{obs}[dB] = \beta n + \gamma d_m \quad (4.2)$$

Os termos β e γ são calculados com base nos resultados obtidos experimentalmente e representam a atenuação por metro e a atenuação por parede respectivamente. Para o trabalho [33] os valores obtidos foram $\beta \approx 9\text{dB}$ e $\gamma \approx 0.4\text{dB/m}$.

Devido à não utilização de nós reais neste trabalho, por condicionantes relacionadas com o custo envolvido, que permitissem chegar a valores reais, optou-se por considerar, com base no Tabela 4.2 do manual da *3Com Wireless Antenas*⁶, os seguintes valores:

Profundidade(cm)	$\beta(\text{dBm})$	$\gamma(\text{m})$
20	106.3	0
10	26.575	0

Tabela 4.1: Valores de atenuação por metro e por parede usados neste trabalho.

BUILDING MATERIAL	5GHZ ATTENUATION (dBi)	2.4GHZ ATTENUATION (dBi)
Solid Wood Door 1.75 "	10	6
Hollow Wood Door 1.75 "	7	4
Interior Office Door w/Window 1.75 "/0.5 "	6	4
Steel Fire/Exit Door 1.75 "	25	13
Steel Fire/Exit Door 2.5 "	32	19
Steel Rollup Door 1.5 "	19	11
Brick 3.5 "	10	6
Concrete Wall 18 "	30	18
Cubical Wall (Fabric) 2.25 "	30	18
Exterior Concrete Wall 27 "	45	53
Glass Divider 0.5 "	8	12
Interior Hollow Wall 4 "	3	5
Interior Hollow Wall 6 "	4	9
Interior Solid Wall 5 "	16	14
Marble 2 "	10	6
Bullet-Proof Glass 1 "	20	10
Exterior Double Pane Coated Glass 1 "	20	13
Exterior Single Pane Window 0.5 "	6	7
Interior Office Window 1 "	6	3
Safety Glass-Wire 0.25 "	2	3
Safety Glass-Wire 1.0 "	18	13

Tabela 4.2: Atenuação de materiais de construção comuns para frequências de 5GHz e 2.4 GHz.

⁶<http://www.scribd.com/doc/32613170/3Com%C2%AE-Wireless-Antennas>

4.4 Simulação da Propagação do Sinal

A escolha de um modelo adequado de propagação do sinal assume particular importância, dado que para métodos de localização probabilísticos a inferência sobre a localização de um determinado nó, será feita a partir de funções densidade de probabilidade, que dependem exactamente das características da propagação do sinal.

Se um modelo demasiado simples fosse usado, poderia criar resultados diferentes do esperado e que não correspondessem a uma situação real.

Habitualmente a propagação de um sinal num ambiente interior é afectada por três factores:

- *Path-loss*: perdas resultantes da atenuação causada pela distância de propagação;
- *Shadowing* : perdas que resultam de flutuações aleatórias devido a obstruções;
- *Fading* : perdas resultantes da existência de reflexões que provocam a multi-propagação do sinal.

Neste trabalho, são simuladas a *Path-Loss* e a *Shadowing* usando para isso modelos disponíveis no MiXiM para simular este comportamento: o *SimplePathlossModel* e o *LogNormalShadowing*.

O *SimplePathLossModel* implementa um modelo simples de perdas, com atenuação dada por:

$$L_{pathloss} = \frac{\lambda^2}{16\pi^2} d^{-2\alpha} \quad (4.3)$$

onde: d é a distância percorrida, $\lambda = \frac{c}{f}$ e α é o coeficiente de perdas.

Por sua vez o *LogNormalShadowing* é implementado recorrendo a uma distribuição normal para gerar uma atenuação, em que os valores da média e do desvio padrão são calculados de forma aleatória.

5

Arquitectura do Sistema

Conteúdo

5.1	Sistema de Monitorização EMoS	38
5.1.1	Nó Móvel (MN)	40
5.1.2	Nó Fixo (SN)	41
5.1.3	Nó Base (BN)	41
5.2	Camada <i>Network</i> : <i>AODVRoute</i>	43
5.2.1	Tipos de Mensagens	43
5.2.2	Estruturas de Dados	44
5.2.3	Modo de Funcionamento	45
5.3	Camada <i>Application</i> : Localização	51
5.3.1	HORUS Modificado	51
5.3.2	Modelo Matemático	53
5.3.3	Estruturas de Dados e Ficheiros XML	56
5.3.4	Modo de Funcionamento	56
5.4	Mobilidade	61

5.1 Sistema de Monitorização EMoS

Propõe-se nesta tese o *Elder Monitoring System* (EMoS), uma solução simulada para o problema da monitorização de pessoas em ambiente doméstico. O EMoS é uma rede WSN constituída por diversos nós com comunicação wireless colocados de forma homogénea numa casa. Embora o sistema possa efectuar a monitorização de todo o tipo de pessoas, neste trabalho é focada a monitorização de idosos ou pessoas com necessidades especiais.

São sugeridas opções de hardware comercialmente disponíveis para cada componente do sistema, pretendendo-se desta forma ir para além da simples simulação e obter parâmetros reais para a configuração da mesma. Alguns aspectos de hardware mencionados não serão simulados por limitação de tempo na execução deste trabalho e também por não corresponderem ao âmbito estabelecido no Capítulo 1.2.

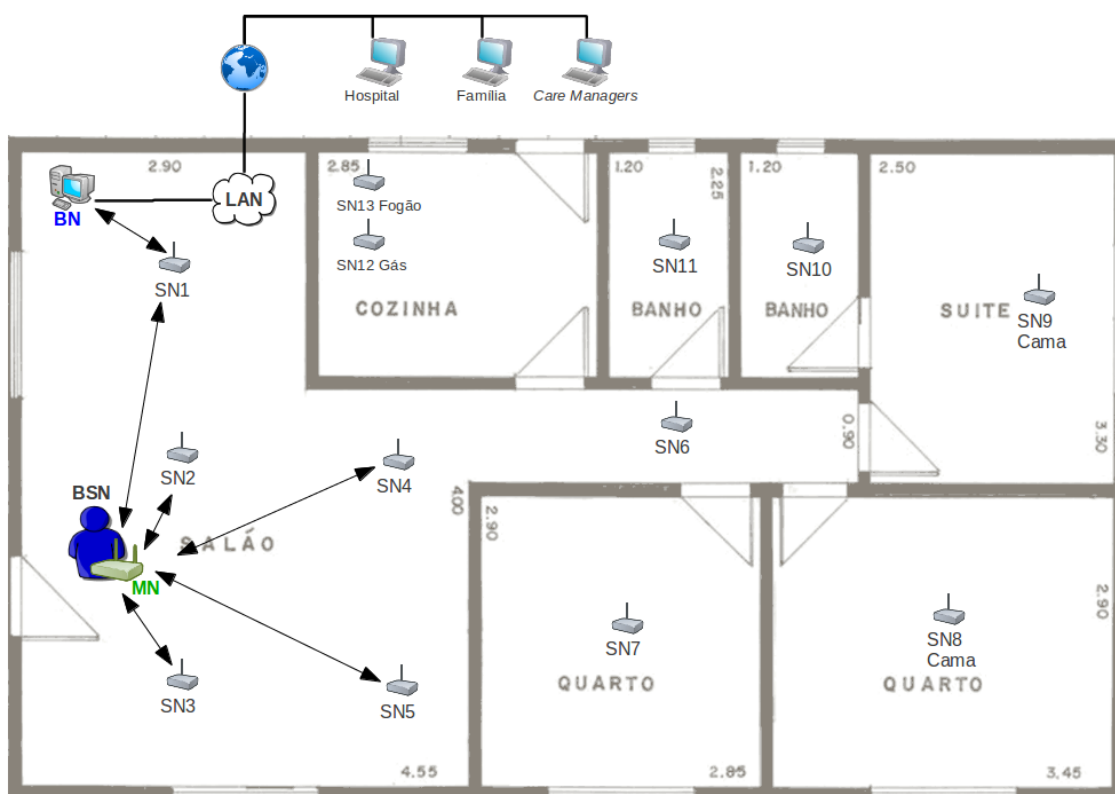


Figura 5.1: Esquema modular do *Elder Monitoring System* (EMoS).

O sistema EMoS (Figura 5.1) é caracterizado pelos seguintes tipos de dispositivos:

- Nó Móvel (*Mobile Node* (MN)): Monitorização de pessoas e calibração do sistema;
- Nó Fixo (*Static Node* (SN)): Envio de *beacons* para localização, comunicação e monitorização doméstica;
- Nó Base (*Base Node* (BN)): Gestão central do sistema recebendo os dados de monitoriza-

ção ou pedidos do utilizador e enviando mensagens para a rede WSN ou para o exterior através de uma LAN.

Os nós fixos estão ligados à rede eléctrica e enviam periodicamente mensagens de assinatura em modo *broadcast*. No caso dos nós *SN8*, *SN9*, *SN12* e *SN13*, pode existir também envio de mensagens para outros nós fixos ou para o nó base. Isto pode acontecer quando por exemplo, é detectado gás no caso do *SN12*, quando é ligado/desligado o fogão no caso do *SN13* ou quando alguém se deita/levanta numa das camas onde estão os sensores *SN8* e *SN9*.

O nó base recebe informação dos nós fixos ou do exterior através da LAN e toma decisões com base nessa informação. Pode, por exemplo, avisar num monitor que existe alguma anomalia na casa, enviar uma mensagem para o nó móvel ou para um nó fixo, ou comunicar com o exterior caso seja necessário.

O nó móvel pode funcionar em dois modos: calibração e normal. No modo de calibração limita-se a receber assinaturas dos nós fixos e a registar essa informação, gerando um ficheiro XML com o mapa rádio do casa. No modo normal recebe de igual forma as assinaturas, mas periodicamente envia de volta para o nó fixo mais próximo, um conjunto de médias das potências recebidas. O nó fixo por sua vez envia esta informação para o nó base que irá calcular a localização do nó móvel.

Todos os nós formam uma estrutura em *flat-routing* usando o AODV para comunicar e apresentam uma estrutura interna idêntica (Figura 5.2). O sistema é perfeitamente escalável, sendo possível adicionar-se vários outros nós móveis. No caso de existir um número muito elevado de nós móveis, é possível criar novos nós base associados a *clusters* de nós fixos, que comuniquem através da LAN para eliminar sobreposições.

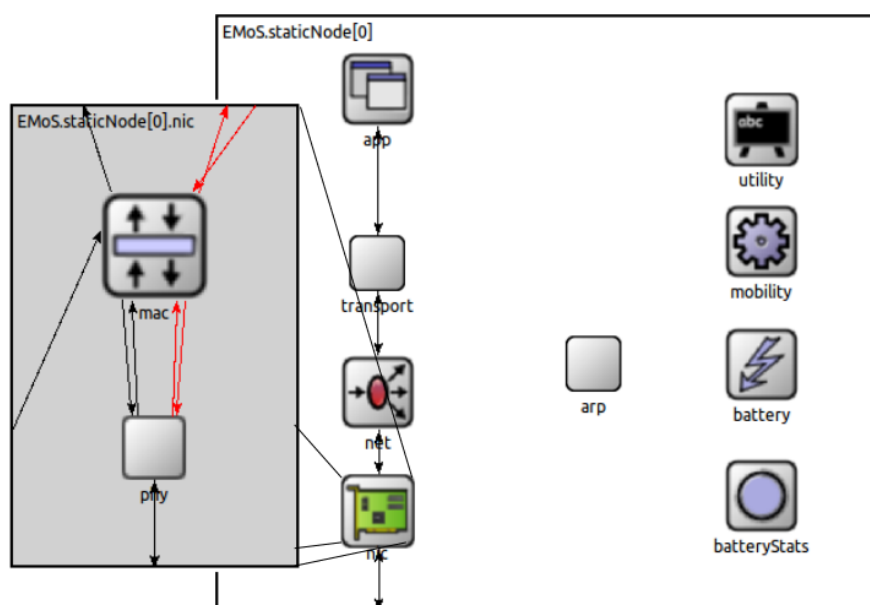


Figura 5.2: Estrutura interna de um nó no sistema EMoS.

5. Arquitectura do Sistema

Neste trabalho foi criada uma camada *Network* comum para todos os nós e uma camada *Application* por cada tipo de nó que implementa cada um dos tipos descritos. Nas próximas secções é feita uma exposição detalhada de todo o sistema.

5.1.1 Nó Móvel (MN)

O nó móvel é um sensor equipado com rádios IEEE 802.15.4 e Bluetooth, um acelerómetro, um botão de pânico e capacidade de armazenamento em cartão SD. O facto de ter dois tipos de rádio permite-lhe comunicar com ambas as redes BSN para monitorização local biomédica e WSN. Está instalado num objecto de uso diário, como uma bengala, um andarilho ou uma cadeira de rodas e possui baterias recarregáveis. O carregamento é feito no sítio habitual de apoio da bengala ou então por ligação com as baterias de uma cadeira de rodas eléctrica.

A rede BSN é composta por um electrocardiógrafo e um medidor de pressão arterial ambos equipados com Bluetooth, que efectuem monitorização contínua. Quando detectam uma situação anómala comunicam a mesma ao nó móvel. Na simulação este comportamento é recreado com um temporizador aleatório no nó móvel que simula a comunicação vinda da BSN ao nível da aplicação.

Como solução de hardware sugere-se a utilização do módulo *Waspmote* (Figura 5.3, com rádio *XBee-802.15.4* (Figura 5.4), da *Libelium*¹ que permite a utilização de dois tipos de rádio, tem acelerómetro integrado, microSD e pins para obtenção do RSSI.



Figura 5.3: Módulo *Waspmote* da *Libelium*.

¹<http://www.libelium.com/products/waspmote>



Figura 5.4: Rádio *XBee* IEEE 802.15.4 usado no módulo *Waspmote*.

O rádio *XBee* simulado terá os seguintes parâmetros obtidos do manual ²⁾:

Parâmetro	Valor
Modulation	O-QPSK
Receiver Sensitivity	-92 dBm
Transmit Power	1mW
Sleep Current	<10 μ A
Current Consumption RX	50 mA
Current Consumption TX (P=0dBm)	45 mA

Tabela 5.1: Parâmetros do rádio *XBee* da *Digi*.

5.1.2 Nó Fixo (SN)

O nó fixo está equipado com rádio IEEE 802.15.4. Ficará ligado à rede eléctrica dada a necessidade de estar periodicamente a enviar assinaturas em *broadcast*. O hardware escolhido para a implementação deste nó será o *MicaZ* com rádio *Texas Instruments CC2420* da *Crossbow*³ (Figura 5.5).

A escolha deste nó é motivada pelo baixo custo do mesmo e o facto de ser o nó que estará presente em maior número na rede.

O rádio *Texas Instruments CC2420* tem os seguintes parâmetros obtidos do manual⁴;

5.1.3 Nó Base (BN)

O nó base tem como função efectuar toda a gestão centralizada da informação gerada por cada um dos sensores. Dada a grande necessidade de rapidez de processamento, espaço de

²http://ftp1.digi.com/support/documentation/90000982_H.pdf

³<http://www.xbow.com/>

⁴<http://www.ti.com/lit/gpn/cc2420>



Figura 5.5: Módulo *Micaz* da *Crossbow*.

Parâmetro	Valor
Modulation	O-QPSK
Receiver Sensitivity	-95 dBm
Transmit Power	1.1mW
Sleep Current	0.02 uA
Current Consumption RX	18.8 mA
Current Consumption TX (P=0dBm)	17.4 mA

Tabela 5.2: Parâmetros do rádio *Texas Instruments CC2420*.

armazenamento e comunicação com o exterior, optou-se por usar um PC ligado a uma LAN que está por sua vez ligada à Internet. Assim o hardware para a implementação física deste nó e que fará a ponte entre o PC e a rede WSN, será o *Wasmote Gateway*, com rádio *XBee-802.15.4* (Figura 5.4), da *Libelium*.



Figura 5.6: Módulo *Wasmote Gateway* da *Libelium*.

Este nó usa o mesmo rádio que o *Wasmote*, com parâmetros para simulação já registados na Tabela 5.1.

5.2 Camada Network : AODVRoute

A tecnologia ZigBee é hoje bastante usada nas redes WSN com vários produtos para as mais diversas áreas. O ZigBee usa como primeiro protocolo na camada *Network* o *Ad hoc On-demand Vector Routing* (AODV) e seguidamente quando este falha um protocolo hierárquico. Assim por forma a garantir uma compatibilidade do sistema desenvolvido com o ZigBee optou-se por implementar este protocolo no MiXiM.

O AODV é um protocolo *on-demand* que permite descobrir um caminho apenas quando este é necessário, recuperar caminhos perdidos e reutilizar caminhos já encontrados. Achou-se por isso conveniente implementá-lo no MiXiM para ser usado no EMoS. Este usa caminhos bidireccionais, o que significa que quando é feita a descoberta de um novo caminho são sempre gerados dois, acelerando o processo de procura de novos caminhos. A utilização de contadores sequenciais impede a formação de loops em todo o processo e é feita manutenção sobre os caminhos para eliminar os que deixaram de ser usados durante um determinado espaço de tempo.

Neste trabalho foi implementada uma versão do AODV que apenas contém as funcionalidades necessárias para descoberta de um novo caminho ou para a recuperação de um caminho perdido. O *multicast* bem como a reparação local de caminhos perdidos (*local-repair*⁵) não foram implementadas. Ao módulo implementado foi dado o nome de *AODVRoute*.

5.2.1 Tipos de Mensagens

O AODV funciona utilizando pelo menos três tipos de mensagens para descobrir o caminho de um nó A para um nó B.

- *Route Request* (RREQ);
- *Route Response* (RREP);
- *Route Error* (RERR).

Pode ainda existir um quarto tipo de mensagem, opcional, o *Route Response Acknowledgement* (RREP-ACK), utilizado quando existe o perigo de existirem ligações unidireccionais que impeçam a chegada de um RREP, mas que não será referido neste trabalho.

Na Figura 5.7 estão as mensagens criadas no MiXiM. A criação de mensagens no OMNeT++ é relativamente simples uma vez que todo o *boilerplate code* é gerado automaticamente. Assim é necessário apenas definir ficheiros do tipo *msg* com os parâmetros necessários. Cada uma das mensagens definidas para este módulo estende a mensagem *NetwPkt*, uma mensagem genérica para a camada *Netw* com endereços de emissor e receptor e *Time To Live* (TTL). Para além dos campos herdados existem também os seguintes campos para cada tipo de mensagem:

⁵reparação que ocorre localmente quando um nó intermédio falha no encaminhamento da mensagem.

5. Arquitectura do Sistema

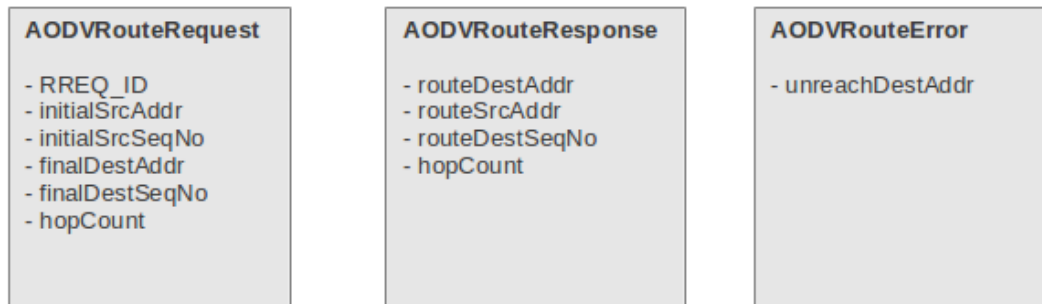


Figura 5.7: Tipos de mensagens do AODVRoute.

- *AODVRouteRequest* : endereços e números de sequência para o nó emissor e receptor, o *RREQ_ID* que é obtido de um contador existente para o efeito no nó emissor e o contador de saltos que permite perceber em que ponto do caminho está o RREQ;
- *AODVRouteResponse* : endereços de emissor e receptor, número de sequência do nó de receptor e número de saltos até ao receptor;
- *AODVRouteError* : endereço do nó de destino que não foi encontrado.

5.2.2 Estruturas de Dados

Foram criadas três estruturas de dados neste novo módulo:

- *pktMap* : utilizada para guardar os pacotes que vieram da camada APP por nó de destino, num FIFO quando para estes ainda não existe um caminho encontrado;
- *RREQVector* : utilizada para guardar os pacotes RREQ recebidos;
- *routeMap* : utilizada para guardar as rotas em cada nó e a lista de precursores⁶ dessa rota.

Na Figura 5.8 observamos a estrutura *PktMap*. Este mapa de pares do tipo de endereços e filas de pacotes, guarda para cada nó de destino os pacotes em espera de um caminho para o seu destino final. Cada elemento da fila tem como parâmetros o endereço do nó de destino, o tempo de vida do elemento ($t_0 + t$) e o pacote que foi enviado da camada APP. Esta estrutura é sujeita a uma manutenção periódica que elimina todos os elementos para os quais $t_{sim} > lifetime$. O tempo de vida é suficientemente curto para que não exista uma disparidade grande entre o momento que foi encontrado um caminho e a nova posição do nó móvel.

Na Figura 5.9 observamos a *RREQVec*, estrutura que guarda a chave pares *RREQ_ID* e *srcAddr*, por forma a distinguir univocamente todos os RREQs recebidos de um determinado nó evitando assim o envio de um RREQ que já foi anteriormente enviado.

⁶Precursores de um caminho, é um nó que antecede o nó actual no caminho e só existe depois de ter enviado pelo menos uma mensagem pelo nó actual.

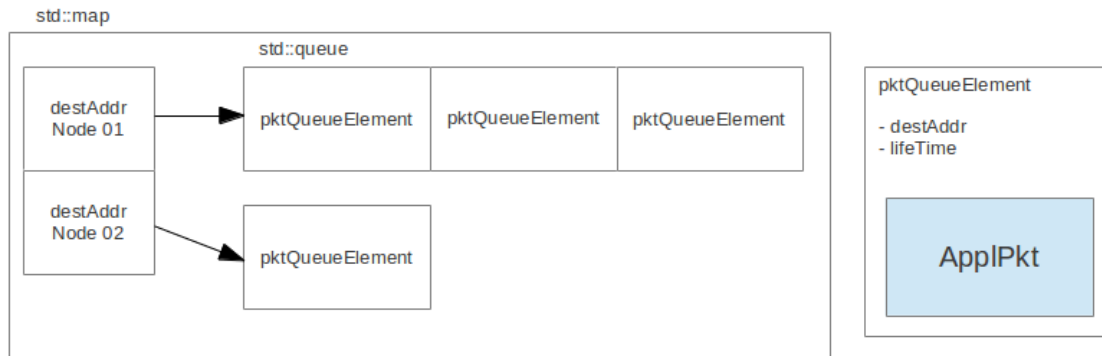


Figura 5.8: PktMap - estrutura para a gestão dos pacotes de aplicação em espera.



Figura 5.9: RREQVector - estrutura que guarda os RREQs recebidos num nó.

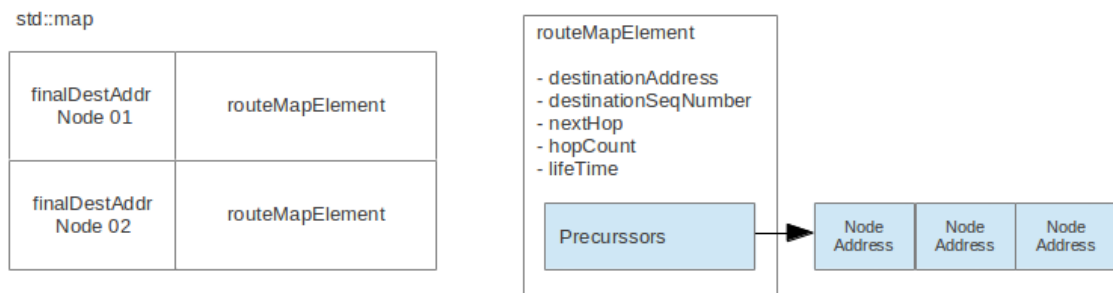


Figura 5.10: RouteMap - estrutura que guarda os caminhos encontrados para nós destino.

Finalmente na Figura 5.10 temos a estrutura *RouteMap* que guarda a informação dos caminhos encontrados em cada nó. Estão presentes em cada *routeMapElement*, o nó final de destino do caminho, o número de sequência conhecido do destino, o próximo nó do caminho, o número de nós entre o nó actual e o nó de destino, o tempo de vida do caminho usado para garantir que os caminhos que deixaram de ser usados não ficam a ocupar espaço no limitado armazenamento disponível e uma lista de nós precursores que servirá para para enviar um RERR de volta para todos os nós que anteriormente usaram o caminho que deixou de existir.

5.2.3 Modo de Funcionamento

Neste trabalho a camada de aplicação poderá gerar vários tipos de mensagens. Essas mensagens poderão ser em *broadcast* no caso das mensagens de assinatura enviadas pelos nós

5. Arquitectura do Sistema

fixos ou *unicast* quando existe um nó que pretende comunicar com outro. Todos os nós do sistema EMoS podem comunicar uns com os outros, no entanto, nem sempre de forma directa. Sendo a camada *Network* comum a todos os nós o comportamento é idêntico para todos.

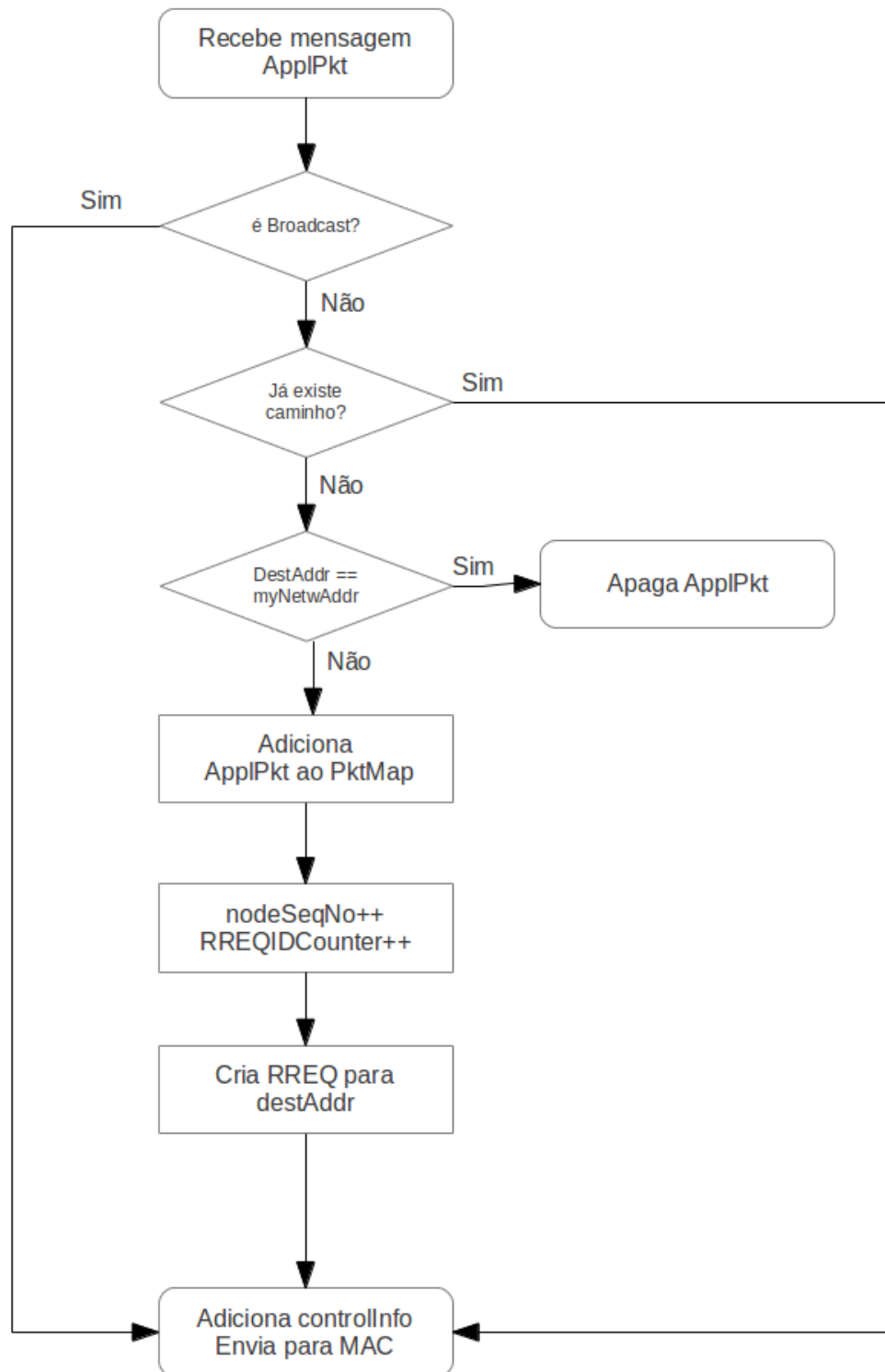
Suponhamos que a camada de aplicação de um nó fixo pretende enviar uma mensagem para o nó base, informando que o existe uma fuga de gás. A partir desse momento a mensagem de aplicação é anexada informação de controlo (*NetwControlInfo*), contendo o endereço *Network* do nó de destino, sendo enviada para a camada abaixo. A mensagem segue então o fluxograma da Figura 5.11. De notar que o caso em que o endereço de destino é igual ao do próprio nó, decorre de um erro de atribuição do endereço final na camada de aplicação. A incrementação da sequência do nó é feita para o caso de ainda existir RREQs perdidos no ambiente.

Cada nó guarda também um registo, para cada nó com o qual já tenha comunicado um registo do número de sequência desse nó. Este valor é colocado no RREQ para que, caso seja encontrado um caminho mais recente, este pedido seja ignorado.

Portanto quando não existe caminho para o nó pretendido de destino, o nó de origem gera o RREQ que é enviado em modo *broadcast* para o resto da rede. Na Figura 5.12 está a recepção de outro nó do pacote enviado pelo nó de origem.

Dado que o nó guarda todos os RREQ que recebe no *RouteVector*, este consegue verificar se já recebeu aquele mesmo pacote ou não. No caso de não ter recebido é incrementado o contador de saltos para que a distância até à origem seja actualizada. A informação do nó de origem, o número de saltos até ao mesmo são usados para criar um caminho inverso para o nó de origem. Enquanto este não for o nó de origem a mensagem vai sendo enviada de nó para nó até chegar ao destino. Pode ainda se dar o caso do nó não ser o nó de destino mas já existir um caminho no *RouteMap*, o que permite ao nó intermédio devolver logo um RREP. Quando se trata do nó de destino, é incrementado o contador do nó para garantir que não haverá mais nenhum caminho que se sobreponha.

Então, assim que o caminho é encontrado, existe um RREP que volta pelo caminho inverso em *unicast* construindo o caminho do nó de origem para o nó de destino iniciais.

Figura 5.11: Fluxograma de chegada da mensagem de aplicação à camada *Network*.

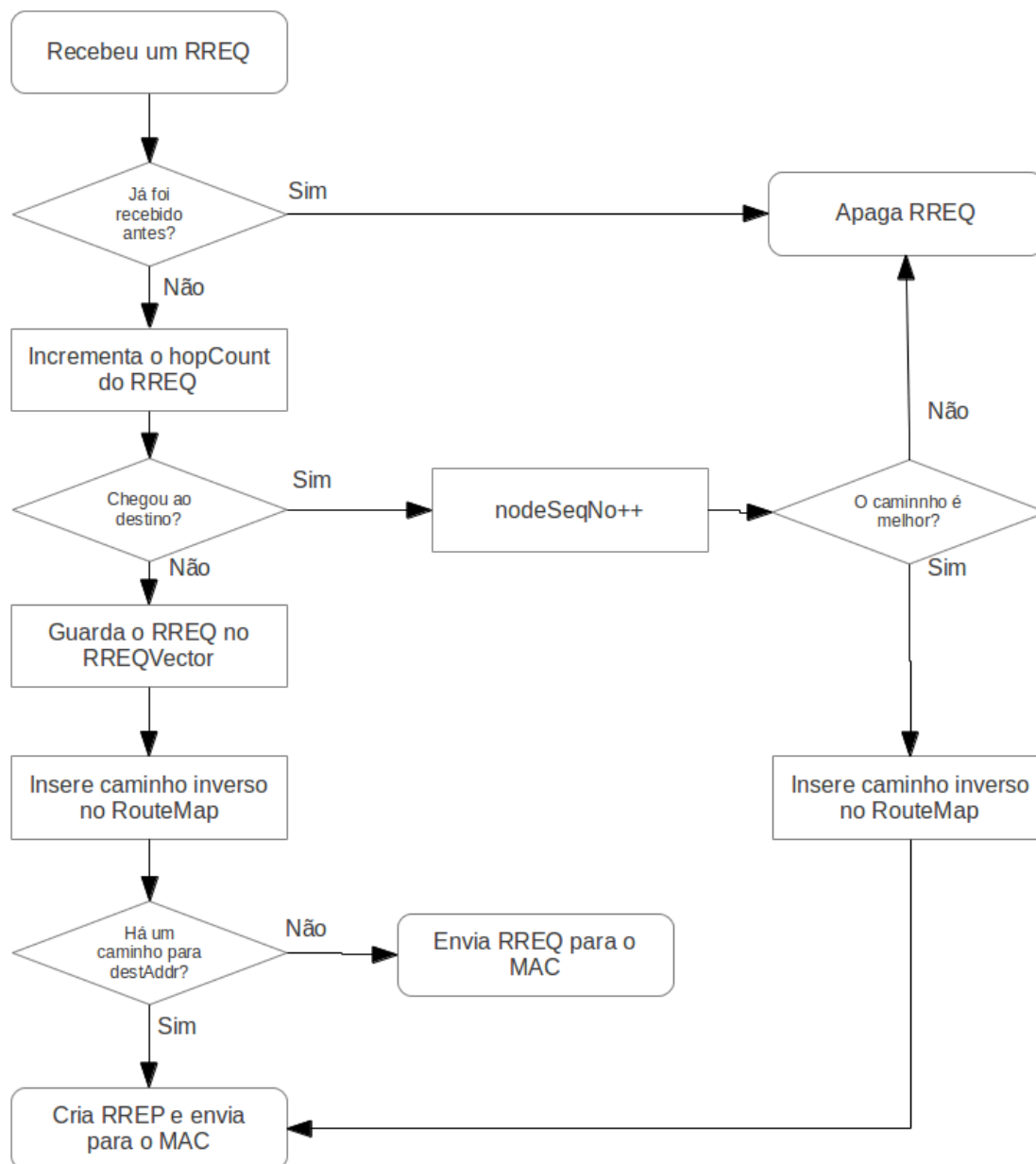


Figura 5.12: Fluxograma de chegada de um *RREQ* vindo da camada *MAC*.

Na Figura 5.13 podemos observar o que acontece a essa mensagem à medida que vai passando pelos nós intermédios até chegar ao nó de origem original. Quando é alcançada a origem, o nó pode então ir buscar ao pktMap, a próxima mensagem e enviá-la para aquele endereço de destino.

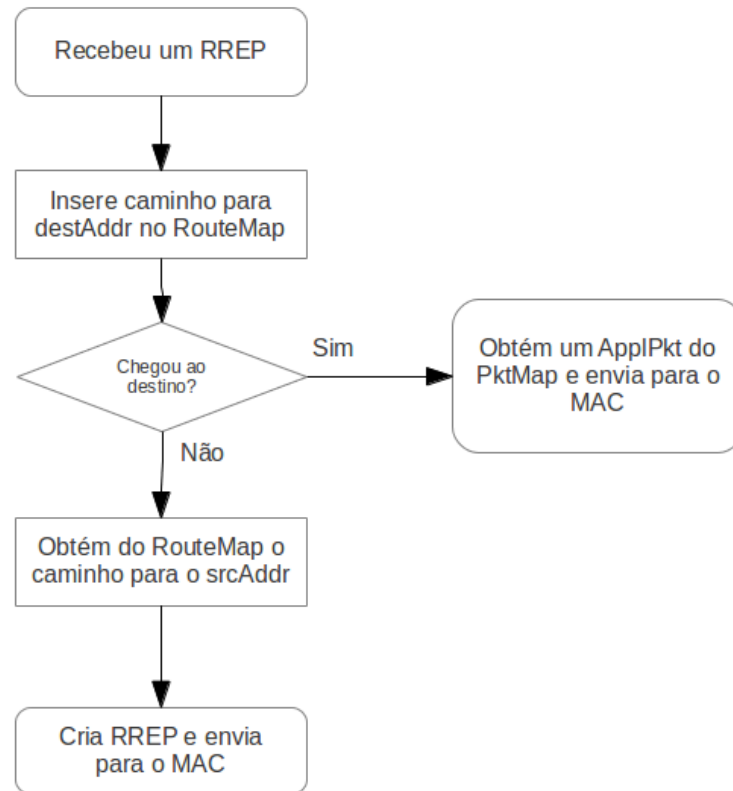


Figura 5.13: Fluxograma de chegada de um *RREP* vindo da camada *MAC*.

Quando ocorre um erro, ou seja, existe uma mensagem que é entregue mas o MAC não recebe uma resposta de confirmação ACK, é necessário tomar medidas para que a origem seja informada que aquele caminho já não é válido. Assim são criados RERR que viajam pelos precursores até atingir nó de origem. Na Figura 5.14 podemos ver a primeira fase quando a falha na entrega da mensagem ocorre. Nesse momento o nó procura todos os caminhos que estejam no *RouteMap* e elimina-os, enviando depois um RERR para cada um dos precursores.

Na Figura 5.15 temos o comportamento dos nós intermédios e destino (origem do caminho eliminado). De notar que a mensagem vai sendo apagada, nó a nó, quando para cada percurso na árvore já não existem mais precursores.

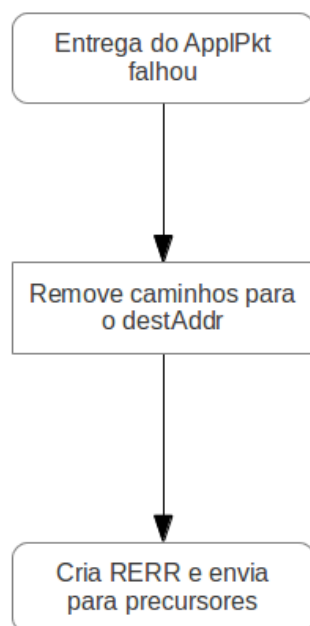


Figura 5.14: Fluxograma de chegada de uma mensagem de controlo com o pacote cuja entrega falhou no .MAC.

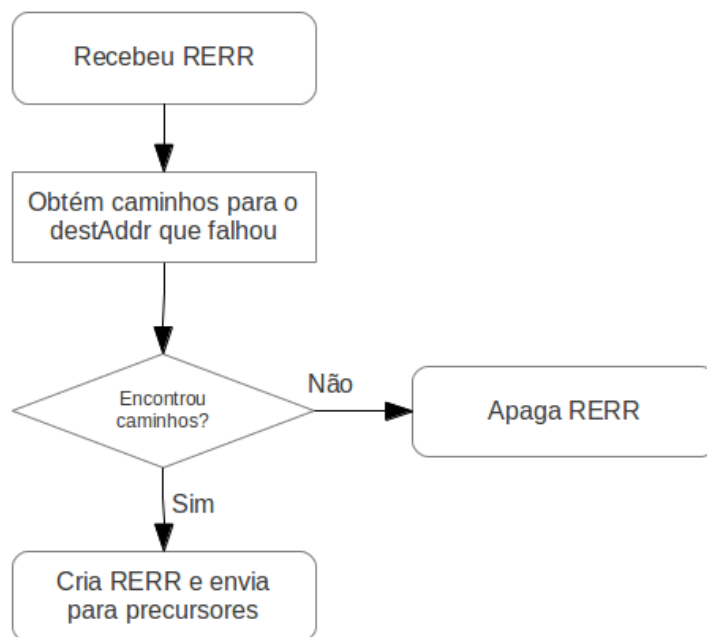


Figura 5.15: Fluxograma da chegada de um *RERR* vindo da camada *MAC*.

5.3 Camada *Application* : Localização

O sistema EMOs usa para a localização o método *HORUS* [21] com algumas modificações que serão descritas na Secção 5.3.1. É um processo probabilístico com precisão elevada, razão pela qual foi escolhido para este trabalho. O processo foi implementando completamente ao nível da camada de aplicação e é constituído por duas fases distintas:

- Fase *Offline* : fase onde é obtido o mapa rádio e criados os *clusters*. O nó móvel (MN) é configurado em modo de calibração e guarda numa estrutura de dados, as assinaturas recebidas vindas dos diversos nós fixos que estão à distância de recepção. No fim da simulação são criados dois ficheiros *XML* que guardam posições ou *clusters*. Esses ficheiros serão posteriormente interpretados pelo nó base.
- Fase *Online* : fase em que os nó móveis enviam periodicamente mensagens com um conjunto de médias das potências recebidas dos vários nós fixos para o nó base. Utilizando o mapa rádio entretanto criado o nó base efectua o cálculo da posição seleccionando primeiramente um *cluster* de posições e depois calculando a probabilidade de cada posição.

5.3.1 HORUS Modificado

O sistema *HORUS* usado no EMOs difere do apresentado pelo seu autor em [21], na medida em que o cálculo da posição não é feito no nó móvel mas sim no nó base, sendo a correlação entre amostras a única operação que permanece no dispositivo móvel. Esta opção de arquitectura deveu-se às seguintes razões:

- Poupança de bateria no nó móvel;
- Fraca capacidade de processamento do nó móvel;
- Necessidade de centralização das posições de vários nós móveis num único sítio podendo assim ser feita a correlação de resultados vindos de todos os nós;
- Complexidade do sistema de localização.

Fase Offline

Na Figura 5.16 é descrito de forma global e modular todo o sistema de localização implementado. Na fase *offline* o nó móvel recolhe para cada posição da casa uma série de amostras (*Raw Data*), pares (nodeAddress,RSSI) obtidos a partir da informação de controlo enviada pela camada *Network*. Esta informação vai sendo guardada, sem qualquer tratamento, para todas as posições registadas.

Depois de concluída a amostragem de todo o espaço a informação é agrupada por posição no *Radio Map Builder* e correlacionada efectuando para cada posição e para cada nó, o cálculo

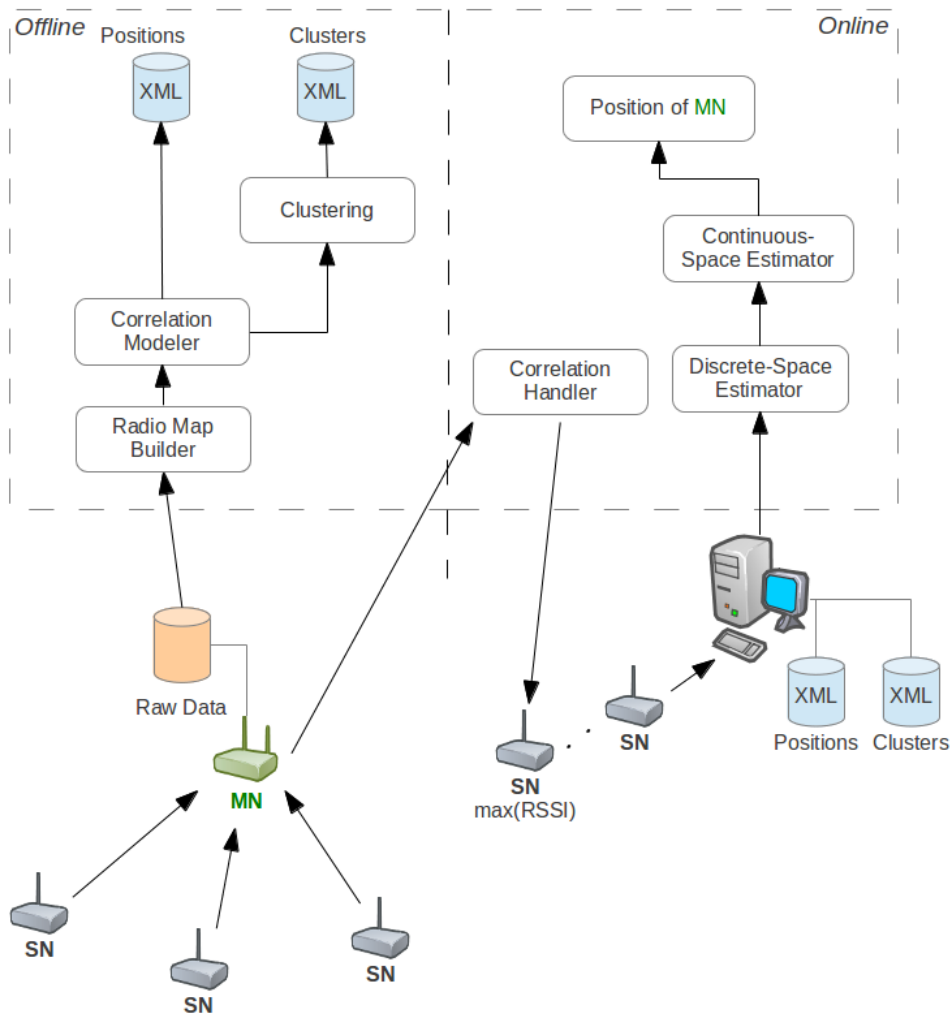


Figura 5.16: Componentes do sistema HORUS modificado e fluxo de informação.

da média e do desvio padrão, das potências recebidas no *Correlation Modeler*. Em cada posição é mantido um número máximo configurável de nós registados, sendo todos os outros removidos, pretendendo-se com isto garantir que na fase *Online* as amostras recebidas são cruzadas com amostras do mesmo tamanho para cada posição.

Por fim é feita a criação de *clusters* no módulo *Clustering*. O processo decorre ordenando por potência recebida as amostras para cada posição e depois escolhendo as n primeiras como chave da posição. Para cada chave diferente é criado um novo *cluster*. De notar que a chave (1000,1001) é igual à chave (1001,1000) não havendo o conceito de ordenação na chave. Neste trabalho considerou-se $n = 2$ por ser um valor apontado no trabalho [21] como tendo bons resultados num sistema real.

Para efeitos de simulação este comportamento foi emulado usando o módulo de mobilidade *TractorMobility* disponibilizado pelo MiXiM e descrito na Secção 5.4. Foram registadas posições separadas na horizontal por 1 metro e na vertical por 2 metros para toda a área de trabalho,

em períodos de 40 segundos. Após cada mudança de posição, o nó móvel simulado calcula a média e o desvio padrão dos valores de RSSI recebidos e guarda numa estrutura de dados de posições *radioMap* (ver Secção 5.3.2). Quando a simulação termina o *radioMap* é transformado num ficheiro XML e são criados os *clusters* da forma descrita no parágrafo anterior, que também são transformados num ficheiro XML.

Fase Online

Nesta fase o nó móvel recolhe e guarda assinaturas dos nós fixos durante um determinado tempo. Após esse momento usa o *Correlation Handler* para efectuar uma média por nó fixo, das diversas potências recebidas. Esse resultado é enviado para o nó com o maior valor de RSSI detectado até esse momento. O nó fixo reconhece o tipo de mensagem e envia directamente ou através de outros nós fixos, a mensagem para o nó base.

O nó base contém os ficheiros XML gerados na fase *offline* e como tal possui toda a informação necessária para determinar a probabilidade do nó estar numa determinada posição. Inicialmente a estimativa é discreta (*Discrete-Space Estimator*) e apenas são determinadas posições para o nó móvel que constam no mapa rádio.

Nesse momento o nó fixo analisa o sinal recebido, verifica se estão presentes um número mínimo de nós fixos e determina a chave da amostra (neste caso os dois nós com maior RSSI). Com essa chave obtém o *cluster* correspondente e as posições associadas. Para cada posição é calculada a densidade de probabilidade da distribuição normal (Ver Secção 5.3.2). A posição com maior valor é a posição obtida à saída do *Discrete-Space Estimator*.

Uma vez que existem diversas posições onde o nó móvel poderá estar, que não estão registadas no mapa rádio é necessária uma estimação no espaço contínua. É essa precisamente a função do *Continuous-Space Estimator* que irá através de duas técnicas: centro de massa das posições e média temporal do espaço físico, determinar uma posição estimada da posição real do nó.

5.3.2 Modelo Matemático

O sistema *HORUS* é um método probabilístico. Neste trabalho optou-se pela parametrização do sinal recebido, que permite obter uma distribuição normal que se ajusta ao histograma das potências recebidas tal como podemos ver na Figura 5.17.

A função densidade de probabilidade da distribuição normal é dada por:

$$f dp(q) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(q-\mu)^2}{2\sigma^2}} \quad (5.1)$$

Para n amostras recebidas por um nó fixo i os parâmetros de (5.1) são obtidos, calculando a média e o desvio padrão das amostras obtidas, usando as equações 5.2 e 5.3.

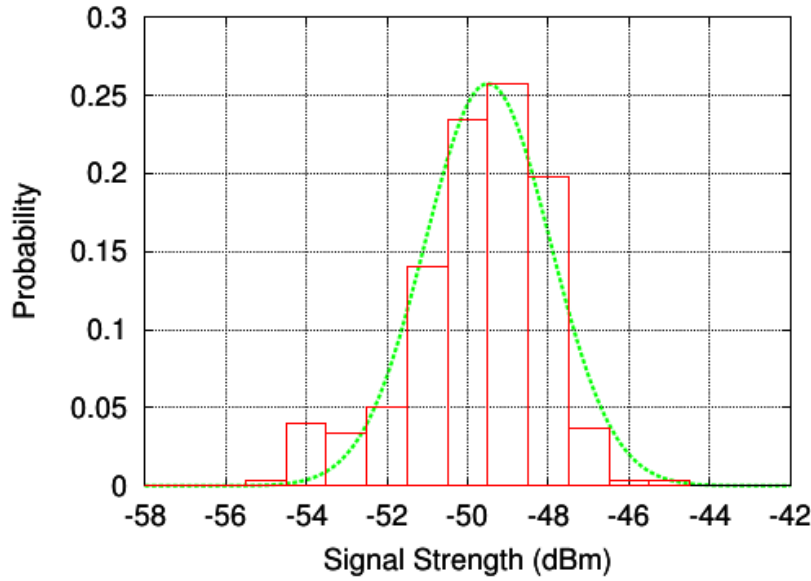


Figura 5.17: Exemplo de parametrização da distribuição de potência do sinal recebido [21].

$$\mu = \frac{1}{n} \sum_{j=1}^n s_i(j) \quad (5.2)$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{j=1}^n (s_i(j) - \mu)^2} \quad (5.3)$$

onde $s_i(j)$ é a amostra j proveniente do nó fixo i .

O facto de se aproximar as amostras a uma função densidade de probabilidade permite uma poupança significativa de espaço de armazenamento, uma vez que deixa de ser necessário guardar os valores das potências recebidas. Para cada par (nó, posição) passa por isso a ser necessário guardar apenas a média e desvio padrão da função. Esta abordagem permite também obter valores de probabilidade para potências de sinal que não tenham sido medidas e filtrar possíveis anomalias registadas durante a fase *offline*.

Na fase *online* é necessário calcular a probabilidade de um nó estar numa determinada posição. Assim é necessário para um vector $s = (s_1, \dots, s_k)$ encontrar a posição x que maximiza a probabilidade $P(x/s)$. Para obter esta probabilidade usando a função densidade de probabilidade entretanto guardada recorreremos à equação:

$$P(s_i/x) = P(s_i \leq 0.5) = \int_{s_i-0.5}^{s_i+0.5} f dp(q) dq \quad (5.4)$$

Dado que queremos evitar a utilização de algoritmos numéricos, computacionalmente pesados, para a resolução do integral, transformou-se a função densidade de probabilidade encontrada para cada nó estático, numa função de distribuição normal standard, cujos valores podem

ser tabelados. Para efeitos de simulação criou-se por isso, um ficheiro XML que contém os valores necessários.

Assim sabendo que:

$$P(X \leq x) = P\left(\frac{X - \mu}{\sigma} \leq \frac{x - \mu}{\sigma}\right) = P(Z \leq z) \quad (5.5)$$

onde Z é uma variável normal aleatória standard, podemos calcular a probabilidade da (5.4) como:

$$P(s_i \leq 0.5) = P\left(Z \leq \frac{s_i + 0.5 - \mu_i}{\sigma_i}\right) \quad (5.6)$$

onde com os valores tabelados rapidamente chegamos a um resultado.

Para um amostra que contenha vários nós estáticos, a probabilidade conjunta é dada pela multiplicação das probabilidades individuais calculadas para cada nó:

$$P = \prod_{j=1}^n P_i \quad (5.7)$$

Obtida uma posição usando o *Discrete-Space Estimator* torna-se necessário ainda usar o *Continous-Space Estimator* para chegar a um valor mais próximo da localização real.

A primeira técnica é o centro de massa de N posições obtidas. A posição estimada no espaço contínuo (x, y) é obtida a partir das seguintes equações, considerando o espaço bi-dimensional utilizado neste trabalho:

$$x = \frac{\sum_{j=1}^{\min(N,P)} x_i P_i}{\sum P_i} \quad (5.8)$$

$$y = \frac{\sum_{j=1}^{\min(N,P)} y_i P_i}{\sum P_i} \quad (5.9)$$

Onde P é o número de posições encontradas para um determinado nó móvel.

A segunda técnica consiste em fazer uma média temporal das K posições encontradas numa localização anterior para o mesmo nó, dada pelas equações:

$$x = \frac{\sum_{j=1}^K x_i}{K} \quad (5.10)$$

$$y = \frac{\sum_{j=1}^K y_i}{K} \quad (5.11)$$

Neste trabalho serão usados os valores de base $N = 6$ e $K = 10$ referenciados em [21] como valores com bons resultados numa situação real. Serão no entanto, obtidos resultados na simulação, de modo a aferir quais os valores que dão origem aos melhores resultados.

5.3.3 Estruturas de Dados e Ficheiros XML

São definidas duas estruturas de dados no *HORUS* modificado.

Na Figura 5.18 temos a estrutura de dados escolhida para guardar as posições obtidas à saída do *Correlation Modeler*. O resultado é um conjunto de posições, onde para cada posição, existem várias funções de densidade de probabilidade para cada nó, *Probability Density Functions* (PDFs) caracterizadas pelos seus valores de média e desvio padrão.

Na Figura 5.19 está representada a estrutura de dados utilizada para guardar os *clusters*.

Estas estruturas estão presentes no nó móvel e no nó fixo e servem para guardar em memória o conteúdo que será escrito ou lido nos ficheiros XML. Nesta simulação são, por isso, usados dois tipos de ficheiro XML para guardar o mapa rádio de posições e os *clusters*. Estes ficheiros são gerados quando termina a simulação do modo *offline* e carregados pelo nó base quando começa a simulação do modo *online*. No Anexo A.1 está um exemplo de um ficheiro XML de posições rádio e no Anexo A.3 pode ser consultado um exemplo de um ficheiro XML de clusters.

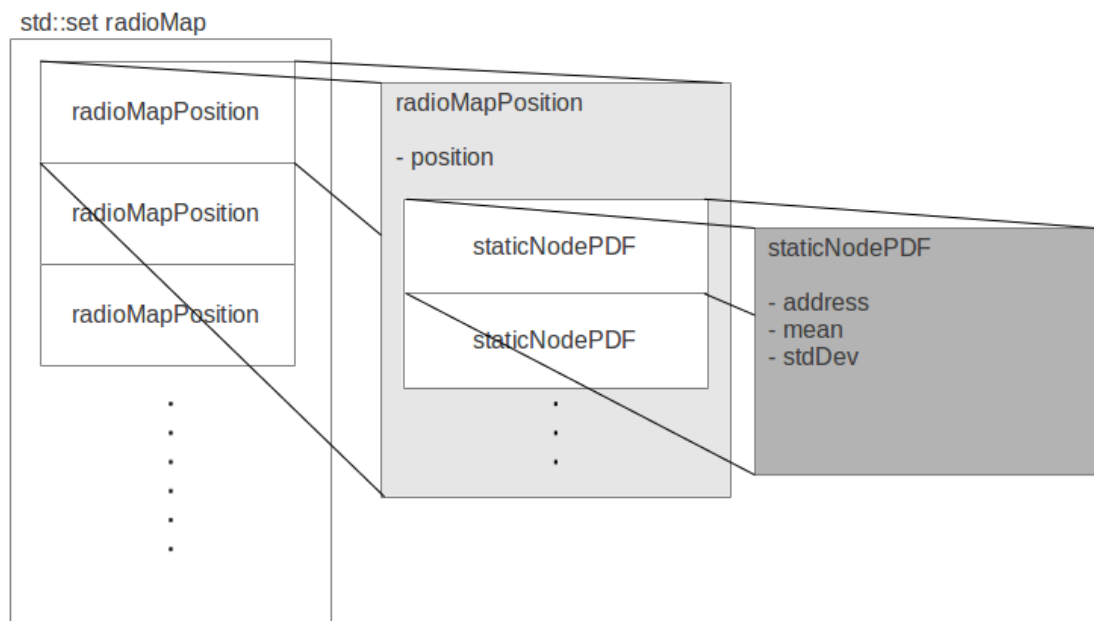


Figura 5.18: Estrutura de dados para armazenamento das posições do mapa rádio obtidas na fase *offline*.

5.3.4 Modo de Funcionamento

Na camada de aplicação dos nós pertencentes ao EMoS é definido o papel do nó da rede. Cada nó tem a sua camada própria, o que em termos reais corresponderia a um programa em *nesC* a ser executado sobre o *TinyOS*.

O nó móvel e o nó base, têm ambos um papel fulcral na localização. Apresentam-se por isso nas Figuras 5.20 e 5.21 os fluxogramas correspondentes ao funcionamento da camada de

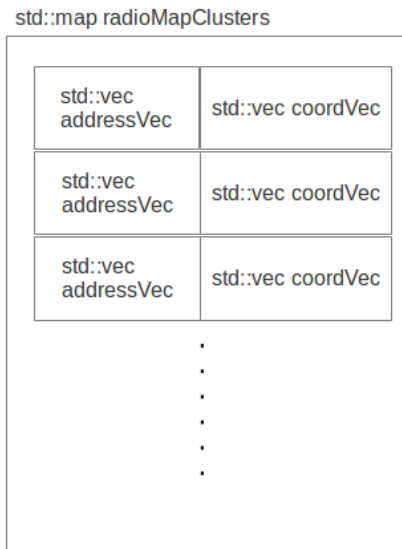


Figura 5.19: Estrutura de dados para armazenamento dos *clusters* obtidos na fase *offline*.

aplicação desses nós no contexto da localização.

No fluxograma da Figura 5.20 observamos o comportamento do nó móvel. Este nó está permanentemente a receber e guardar mensagens de assinatura vindas dos nós fixos espalhados pela casa.

No ramo esquerdo do fluxograma podemos observar o que acontece quando uma assinatura de um nó fixo é recebida. Se o nó móvel estiver em modo de calibração, utilizado durante a fase *offline*, o nó verifica se houve mudança de posição e se existem nós estáticos suficientes para considerar a posição válida. Caso tal aconteça são calculados os parâmetros da função densidade de probabilidade normal correspondente às potências amostradas (calculando-se a média e o desvio padrão) sendo o resultado guardado no mapa rádio. Pelo contrário se estivermos na fase *online* são apenas registados os valores do endereço e potência de sinal recebidos.

Quando não está no modo de calibração, o nó móvel usa um conceito de agendamento do OMNeT++ que permite agendar eventos, através de mensagens que são enviadas, num determinado tempo da simulação, para o próprio nó. Assim este conceito é usado para enviar periodicamente mensagens para o nó base para que este possa calcular a posição do nó móvel. Este comportamento está exemplificado na ramo direito do fluxograma.

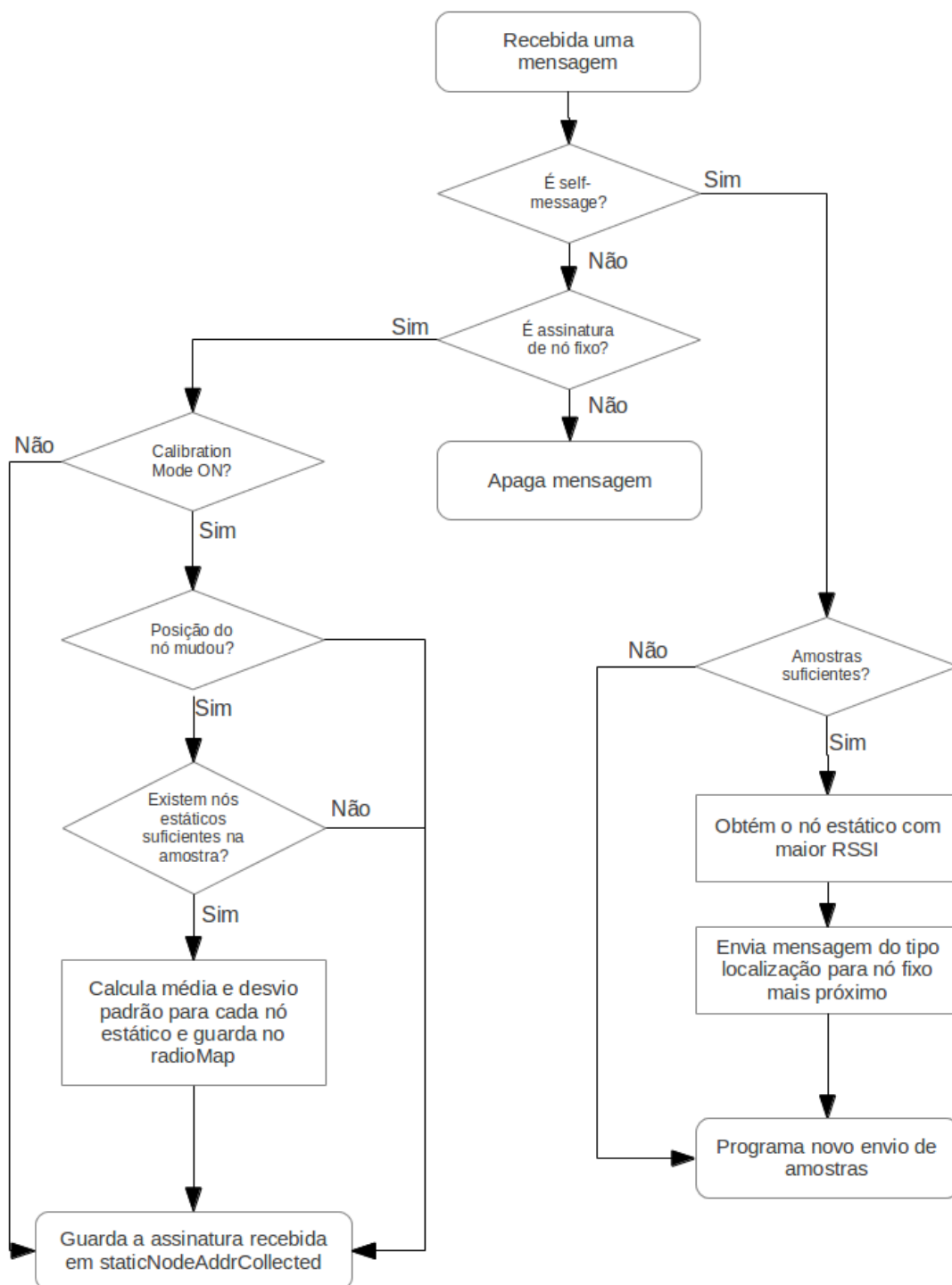


Figura 5.20: Fluxograma do modo de funcionamento do nó móvel na localização.

Por sua vez na Figura 5.21 temos um fluxograma do modo de funcionamento do nó base. Este nó recebe de um nó móvel, por intermédio de um ou mais nós fixos, uma mensagem que contém as médias das potências de nós fixos, recebidas pelo nó móvel durante um determinado espaço de tempo. Assim quando o nó base recebe esta mensagem, ordena por potência de forma descendente e selecciona apenas os N primeiros, em que N representa o número de nós estáticos presentes em cada posição do mapa rádio.

São então seleccionados os endereços dos dois primeiros nós estáticos da amostra que constituem a chave da amostra. Com a chave são obtidas as posições do *cluster* cuja chave é igual à chave da amostra. Percorrendo então todas as posições obtidas é calculada para cada uma, a probabilidade do nó móvel estar nessa posição.

Se a probabilidade for maior que zero então é guardada numa lista. Uma vez percorridos todas as posições se ainda não houver na lista pelo menos uma posição, decrementa-se o valor de N e torna-se a obter as posições do *cluster* para novo cálculo. Se N chegar a zero então o processo acaba sem conseguir determinar uma posição para o nó móvel. Se houver uma posição então é efectuada a estimação de espaço contínuo através do centro de massa das posições e é calculada a média temporal de espaço contínuo com as últimas posições calculadas em amostras anteriores.

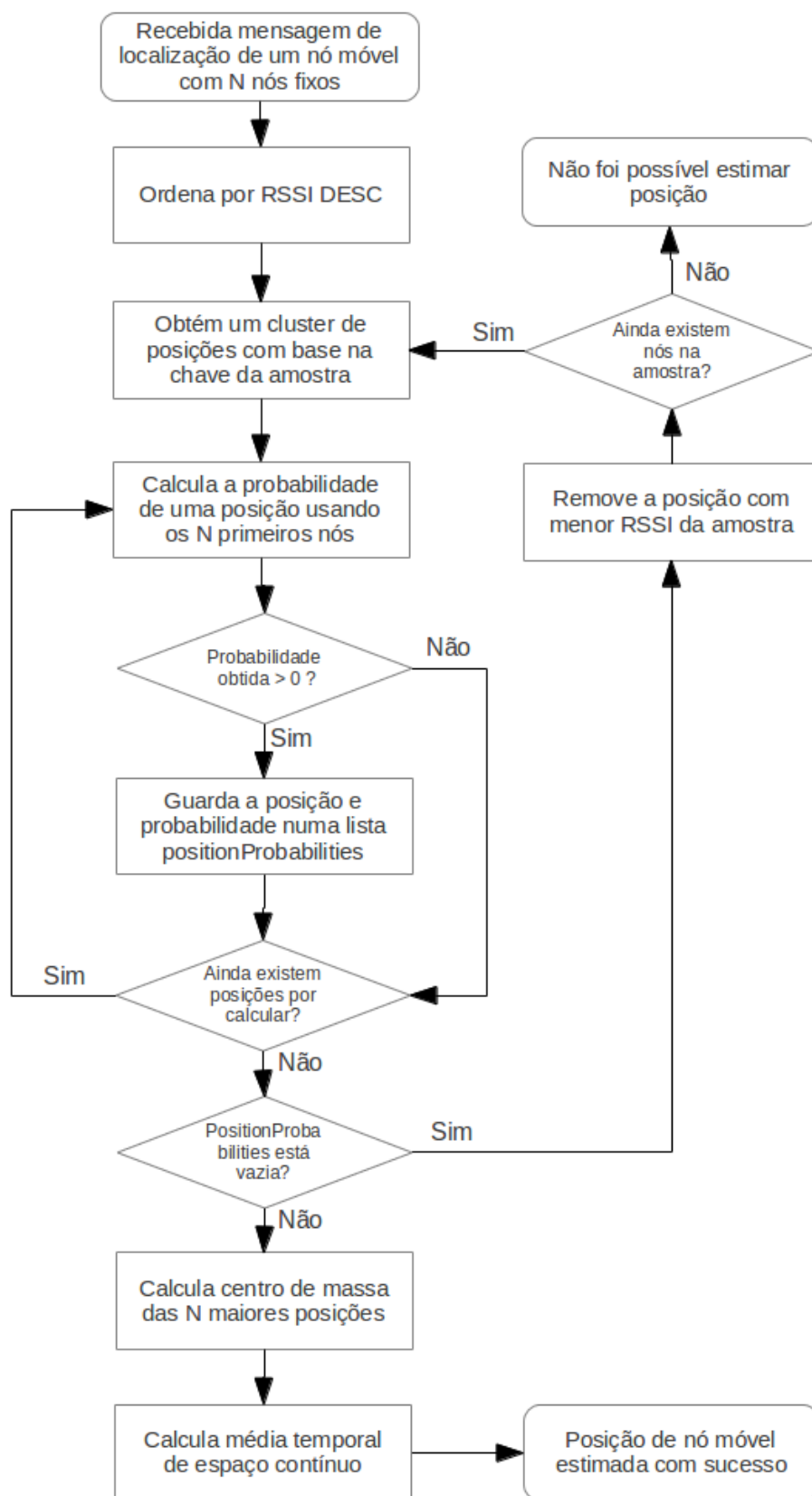


Figura 5.21: Fluxograma do modo de funcionamento do nó base na localização.

5.4 Mobilidade

Neste trabalho são usados dois modelos de mobilidade implementados no MiXiM. São eles o *TurtleMobility* e o *TractorMobility*.

O *TurtleMobility* permite criar um percurso com comandos introduzidos na simulação através dum ficheiro XML. Desta forma é simulado o movimento de uma pessoa dentro de uma casa de forma simples, flexível e sem ter de efectuar uma compilação do programa de cada vez que são feitas alterações ao movimento. Esta opção permite ainda que outra aplicação possa gerar um ficheiro XML com dados reais substituindo o ficheiro existente. No Anexo A.2 está um exemplo de um ficheiro XML de configuração do *TurtleMobility*.

Por sua vez o *TractorMobility* usado especificamente para a criação do mapa de rádio da simulação, na fase *offline*, permite criar de forma simples um percurso para a obtenção de pontos igualmente espaçados, criando uma malha sobre todo o ambiente de simulação. Na Figura 5.22 é possível observar um exemplo do percurso disponibilizado neste modelo. De notar que numa situação real seria impossível construir um mapa rádio igual devido à existência de paredes a meio do percurso, mas para um cenário de simulação onde o objectivo é obter um mapa de rádio minimamente de forma dinâmica e rápida, a solução encontrada é perfeitamente aceitável.

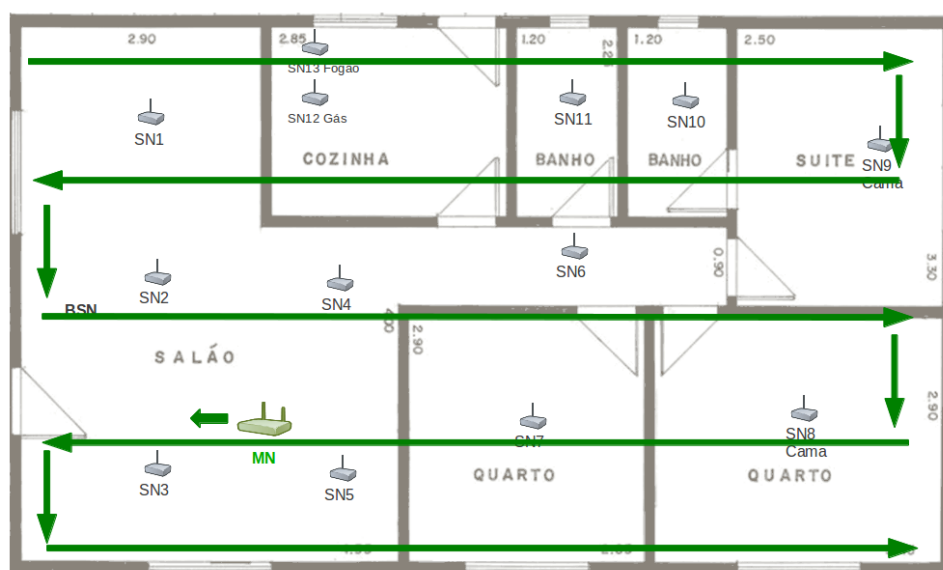


Figura 5.22: Movimento do nó móvel durante a fase *offline* usando o modelo *TractorMobility* disponível no MiXiM.

6

Resultados

Conteúdo

6.1	<i>Testbed</i>	64
6.2	Sistema de Localização	64

6.1 Testbed

Por forma a validar os resultados e a simulação desenvolvida, foi necessário criar um cenário de monitorização em ambiente doméstico. No cenário criado foram colocados 9 nós fixos, 1 nó móvel e 1 nó base, com o intuito de monitorizar uma pessoa idosa na sua rotina normal.

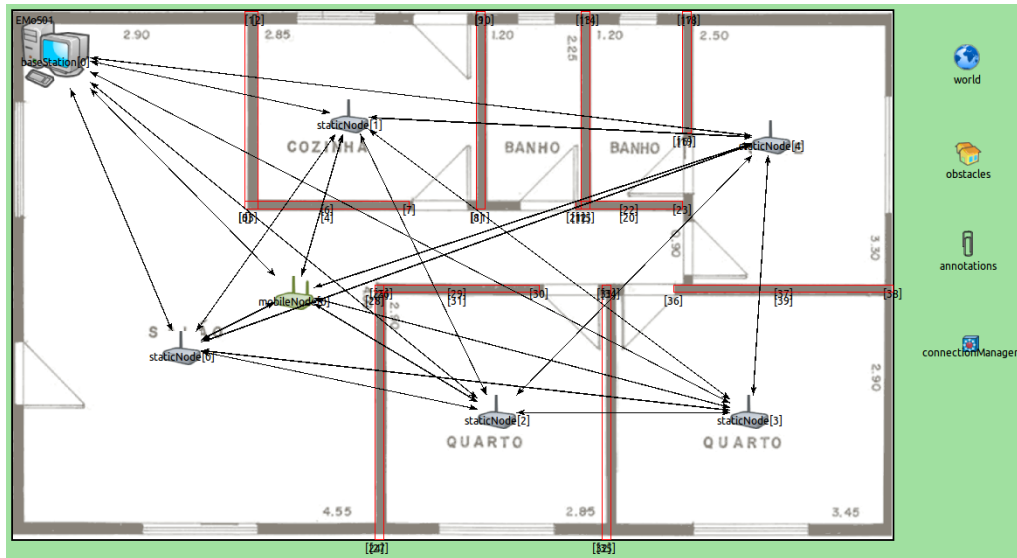


Figura 6.1: Cenário da simulação criada para testar o EMoS.

Na Figura 6.1 é possível observar os 9 nós, que estão todos interligados. A ligação entre nós não se deve ao facto de estarem todos à distância de receção, embora neste cenário seja o caso das distâncias consideradas, mas porque no MiXiM todos os nós que estejam à distância de interferência consideram-se ligados, mesmo que seja impossível a receção.

O cenário escolhido consiste numa casa com cerca de 93.75 m² de área (12.5 m x 7.5 m) onde foram colocados obstáculos para cada parede interior, visíveis a vermelho na imagem. O conteúdo do ficheiro de configuração de obstáculos está no Anexo B.2.

6.2 Sistema de Localização

Para testar o protocolo de localização foi feito um mapa rádio da *testbed* usando o modelo de mobilidade *TractorMobility* tal como indicado no capítulo anterior.

Foi também necessário criar um caminho para o nó móvel para simular uma pessoa em movimento dentro da casa. Criou-se então o cenário de mobilidade da Figura 6.2 onde existe um percurso pela casa, em que o nó móvel executa o percurso indicado pelas setas a vermelho, parando durante algum tempo em alguns dos pontos por onde passou.

O ficheiro criado para o efeito, está disponível no Anexo B.3 bem como o ficheiro de configuração da simulação *omnetpp.ini*, no Anexo B.4.

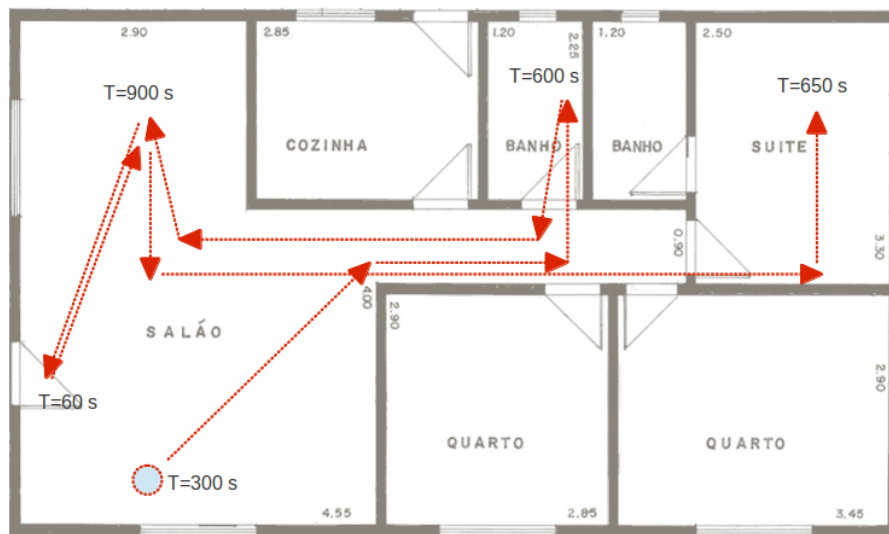


Figura 6.2: Cenário de mobilidade para teste do sistema de localização.

Assim para o mesmo percurso foram feitos diversos testes onde se avaliou o efeito das técnicas de estimação de espaço contínuo.

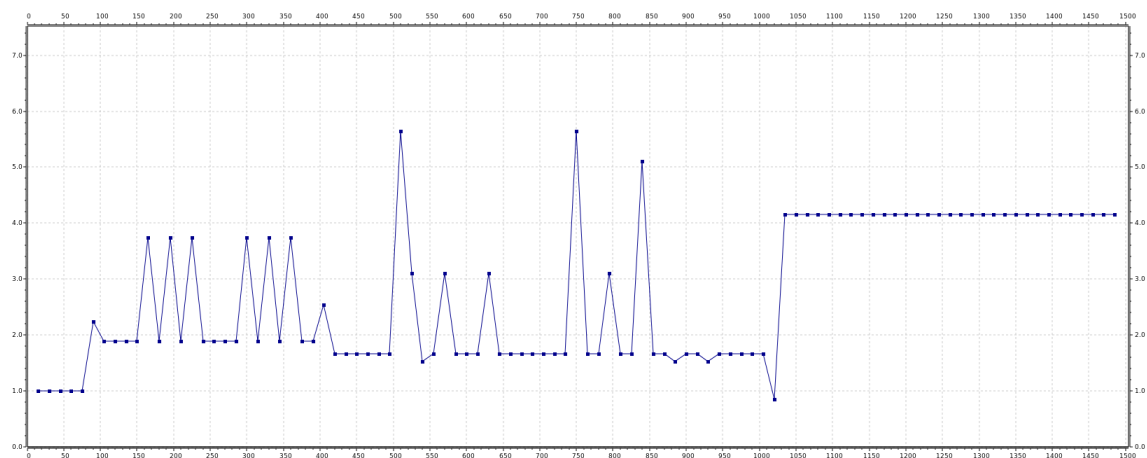


Figura 6.3: Erros de estimativa no sistema HORUS sem técnicas de centro de massa e sem média.

Podemos observar na Figura 6.3 a estima de localização ao longo do percurso, obtida a partir do estimador de espaço discreto. Quanto mais próximo o nó móvel está dos pontos que foram anteriormente medidos na fase *offline*, melhores são as estimativas da posição. Na Figura 6.4, está o resultado da estimação ao longo do percurso efectuado usando a técnica do centro de massa. Observa-se uma melhoria relativamente ao anterior.

Por fim na Figura 6.5 vemos o melhor valor de precisão alcançado pelas três simulações, no mínimo do gráfico em questão, mas por outro lado vemos um sistema bastante mais afectado por mudanças súbitas da posição estimada.

Outro ponto que pode ser interessante é a taxa com que o nó móvel reúne as assinaturas dos

6. Resultados

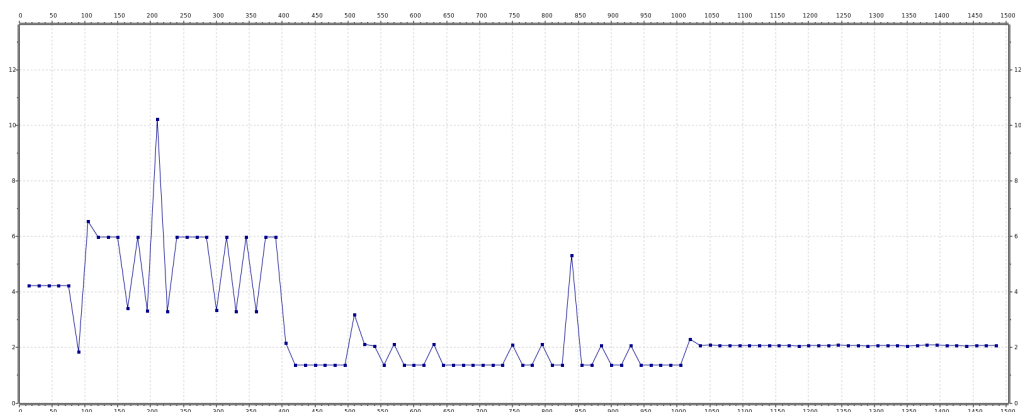


Figura 6.4: Erros de estimativa no sistema HORUS com centro de massa e sem média.

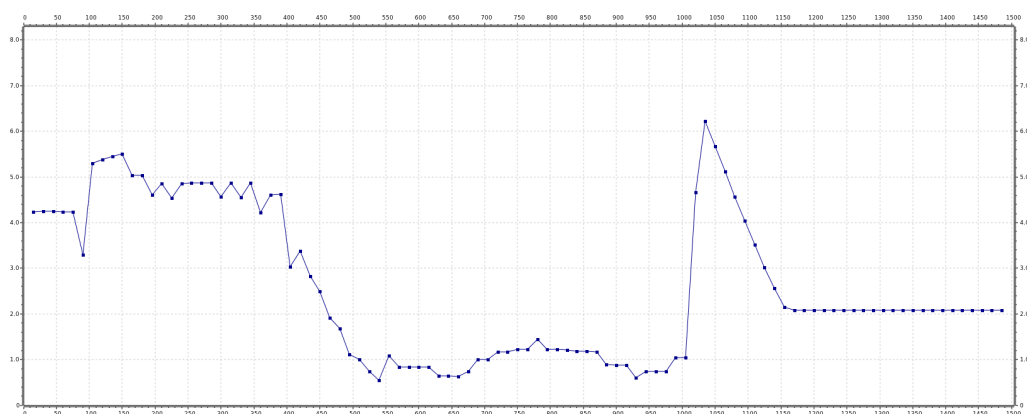


Figura 6.5: Erros de estimativa no sistema HORUS com centro de massa e com média.

nós fixos e envia tudo para o nó base. Os testes anteriores foram feitos com 15 segundos entre envios. Se for alterado esse valor para 40 s obtemos uma curva bastante mais estável, uma vez que a informação reunida é muito mais semelhante à que está no mapa rádio, dada a correlação existente entre amostras.

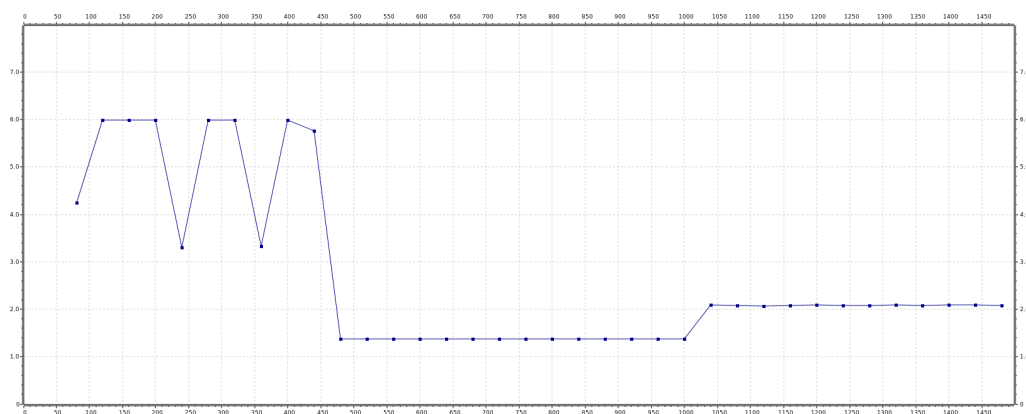


Figura 6.6: Erros de estimativa no sistema HORUS com centro de massa e sem média para um tempo de amostragem do nó móvel de $t=40s$.

7

Conclusões e Trabalho Futuro

7. Conclusões e Trabalho Futuro

Com o intuito de obter uma simulação o mais fidedigna possível, foi necessário ao longo deste trabalho, encontrar soluções que não se limitassem a simular um ou outro aspecto do problema, mas sim o conjunto completo, de funcionalidades do sistema proposto.

Grande parte do problema ficou resolvido com o MiXiM, no entanto foi necessário encontrar um protocolo de encaminhamento adequado e um sistema de localização que conjuntamente, permitissem criar um sistema completamente funcional de monitorização.

Este trabalho permitiu assim desenvolver uma solução simulada de um sistema de monitorização de pessoas idosas, sem deixar de referenciar o hardware necessário e adquirir um conhecimento alargado do protocolo de encaminhamento AODV e do sistema de localização HORUS.

Futuramente poderá ser melhorado o protocolo de localização para que deixe de ser necessário existir uma fase *offline*, utilizando-se nós fixos que saibam a sua própria localização e que com isso criem de forma automática e periódica novos mapas rádio sem a necessidade de intervenção.

Outra possibilidade interessante, seria implementar no nó móvel um conjunto de camadas paralelo com rádio bluetooth, ligando assim esse nó também a uma rede BSN simulada, o que permitiria recriar no ambiente de trabalho eventos humanos que despoletassem mensagens, como aumento da temperatura ou batimento cardíaco.

O melhoramento do modelo de obstáculos ou até mesmo a utilização de nós reais que permitissem chegar aos parâmetros do modelo utilizado, seria também uma possibilidade interessante de trabalho futuro.

Bibliografia

- [1] D. of Economic and S. A. P. Division, World Population Aging 1950-2050, United Nations Std., 2001. [Online]. Available: <http://www.un.org/esa/population/publications/worldageing19502050/>
- [2] A. M. Tabar, A. Keshavarz, and H. Aghajan, "Smart home care network using sensor fusion and distributed vision-based reasoning," in In Proc. of VSSN 2006. ACM Press, 2006, pp. 145–154.
- [3] H. K. Aghajan, J. C. Augusto, C. Wu, P. J. McCullagh, and J.-A. Walkden, "Distributed vision-based accident management for assisted living," in ICOST, ser. Lecture Notes in Computer Science, T. Okadome, T. Yamazaki, and M. Makhtari, Eds., vol. 4541. Springer, 2007, pp. 196–205. [Online]. Available: <http://dblp.uni-trier.de/db/conf/icost/icost2007.html#AghajanAWMW07>
- [4] A. Keshavarz, A. M. Tabar, and Ham, "Distributed vision-based reasoning for smart home care," in ACM SenSys Workshop on Distributed Smart Cameras DSC 06, 2006.
- [5] A. G. Hauptmann, J. Gao, R. Yan, Y. Qi, J. Yang, and H. D. Wactlar, "Automated analysis of nursing home observations," in Pervasive Computing, IEEE, vol. 3. Carnegie Mellon Univ., Pittsburgh, PA, USA, April-June 2004, pp. 15–21.
- [6] A. Williams, D. Xie, S. Ou, R. Grupen, A. Hanson, and E. Riseman, "Distributed smart cameras for aging in place," in In ACM SenSys Workshop on Distributed Smart Cameras, 2006.
- [7] B. U. Töreyn, Y. Dedeoglu, and A. E. Çetin, "Hmm based falling person detection using both audio and video," in ICCV-HCI, ser. Lecture Notes in Computer Science, N. Sebe, M. S. Lew, and T. S. Huang, Eds., vol. 3766. Springer, 2005, pp. 211–220. [Online]. Available: <http://dblp.uni-trier.de/db/conf/iccv/iccv-hci2005.html#ToreyinDC05>
- [8] C. Otto, A. Milenković, C. Sanders, and E. Jovanov, "System architecture of a wireless body area sensor network for ubiquitous health monitoring," J. Mob. Multimed., vol. 1, no. 4, pp. 307–326, Jan. 2005. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2010498.2010502>

- [9] S. Ullah, P. Khan, N. Ullah, S. Saleem, H. Higgins, and K. S. Kwak, "A review of wireless body area networks for medical applications." IJCNS, vol. 2, no. 8, pp. 797–803, 2009. [Online]. Available: <http://dblp.uni-trier.de/db/journals/ijcns/ijcns2.html#UllahKUSHK09>
- [10] B. H. Jung, J. W. Chong, S. H. Jeong, H. Y. Hwang, S. M. Kim, M. S. Kang, and D. K. Sung, "Ubiquitous wearable computer (uwc)-aided coexistence algorithm in an overlaid network environment of wlan and zigbee networks," in Proceedings of the 9th international conference on Communications and information technologies, 2009.
- [11] J. Y. Jung and J. W. Lee, "Zigbee device design and implementation for context-aware u-healthcare system," in Proceedings of the Second International Conference on Systems and Networks Communications, ser. ICSNC '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 22–. [Online]. Available: <http://dx.doi.org/10.1109/ICSNC.2007.88>
- [12] V. M. Stanford, "Pervasive computing: Applications - using pervasive computing to deliver elder care." IEEE Distributed Systems Online, vol. 3, no. 3, pp. 10–13, 2002. [Online]. Available: <http://dblp.uni-trier.de/db/journals/dsonline/dsonline3.html#Stanford02>
- [13] Y. Kaddoura, J. King, and A. S. Helal, "Cost-precision tradeoffs in unencumbered floor-based indoor location tracking," Proceedings of the third International Conference On Smart homes and health Telematic (ICOST), Sherbrooke, Qu?bec, 2005.
- [14] S. S. Intille, K. Larson, J. S. Beaudin, J. Nawyn, E. M. Tapia, and P. Kaushik, "A living laboratory for the design and evaluation of ubiquitous computing technologies," in CHI '05: CHI '05 extended abstracts on Human factors in computing systems. New York, NY, USA: ACM, 2005, pp. 1941–1944.
- [15] M. Beigl, H.-W. Gellersen, and A. Schmidt, "Mediacups: experience with design and use of computer-augmented everyday artefacts." Computer Networks, vol. 35, no. 4, pp. 401–409, 2001. [Online]. Available: <http://dblp.uni-trier.de/db/journals/cn/cn35.html#BeiglGS01>
- [16] D. H. Wilson, "Assistive intelligent environments for automatic in-home health monitoring," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, September 2005.
- [17] M. Wallace and M. Shelkey, "Monitoring functional status in hospitalized older adults." Am J Nurs, vol. 108, pp. 64–71; quiz 71–2, 2008.
- [18] I. Amundson and X. Koutsoukos, "A survey on localization for mobile wireless sensor networks," in Mobile Entity Localization and Tracking in GPS-less Environments, ser. Lecture Notes in Computer Science, R. Fuller and X. Koutsoukos, Eds. Springer Berlin / Heidelberg, 2009, vol. 5801, pp. 235–254. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-04385-7_16

-
- [19] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in INFOCOM, 2000, pp. 775–784.
- [20] K. Lorincz and M. Welsh, "Motetrack: a robust, decentralized approach to rf-based location tracking." Personal and Ubiquitous Computing, vol. 11, no. 6, pp. 489–503, 2007. [Online]. Available: <http://dblp.uni-trier.de/db/journals/puc/puc11.html#LorinczW07>
- [21] M. A. A. A. Y. A. Rehim, "Horus: A wlan-based indoor location determination system," Ph.D. dissertation, University of Maryland, April 2004. [Online]. Available: <http://www.lib.umd.edu/drum/handle/1903/1364>
- [22] M. Ayaz, I. Baig, A. Abdullah, and I. Faye, "A survey on routing techniques in underwater wireless sensor networks." J. Network and Computer Applications, vol. 34, no. 6, pp. 1908–1927, 2011. [Online]. Available: <http://dblp.uni-trier.de/db/journals/jnca/jnca34.html#AyazBAF11>
- [23] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks." in MOBICOM, H. Kodesh, V. Bahl, T. Imielinski, and M. Steenstrup, Eds. ACM, 1999, pp. 174–185. [Online]. Available: <http://dblp.uni-trier.de/db/conf/mobicom/mobicom1999.html#HeinzelmanKB99>
- [24] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks." in MOBICOM, R. L. Pickholtz, S. K. Das, R. Cáceres, and J. J. Garcia-Luna-Aceves, Eds. ACM, 2000, pp. 56–67. [Online]. Available: <http://dblp.uni-trier.de/db/conf/mobicom/mobicom2000.html#IntanagonwiwatGE00>
- [25] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (aodv) routing," in EXPERIMENTAL RFC 3561, July 2003.
- [26] D. B. Johnson, D. A. Maltz, and J. Broch, "Ad hoc networking," in In Ad Hoc Networking, edited by Charles E. Perkins, Chapter 5. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001, ch. DSR: the dynamic source routing protocol for multihop wireless ad hoc networks, pp. 139–172. [Online]. Available: <http://dl.acm.org/citation.cfm?id=374547.374552>
- [27] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks." in HICSS, 2000. [Online]. Available: <http://dblp.uni-trier.de/db/conf/hicss/hicss2000-8.html#HeinzelmanCB00>
- [28] PEGASIS: Power-efficient gathering in sensor information systems.
- [29] Y. Yu, R. Govindan, and D. Estrin, "Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks," UCLA - University of California, Los Angeles, Tech. Rep., 2001.
-

- [30] J. L. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister, "System architecture directions for networked sensors." in ASPLOS, L. Rudolph and A. Gupta, Eds. ACM Press, 2000, pp. 93–104. [Online]. Available: <http://dblp.uni-trier.de/db/conf/asplos/asplos2000.html#HillSWHCP00>
- [31] A. Varga, "The omnet++ discrete event simulation system," Proceedings of the European Simulation Multiconference (ESM'2001), June 2001.
- [32] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. K. Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin, "Simulating wireless and mobile networks in omnet++ the mixim vision." in SimuTools, S. Molnár, J. R. Heath, O. Dalle, and G. A. Wainer, Eds. ICST, 2008, p. 71. [Online]. Available: <http://dblp.uni-trier.de/db/conf/simutools/simutools2008.html#KopkeSWWHPVLV08>
- [33] C. Sommer, D. Eckhoff, R. German, and F. Dressler, "A computationally inexpensive empirical model of ieee 802.11p radio shadowing in urban environments."
- [34] K. Wessel, M. Swigulski, A. Köpke, and D. Willkomm, "Mixim: the physical layer an architecture overview," in Proceedings of the 2nd International Conference on Simulation Tools and Techniques, ser. Simutools '09. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, pp. 78:1–78:8. [Online]. Available: <http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5555>



Apêndice 1 - Ficheiros XML de Exemplo

A. Apêndice 1 - Ficheiros XML de Exemplo

Listagem A.1: Exemplo de ficheiro XML de configuração do mapa rádio.

```
1 <?xml version="1.0" encoding="UTF-8"?><radioMap maxPositionPDFsSize="4">
2   <position x="2" y="2">
3     <staticNodePDF address="1000" mean="-6.86524" stdDev="-2.39173"/>
4     <staticNodePDF address="1005" mean="-19.6319" stdDev="-15.9808"/>
5     <staticNodePDF address="1001" mean="-21.0037" stdDev="-19.5426"/>
6     <staticNodePDF address="1006" mean="-22.8628" stdDev="-19.3251"/>
7   </position>
8   <position x="4" y="2">
9     <staticNodePDF address="1000" mean="-8.58641" stdDev="-7.49053"/>
10    <staticNodePDF address="1006" mean="-18.8375" stdDev="-16.4028"/>
11    <staticNodePDF address="1005" mean="-19.2409" stdDev="-17.7517"/>
12    <staticNodePDF address="1001" mean="-19.2659" stdDev="-19.1632"/>
13  </position>
14  <position x="3" y="2">
15    <staticNodePDF address="1000" mean="31.6669" stdDev="33.1443"/>
16    <staticNodePDF address="1005" mean="-22.0094" stdDev="-19.9634"/>
17    <staticNodePDF address="1006" mean="-22.8608" stdDev="-21.7965"/>
18    <staticNodePDF address="1001" mean="-23.1735" stdDev="-21.8065"/>
19  </position>
20  <position x="6" y="2">
21    <staticNodePDF address="1001" mean="-10.6905" stdDev="-6.60051"/>
22    <staticNodePDF address="1000" mean="-15.8409" stdDev="-15.7502"/>
23    <staticNodePDF address="1006" mean="-18.4932" stdDev="-17.3782"/>
24    <staticNodePDF address="1005" mean="-19.1748" stdDev="-18.4118"/>
25  </position>
26  <position x="5" y="2">
27    <staticNodePDF address="1000" mean="-13.661" stdDev="-11.7475"/>
28    <staticNodePDF address="1001" mean="-15.1687" stdDev="-13.708"/>
29    <staticNodePDF address="1006" mean="-23.314" stdDev="-20.5445"/>
30    <staticNodePDF address="1005" mean="-23.8983" stdDev="-22.9379"/>
31  </position>
32  <position x="7" y="2">
33    <staticNodePDF address="1001" mean="-9.56809" stdDev="-8.91176"/>
34    <staticNodePDF address="1000" mean="-16.7335" stdDev="-15.6741"/>
35    <staticNodePDF address="1006" mean="-17.5364" stdDev="-15.7507"/>
36    <staticNodePDF address="1002" mean="-17.5544" stdDev="-15.1108"/>
37  </position>
38 </radioMap>
```

Listagem A.2: Exemplo de ficheiro XML de configuração do módulo *TurtleMobility* do MiXiM.

```
1 <movement>
2   <set speed="10" angle="180"/>
3   <repeat n="4">
4     <forward d="50"/>
5     <turn angle="90"/>
6   </repeat>
7   <repeat>
8     <set speed="uniform(10,20)"/>
9     <turn angle="uniform(-30,30)"/>
10    <forward t="uniform(0.1,1)"/>
11  </repeat>
12 </movement>
```

Listagem A.3: Exemplo de ficheiro XML de configuração de *clusters* de posições.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <radioMapClusters clusterKeySize="1">
3   <cluster>
4     <clusterKey>
5       <staticNode address="1001"/>
6     </clusterKey>
7     <position x="6" y="2"/>
8     <position x="7" y="2"/>
9     <position x="8" y="2"/>
10    <position x="9" y="2"/>
11    <position x="10" y="2"/>
12    <position x="11" y="4"/>
13    <position x="10" y="4"/>
14    <position x="9" y="4"/>
15    <position x="8" y="4"/>
16    <position x="7" y="4"/>
17    <position x="6" y="4"/>
18    <position x="10" y="6"/>
19  </cluster>
20  <cluster>
21    <clusterKey>
22      <staticNode address="1000"/>
23    </clusterKey>
24    <position x="2" y="2"/>
25    <position x="3" y="2"/>
26    <position x="4" y="2"/>
27    <position x="5" y="2"/>
28    <position x="5" y="4"/>
29    <position x="4" y="4"/>
30    <position x="3" y="4"/>
31    <position x="2" y="4"/>
32    <position x="2" y="5"/>
33    <position x="2" y="6"/>
34    <position x="3" y="6"/>
35  </cluster>
36 </radioMapClusters>
```



Apêndice 2 - Ficheiros XML de Configuração

Listagem B.1: Ficheiro XML de configuração da camada NIC.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <root>
3   <!-- ANALOGUE MODELS -->
4   <AnalogueModels>
5     <AnalogueModel type="SimplePathlossModel">
6       <parameter name="alpha" type="double" value="3.0"/>
7       <parameter name="carrierFrequency" type="double" value="2.412e+9"/>
8     </AnalogueModel>
9     <!-- we add a log-normal shadowing effect on top of the IEEE 802.15.4 path loss -->
10    <AnalogueModel type="LogNormalShadowing">
11      <parameter name="mean" type="double" value="0.0"/>
12      <parameter name="stdDev" type="double" value="6"/>
13      <parameter name="interval" type="double" value="1"/>
14    </AnalogueModel>
15    <!-- we add obstacle shadowing on top of the log-normal shadowing effect -->
16    <AnalogueModel type="SimpleObstacleShadowing">
17      <parameter name="carrierFrequency" type="double" value="2.412E+9"/>
18    </AnalogueModel>
19  </AnalogueModels>
20 </root>

```

Listagem B.2: Ficheiro de configuração dos objectos da *testbed*.

```

1 <obstacles>
2   <!--<poly id="wall#0" type="brickWall20cm" color="#F00" shape="x,y x,y x,y x,y" />-->
3   <poly id="wall#0" type="brickWall20cm" color="#F00" shape="3.29,0 3.47,0 3.47,2.81 3.29,2
4     .81" />
5   <poly id="wall#0" type="brickWall20cm" color="#F00" shape="3.29,2.81 3.29,2.69 5.62,2.69
6     5.62,2.81" />
7   <poly id="wall#0" type="brickWall20cm" color="#F00" shape="6.58,0 6.7,0 6.7,2.81 6.58,2.81"
8     />
9   <poly id="wall#0" type="brickWall20cm" color="#F00" shape="8.07,0 8.19,0 8.19,2.81 8.07,2
10     .81" />
11   <poly id="wall#0" type="brickWall20cm" color="#F00" shape="9.51,0 9.63,0 9.63,1.73 9.51,1
12     .73" />
13   <poly id="wall#0" type="brickWall20cm" color="#F00" shape="8,2.81 8,2.69 9.51,2.69 9.51,2
14     .81" />
15   <poly id="wall#0" type="brickWall20cm" color="#F00" shape="5.14,3.89 5.26,3.89 5.26,7.5
16     5.14,7.5" />
17   <poly id="wall#0" type="brickWall20cm" color="#F00" shape="5.14,3.89 7.47,3.89 7.47,3.98
18     5.14,3.98" />
19   <poly id="wall#0" type="brickWall20cm" color="#F00" shape="8.37,3.89 8.49,3.89 8.49,7.5
20     8.37,7.5" />
21   <poly id="wall#0" type="brickWall20cm" color="#F00" shape="9.39,3.89 12.5,3.89 12.5,3.98
22     9.39,3.98" />
23 </obstacles>

```

Listagem B.3: Ficheiro de configuração para o cenário de mobilidade usado nos testes do sistema de localização.

```

1 <movement>
2   <wait t="100"/>
3   <set speed="0.8"/>
4   <moveto x="2.39" y="5.98"/>
5   <wait t="300"/> <!-- ver TV -->
6   <moveto x="5.38" y="3.29"/>
7   <moveto x="7.72" y="3.29"/>
8   <moveto x="7.72" y="1.5"/>
9   <wait t="600"/> <!-- WC -->
10  <moveto x="7.72" y="3.29"/>
11  <moveto x="1.61" y="3.29"/>
12  <moveto x="1.44" y="1.73"/>
13  <wait t="900"/> <!-- jantar -->
14  <moveto x="0.6" y="3.89"/> <!--tocaram a campainha -->
15  <wait t="60"/>
16  <moveto x="1.44" y="1.73"/>
17  <wait t="300"/> <!-- termina jantar -->
18  <moveto x="1" y="3.29"/>
19  <moveto x="11.36" y="3.29"/>
20  <moveto x="11.36" y="0.9"/>
21  <wait t="650"/> <!-- dormir um bocado -->
22 </movement>

```

B. Apêndice 2 - Ficheiros XML de Configuração

Listagem B.4: Ficheiro de configuração omnetpp.ini para testes de localização.

```
1 [General]
2 network = EMoS01
3 repeat = 1
4 ned-path=/home/mnobreaga/workspace/mixim/src;/home/mnobreaga/workspace/mixim
5 tkenv-image-path=/home/mnobreaga/workspace/mixim/images/EMoS
6 load-libs=/home/mnobreaga/workspace/mixim/src/modules/miximmodules
   /home/mnobreaga/workspace/mixim/src/base/miximbase
7
8 **.coreDebug = false
9
10 ## WORLD - ANNOTATIONS ##
11 *.annotations.draw = true
12 **.world.use2D = true
13 **.playgroundSizeX = 12.5 m
14 **.playgroundSizeY = 7.5 m
15 **.playgroundSizeZ = 0 m
16
17 ## CONNECTION ##
18 **.connectionManager.sendDirect = false
19 **.connectionManager.pMax = 1.1mW
20 **.connectionManager.sat = -100dBm
21 **.connectionManager.alpha = 2.0
22 **.connectionManager.carrierFrequency = 2.4E+9Hz
23
24 ## OBSTACLES ##
25 *.obstacles.debug = true
26
27 ## NODES ##
28 ***.app.debug = true
29 ***.net.debug = true
30
31 #NICs config
32 ***.nic.phy.usePropagationDelay = true
33 ***.nic.phy.useThermalNoise = true
34 ***.nic.phy.analogueModels = xmldoc("xml_config_files/config.nic.xml")
35 ***.nic.phy.decider = xmldoc("xml_config_files/config.nic.xml")
36 **.mobileNode[*].nic.phy.maxTXPower = 1 mW #XBEE
37 **.staticNode[*].nic.phy.maxTXPower = 1.1 mW #CC2420
38 **.baseStation[*].nic.phy.maxTXPower = 1 mW #XBEE
39 **.mobileNode[*].nic.phy.sensitivity = -92dBm #XBEE
40 **.staticNode[*].nic.phy.sensitivity = -95dBm #CC2420
41 **.baseStation[*].nic.phy.sensitivity = -92dBm #XBEE
42
43 ***.netwType = "AODVRoute"
44 ***.net.headerLength = 24 bit
45 ***.mobility.z = 0
46 **.mobileNode[*].arp.offset = 10 #max base stations = 9
47 **.staticNode[*].arp.offset = 1000 #max mobile nodes = 989
48 **.staticNode[*].battery.capacity = 99999mAh #connected to the electric network
49 **.staticNode[*].battery.nominal = 99999mAh #connected to the electric network
50 **.baseStation[*].battery.capacity = 99999mAh #connected to the electric network
51 **.baseStation[*].battery.nominal = 99999mAh #connected to the electric network
52
53
54 [Config testbed1]
55 **.numHostsStatic = 9
56 **.numHostsMobile = 1
57 **.numHostsBase = 1
58 **.staticNode[0].mobility.x = 1.49
59 **.staticNode[0].mobility.y = 1.49
60 **.staticNode[1].mobility.x = 2.39
61 **.staticNode[1].mobility.y = 4.78
62 **.staticNode[2].mobility.x = 4.78
63 **.staticNode[2].mobility.y = 1.49
64 **.staticNode[3].mobility.x = 6.88
65 **.staticNode[3].mobility.y = 5.68
66 **.staticNode[4].mobility.x = 6.88
67 **.staticNode[4].mobility.y = 3.29
68 **.staticNode[5].mobility.x = 7.48
69 **.staticNode[5].mobility.y = 1.49
70 **.staticNode[6].mobility.x = 8.75
71 **.staticNode[6].mobility.y = 1.49
```

```

72 **.staticNode[7].mobility.x = 10.47
73 **.staticNode[7].mobility.y = 5.68
74 **.staticNode[8].mobility.x = 10.77
75 **.staticNode[8].mobility.y = 1.79
76 **.mobileNode[0].mobility.x = 0.5
77 **.mobileNode[0].mobility.y = 0.5
78 **.baseStation[0].mobility.x = 0.6
79 **.baseStation[0].mobility.y = 0.6
80
81
82 #offline phase
83 [Config offline]
84 sim-time-limit = 86 min # for calibration
85 extends = testbed1
86 **.stats = false
87 *.obstacles.obstacles = xmldoc("xml_config_files/config.obstacles-01.xml")
88 **.mobileNode[0].app.minimumStaticNodesForSample = 5
89 **.mobileNode[0].app.clusterKeySize = 1
90 **.mobileNode[0].app.maxPositionPDFsSize = 4
91 **.mobileNode[0].app.calibrationMode = true
92 **.mobileNode[0].mobilityType = "TractorMobility"
93 **.mobileNode[0].mobility.updateInterval = 40 s
94 **.mobileNode[0].mobility.x1 = 0.5m
95 **.mobileNode[0].mobility.y1 = 0.5m
96 **.mobileNode[0].mobility.x2 = 12m
97 **.mobileNode[0].mobility.y2 = 7m
98 **.mobileNode[0].mobility.rows = 10 #  $d = 2*(7-0.5) + 10*(12-0.5) = 128$  m
99 **.mobileNode[0].mobility.speed = 0.025 mps #  $128/0.025 = 5120$  s  $\rightarrow$  85.3 min to complete the
    run
100 **.mobileNode[0].mobility.debug = true
101 **.mobileNode[0].mobility.x = 2
102 **.mobileNode[0].mobility.y = 2
103 **.mobileNode[0].app.firstPosX = 2
104 **.mobileNode[0].app.firstPosY = 2
105 **.staticNode[*].app.beaconMode = true
106 **.staticNode[*].app.nodeSigStartingTime = 1
107 **.staticNode[*].app.nodeSigPeriod = 1
108
109 #online phase
110 [Config online]
111 extends = testbed1
112 sim-time-limit = 40 min
113 **.stats = false
114 *.obstacles.obstacles = xmldoc("xml_config_files/config.obstacles-01.xml")
115 **.mobileNode[0].app.calibrationMode = false
116 **.mobileNode[0].app.minimumStaticNodesForSample = 5
117 **.mobileNode[0].app.collectedDataSendingTimePeriod = 40
118 **.mobileNode[0].mobility.x = 4
119 **.mobileNode[0].mobility.y = 4
120 **.mobileNode[0].app.firstPosX = 4
121 **.mobileNode[0].app.firstPosY = 4
122 **.mobileNode[0].mobilityType = "TurtleMobility"
123 **.mobileNode[0].mobility.updateInterval = 0.1 s
124 **.mobileNode[0].mobility.turtleScript = xmldoc("xml_config_files/config.mobility-01.xml")
125 **.baseStation[0].app.radioMapXML = xmldoc("xml_data_files/radioMap.xml")
126 **.baseStation[0].app.radioMapClustersXML = xmldoc("xml_data_files/radioMapClusters.xml")
127 **.baseStation[0].app.normalStandardDistribXML = xmldoc("xml_data_files/normalStandard.xml")
128 **.baseStation[0].app.useClustering = true
129 **.baseStation[0].app.useCenterOfMass = true
130 **.baseStation[0].app.usePhySpaceTimeAvg = false
131 **.baseStation[0].app.recordStats = true
132 **.baseStation[0].app.maxCenterOfMassPositions = 6 #HORUS VALUE
133 **.baseStation[0].app.phySpaceTimeAvgWindowSize = 10 #HORUS VALUE
134 **.staticNode[*].app.beaconMode = true
135 **.staticNode[*].app.nodeSigStartingTime = 1
136 **.staticNode[*].app.nodeSigPeriod = 1

```
