

PM

Teste e Validação de Redes de Sensores Sem Fios com Simulação, Aplicado ao Museu da Baleia

PROJETO DE MESTRADO

Filipe Rodrigo Ornelas de Freitas

MESTRADO EM ENGENHARIA INFORMÁTICA



UNIVERSIDADE da MADEIRA

A Nossa Universidade

www.uma.pt

junho | 2019

Teste e Validação de Redes de Sensores Sem Fios com Simulação, Aplicado ao Museu da Baleia

PROJETO DE MESTRADO

Filipe Rodrigo Ornelas de Freitas

MESTRADO EM ENGENHARIA INFORMÁTICA

ORIENTAÇÃO

Eduardo Miguel Dias Marques

CO-ORIENTAÇÃO

Lina Maria Pestana Leão Brito



**Teste e Validação de Redes de Sensores Sem Fios com Simulação,
Aplicado ao Museu da Baleia
Filipe Rodrigo Ornelas de Freitas**

Constituição do júri de provas públicas:

Filipe Magno de Gouveia Quintal, (Professor Auxiliar Convidado da Universidade da Madeira), Presidente

Lina Maria Pestana Leão de Brito, (Professora Auxiliar da Universidade da Madeira), Vogal

Karolina Baras, (Professora Auxiliar da Universidade da Madeira), Vogal

Junho 2019

Resumo

Os recentes desenvolvimentos na área das redes de sensores sem fios (Wireless Sensor Networks, WSN), originaram notórias melhorias ao nível do seu hardware e software e, por consequência, melhorias na sua viabilidade, condições de implementação, custos e tempo de vida útil. Por estas razões, são cada vez mais os projetos que tiram proveito destas potencialidades, abrangendo os mais diversos campos de aplicação, desde o militar até à domótica. O projeto WISE-MUSE, baseia-se na implementação de WSN em ambientes interiores (indoor), e tem como objetivo a monitorização remota de parâmetros ambientais, vitais neste caso específico, à gestão dos artefatos expostos no Museu da Baleia. Este tipo de WSN, indoor, define-se por uma maior densidade de obstáculos (paredes), logo com distâncias de transmissão mais curtas, uma maior reflexão dos sinais, mas também pela sua flexibilidade na implementação. O estudo destas características, executado com o recurso a um simulador de rede, permite não só manter a rede do museu em pleno funcionamento, como também ter flexibilidade ao nível dos cenários gerados e capacidade de otimização. Com base nas simulações realizadas, procedeu-se à análise da topologia, do seu (sobre) dimensionamento e da (sobre) carga na rede, determinando no decorrer do processo, indicadores como a melhor posição teórica da estação base e a sua capacidade máxima de transmissão para uma determinada taxa de envio de pacotes. Apesar de apresentada no contexto do Museu da Baleia, considera-se a metodologia utilizada, válida, num contexto mais abrangente de cenários indoor.

Palavras Chave

Redes de sensores sem fios; simuladores; simulações

Abstract

Recent developments in the area of wireless sensor networks (WSN) have led to significant improvements in their hardware and software and, consequently, improvements in their viability, implementation conditions, costs and lifespan. For these reasons, there are more and more projects that take advantage of these potentialities, covering the most diverse fields of application, from military to home automation. The WISE-MUSE project is based on the implementation of WSN in a indoor environment, and aims to remotely monitor environmental parameters, vital in this specific case, to the management of the artifacts exhibited at the Whale Museum. This type of indoor WSN is defined by a greater density of obstacles (walls), therefore with shorter transmission distances, a greater refraction of the signals, but also by its implementation flexibility. The study of these characteristics, executed with the use of a network simulator, allows not only to keep the museum network fully operational, but also to have flexibility in the scenarios generated and the ability to optimize the current solution. Based on the runned simulations, we analyzed the topology, its (over) sizing and (over) loading, determining in the the process, indicators such as the best theoretical position of the base station and its maximum transmission capacity for a given packet rate. Although presented in the context of the Whale Museum, the methodology used is considered valid in a more comprehensive context of indoor scenarios

Agradecimentos

Dedicado aos meus orientadores e às mulheres da minha vida (SF, TF e MF).

Conteúdo

1	Introdução	1
2	Estado da arte	4
2.1	Redes de sensores sem fios	4
2.1.1	Campos de aplicação	5
2.1.2	Desafios e condicionantes	6
2.1.3	Arquitetura de um nó	8
2.1.4	Arquitetura de uma rede	9
2.2	Arquitetura	11
2.2.1	Camada física	11
2.2.2	Camada de acesso ao meio	12
2.2.3	Camada de rede	13
2.3	Gestão da rede e dos nós	16
2.3.1	Implantação e cobertura	16
2.3.2	Gestão da energia	17
2.3.3	Sincronização dos relógios	18
2.3.4	Localização	20
2.3.5	Segurança	21
2.4	IEEE 802.15.4	22
2.4.1	O padrão 802.15.4	22
2.5	Simuladores e Emuladores	24
2.5.1	Simuladores	24
2.5.2	Emuladores	26
2.5.3	Conclusão	28
3	Museu e contextualização das simulações	29
3.1	Museu e o projeto WISE-MUSE	29
3.2	Contextualização do modelo	30
3.2.1	Avaliação e seleção do simulador	31
3.2.2	Hardware	35
3.2.3	Disposição espacial	35
3.2.4	Condições de transmissão	37
3.2.5	Outras considerações	38
3.3	Conclusão	39
4	Análise à simulação e resultados	40
4.1	Estudo prévio	40
4.1.1	Período de transição	41
4.1.2	Período de estabilização	42
4.1.3	Número de repetições da simulação	43

4.2	Análise à topologia da rede	47
4.2.1	Conetividade	47
4.2.2	Otimização da posição para a estação base	55
4.2.3	Conclusões sobre a topologia da rede	57
4.3	Análise à sobrecarga da rede	58
4.3.1	Wiseroute	59
4.3.2	Flooding	65
4.3.3	Conclusões sobre a sobrecarga da rede	68
5	Conclusões e Trabalho Futuro	69
	Appendices	75
	Apêndice A Especificações técnicas	75
	Apêndice B Plantas do museu	76
	Apêndice C Ficheiro de configuração	81
	Apêndice D MiXiM	93
	Apêndice E Análise estatística à escalabilidade do <i>Wiseroute</i>	94
	Apêndice F Resultados: simulações com 5 pacotes	98

Lista de Figuras

1	Evolução de protocolos MAC [1]	12
2	Taxonomia dos mecanismos de controle de topologia [2].	17
3	Efeito do DSSS.	22
4	Planta do piso 1	36
5	Mapeamento de acordo com o material fornecido	37
6	Grau de correlação com 500 ciclos	43
7	Uma repetição	44
8	Vinte e cinco repetições	44
9	Quinhentas repetições	45
10	Simulação <i>template</i> com o SPL	49
11	Simulação <i>template</i> com o SPL e <code>rssThreshold=-96dBm</code> , global	50
12	Simulação com o SPL e $\alpha=4.0$	51
13	Simulação <i>template</i> com o LNS	52
14	Últimos nós a serem descobertos com o módulo LNS	53
15	LNS com $\alpha=4.70$	53
16	LNS com $\alpha=5.30$	54

17	LNS com $\alpha=5.45$	54
18	Número de retransmissões no cenário de envio de 5 pacotes sincronizados e não sincronizados num intervalo de 1 minuto, com taxa de transmissão de 1p/100ms.	62
19	Número de retransmissões para simulações com o envio de 5 pacotes por nó de forma sincronizada utilizando uma combinação de ambos os modelos analógicos e uma taxa de transmissão de 1p/100ms.	63
20	Pisos sobrepostos com a localização dos nós identificados com maior número de retransmissões	64
21	Caso particular onde nem todos os nós recebem os pacotes enviados por broadcast: envio de uma mensagem do nó 0 para o nó 1.	65
22	Comparação entre os níveis de interferência perante o envio sincronizado de 5 pacotes por nó entre os protocolos Wiseroute e Flooding.	67
23	Níveis de latência perante o envio sincronizado de 5 pacotes por nó com o protocolo Flooding.	67
24	Especificações técnicas: Xbee S2 e XBee PRO(S2B)	75
25	Planta do Piso 2	77
26	Planta do Piso 1	78
27	Planta do Piso 0	79
28	Planta do Piso -1	80
29	Disposição dos nós em torno da estação para as três primeiras iterações	95
30	Teste com nós dispostos em círculo, durante 3 ciclos.	96
31	Receção de pacotes para a criação de rotas	96
32	Disposição dos nós em torno da estação para as duas primeiras iterações	97
33	Teste com nós dispostos em dois círculos, durante 3 círculos.	98
34	Uma repetição	99
35	Vinte e cinco repetições	99
36	5 pacotes espaçados em um minuto.	100
37	5 pacotes sincronizados	100

Lista de Tabelas

1	Desafios e mecanismos necessários para os superar	8
2	Classificação dos protocolos de acordo com a sua utilização nas WSN	14
3	Caraterísticas de protocolos aplicados em multi camadas.	15
4	Diferenças entre simuladores e emuladores.	24
5	Comparação entre Castalia e MiXiM.	34
6	Principais caraterísticas do hardware utilizado.	35
7	Análise da densidade às divisórias do museu	38
8	Valores para o expoente de perda de sinal	38
9	Adaptação da tabela	42
10	Relação entre o grau de confiança dos resultados e o número de repetições	45

11	Adaptação da tabela	56
12	Pacotes recebidos na estação base nas simulações com o envio de 1 pacote por nó, num total de 84, com envio sincronizado e aleatório no intervalo de 1 minuto.	60
13	Resultados para os pacotes recebidos nas simulações com o envio sincronizado de 5 pacotes (num total de 420).	61
14	Resultados para os pacotes recebidos nas simulações com o envio não sincronizado num intervalo de 1 minuto de 5 pacotes (num total de 420)	61
15	Resultados para as latências nas simulações com o envio de 5 pacotes, com e sem sincronização num intervalo de 1 minuto.	61
16	Resultados para o envio de 5 pacotes espaçados em intervalos de 1, 2 e 3 minutos.	63
18	Pacotes recebidos nas simulações com o envio de 5 pacotes (em 420 possíveis).	66
17	Número de pacotes recebidos nas simulações com o envio de 1 pacote (em 84 possíveis).	66

Listings

1	Disposição espacial	36
2	Mobilidade	39
3	<i>Bitrate</i>	39
4	Utilização do modelo analógico SPL	49
5	Limite do RSSI máximo de acordo com o hardware	49
6	Utilização do modelo analógico LNS	51
7	Posições dos nós 74 e 78	56
8	Dessincronização	60
9	Exemplo de Flooding	65
10	Template base para a WSN do Museu da Baleia	81
11	Connection Manager	93
12	Parâmetros referentes à distância de comunicação	94

Lista de Acrónimos

AES	<i>Advanced Encryption Standard</i>
AODV	<i>Ad hoc On-Demand Distance Vector</i>
CCA	<i>Clear Channel Assessment</i>
CSMA-CA	<i>Carrier Sense Multiple Access with Collison Avoidance</i>
DSSS	<i>Direct Sequence Spread Spectrum</i>
GUI	<i>Graphic User Interface</i>
GPS	<i>Global Positioning System</i>
GTS	<i>Grant Time Slot</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IoT	<i>Internet of Things</i>
LNS	<i>Log Normal Shadowing</i>
LR-WPAN	<i>Low Rate - Wireless Personnal Area Network</i>
MAC	<i>Medium Access Control</i>
MiXiM	<i>Mixed Simulator</i>
NED	<i>Network Description Language</i>
NS-3	<i>Network Simulator version3</i>
OMNeT++	<i>Objective Modular Network Testbed in C++</i>
PAN	<i>Personal Area Network</i>
PLME-ED	<i>Physical Layer Management Entity - Energy Detect</i>
QoS	<i>Quality of Service</i>
RSSI	<i>Received Signal Strenght Indicator</i>
SNR	<i>Signal-to-Noise Ratio</i>
SPL	<i>Simple Path Loss</i>
WSN	<i>Wireless Sensor Networks</i>

1 Introdução

As redes de sensores sem fios consistem em redes de nós com energia e recursos limitados que possuem sensores e/ou atuadores para monitorizar e modificar o estado do meio em que se inserem. Os exemplos de campos de aplicação desta tecnologia são os mais variados, desde o militar, onde os sensores são utilizados para a monitorização dos campos de batalha, ao nível das redes corporais, de modo a registar indicadores vitais de soldados ou pacientes em hospitais, no meio ambiente, com o intuito de monitorizar indicadores de inundações ou incêndios, em urbanismo, para o controlo e gestão de trânsito ou acompanhamento do estado de conservação de pontes ou outras infraestruturas, e na indústria e agricultura, na gestão de maquinaria ou localização de gado. Toda esta variedade de cenários de aplicação reflete-se depois numa multiplicidade de desafios. As aplicações têm de responder a critérios específicos, o tipo de interação com o ambiente é variável (chuva, vento, ação humana), as redes podem ser constituídas por milhares de nós, havendo a necessidade de existir mecanismos de tolerância a falhas de modo a suprir-las e assim manter a cobertura desejada, e a necessidade de auto configuração e de processamento centrado em dados, para assim tornar a rede mais eficiente. Outro grande desafio é a gestão de energia, limitada na maioria dos cenários, que tem um impacto direto na vida útil do projeto. Por outro lado, o recurso a campos de teste é limitado pois a replicação de cenários com elevado número de nós não é exequível. A gestão de redes desta natureza foca-se depois nos desafios de implantação e cobertura do cenário, nem sempre as condições são as mais favoráveis, podendo existir depois necessidades específicas de sincronização de relógios, localização, segurança, consoante o cenário e objetivos em causa.

A presente dissertação visa complementar analiticamente uma rede de sensores sem fios¹ instalada no Museu da Baleia, cujos objetivos passam por ajudar na conservação, monitorização e segurança dos artefatos expostos.

Para tal, utiliza-se uma metodologia que aborda os diferentes passos que se assumem comuns a este tipo de desafio, desde a contextualização da temática das WSN e o levantamento posicional dos sensores, até à seleção de simulador e a execução das simulações. Como resultado, apresentar-se-ão, não só considerações estatísticas importantes para a utilização e validação deste tipo de modelo em simulações, como também análises e sugestões à sua topologia e tráfego.

Com base nesta abordagem (simulação), é possível criar um modelo que permite recriar um elevado número de características dos cenários *indoor* de uma forma flexível e dinâmica. A capacidade de configurar/modelar características como a densidade de obstáculos, as distâncias de transmissão, a reflexão dos sinais, a posição e características específicas de cada nó, o tráfego introduzido na rede (quantidade, frequência) e a duração das simulações, possibilita não só uma análise cuidada e pormenorizada ao nível da implementação inicial de uma rede WSN, como também, de forma não intrusiva, ao estudo posterior da mesma (já em funcionamento). A resposta a estas necessidades

¹do inglês, *Wireless Sensor Networks*, WSN

é pois, dada através da seleção de um simulador. Esta foi feita após uma pré-seleção cujos critérios utilizados foram o facto de ser *open source*, poder simular cenários *indoor* e suportar a norma IEEE 802.15.4.

Após a seleção e configuração do simulador, e com vista ao estudo da topologia, do seu (sobre) dimensionamento e da (sobre) carga na rede, as simulações permitem identificar indicadores como a melhor posição teórica da estação base e a sua capacidade máxima de transmissão para uma determinada taxa de envio de pacotes. A mesma abordagem pode ser ainda aplicada a outros tópicos de interesse, como por exemplo a identificação de áreas críticas, isto é, áreas cuja funcionalidade e cumprimento dos objetivos pudesse ser posta em causa pela falha de um ou mais nós.

A presente metodologia pode ainda ser implementada em diferentes contextos, como por exemplo o hospitalar, monitorizando desde indicadores vitais e localização dos pacientes, até às condições ambientais e das infraestruturas.

Objetivos

O presente projeto, em concordância com os objetivos funcionais da rede já instalada, visa então alcançar os seguintes objetivos:

- Estudar as redes de sensores sem fios.
- Estudar a área da simulação de redes, em particular das redes de sensores sem fios.
- Escolher um simulador adequado para simulação de WSNs *indoor*.
- Modelar as condições identificadas de modo a executar o estudo através da simulação.
- Simulação da rede de sensores sem fios implementada no Museu da Baleia.
- Análise à posição da estação base.
- Estudo e otimização do tráfego presente na rede do museu.
- Identificação de zonas críticas na rede.

Estrutura do documento

O processo inicia-se assim com um levantamento sobre esta temática, desde a sua importância e dos seus campos de aplicação até aos desafios inerentes, passando pelas cadeias protocolares, hardware e simuladores utilizados para a sua análise. Os conceitos relevantes são identificados e clarificados, ficando também a menção para outras áreas de interesse mais limitado. O objetivo deste capítulo é introduzir os leitores à temática, dotando-os dos conhecimentos mínimos necessários para a leitura da mesma.

Prossegue-se no capítulo seguinte com a contextualização entre o projeto WISE-MUSE implementado no Museu da Baleia, os seus objetivos, e a utilização de simuladores de modo a testar configurações que conduzam a otimizações da rede. Em relação a estes, são explorados os simuladores NS-3 e as *frameworks* Castalia e MiXiM do OMNeT++. Após uma comparação, este último é escolhido para proceder a uma série de simulações, realçando-se sempre que possível as suas virtudes e limitações.

O capítulo 4 identifica indicadores estatísticos subjacentes às simulações, nomeadamente, o período de estabilização da rede, isto é, o número de ciclos necessários para o reconhecimento total dos nós da rede, o período de estabilização, representado pelo número de ciclos necessários para que se possa considerar como estáveis os resultados produzidos pela rede e, o número de repetições da simulação, de modo a tornar os resultados independentes das *seeds* utilizadas em cada simulação, de acordo com o grau de confiança pretendido e com os recursos disponíveis. Através deste levantamento, analisou-se a rede no que respeita à topologia usada, às condições de sobrecarga e ao seu sobredimensionamento, verificando-se, de um ponto de vista analítico, sobre a posição da estação de recolha de dados, o número de nós necessários para que as mesmas funcionalidades sejam garantidas e sobre o congestionamento crítico de certas rotas para a estação base.

Por fim, resumam-se os resultados e conclusões obtidas ao longo da preparação e realização das simulações, nomeadamente em relação aos modelos analógicos utilizados, à posição ótima para a estação base, à densidade da rede, à identificação de estrangulamentos e ainda sobre a eficácia das transmissões, tendo em conta as taxas e condições de transmissão. Apontam-se por fim direções para a continuação do trabalho apresentado, e áreas de exploração para este tipo de implementações *indoor*.

De modo a providenciar alguma claridade e rastreabilidade ao processo utilizado ao longo deste trabalho, em especial no que se refere à edição dos ficheiros de configuração utilizados no simulador, foram inseridos alguns fragmentos destes ficheiros salientando as instruções relevantes ou alteradas para os efeitos dessas simulações. Nos anexos apresentam-se ainda informação técnica sobre o hardware usado e de suporte à escolha de determinados parâmetros utilizados nas simulações.

2 Estado da arte

O presente capítulo faz uma introdução ao conceito de redes de sensores sem fios (*Wireless Sensor Networks*, WSN) desde como funcionam, que desafios se põem à sua utilização, explicando também as arquiteturas de nós e redes utilizadas.

Uma descrição mais técnica é depois realizada ao abordar as camadas física, de acesso ao meio e de rede, assim como à gestão deste tipo de redes, nomeadamente as questões da implantação e cobertura, energia, sincronização temporal, localização e segurança. O capítulo termina com uma introdução aos padrões utilizados nesta área, e com um levantamento aos simuladores e emuladores de WSN, que serve depois como base para a escolha do simulador utilizado nos capítulos seguintes na análise da rede implementada no Museu da Baleia.

2.1 Redes de sensores sem fios

Na presente secção apresentam-se as redes de sensores sem fios (WSN), contextualizando inicialmente as razões pelas quais estão a proliferar e diversos campos de aplicação. É feita uma introdução às WSN, desde como sentem com os seus sensores o meio no qual se inserem, ao modo como colaboram com outros nós de modo a atuar ou transmitir a informação para uma estação base. Por fim, complementam-se as interações anteriores com a descrição das arquiteturas utilizadas para a constituição dos nós, assim como para as suas redes.

As redes de sensores sem fio (WSN) consistem em redes de nós com energia e recursos limitados que possuem sensores e/ou atuadores para monitorizar e modificar o estado do meio em que se inserem. Os nós podem comunicar sem fios e ter recursos computacionais suficientes para executar tarefas colaborativas de processamento de sinal. Uma visão de referência destas redes pode ser identificada no conceito de "poeira inteligente" na qual os nós sensores tornar-se-ão tão pequenos e baratos que podem ser incorporados em quase todos os lugares tornando o nosso meio ambiente inteligente [3]. Algumas das principais frentes de desenvolvimento tecnológico são portanto a miniaturização das componentes, o *design* de hardware de baixa potência, criação de protocolos e aplicações de software específicas e eficientes, e a capacidade de auto-organização das redes de sensores. A tarefa principal de cada nó é então colaborar com outros nós na monitorização do meio ambiente. As redes WSN são frequentemente projetadas para executar apenas uma aplicação ou muito poucas aplicações relacionadas. Isto deriva do design centrado em dados sendo um exemplo chave a agregação de dados no processo de encaminhamento pelos nós: em vez de encaminhar pacotes imediatamente, cada nó pode coletar um número de medições, calcular um valor agregado a partir destes (por exemplo a média) e encaminhar apenas um único pacote que transporta o valor agregado [4]. Nestas situações a perda da granularidade dos dados é compensada pela redução da largura de banda utilizada e pela economia de energia.

As WSN apresentam propriedades únicas que transformam o seu dimensionamento em projetos singulares nos quais é necessário definir desde a composição de um nó

sensor e da topologia da rede, ao método de implementação e à sua gestão. A definição dos protocolos e aplicações a utilizar neste campo também estão condicionados, especialmente em relação à energia finita disponível nos nós, à sua capacidade de processamento de dados limitada, aos requisitos de escalabilidade para redes com grande densidade de nós e à necessidade de colaboração inteligente entre nós. O desenvolvimento destes novos projetos, sobretudo devido à sua flexibilidade, é o catalisador que permite o aparecimento de novas aplicações envolvendo a interação inteligente entre os nós sensores e o meio ambiente. Exemplos destas novas interações são as implementações criadas para automatização de edifícios, gestão de instalações, agricultura inteligente, cuidados de saúde e monitorização da natureza.

2.1.1 Campos de aplicação

A flexibilidade de adaptação das WSN ao meio potenciou o seu desenvolvimento nos mais diversos campos de aplicação. Em particular, os contextos militar, de redes corporais, meio ambiente, urbanismo, indústria e da agricultura, identificaram desde cedo as potencialidades das WSN tornando-se propulsores da tecnologia. Apresentam-se de seguida exemplos dessas aplicações [5]:

- Militar e prevenção de crimes - utilização de redes inteligentes em operações envolvendo proteção das forças militares com sensores implantados de forma proativa no campo de batalha, seja em redor de bases operacionais avançadas ou em território inimigo. Outras aplicações são a computação de dados, comunicações, vigilância, reconhecimento e deteção de alvos. A valorização da vida humana e a flexibilidade, especialmente no que toca à sua implantação em anfiteatros de guerra, tornam estas soluções atrativas [6].
- Redes corporais - usadas de forma a monitorizar sinais vitais em diversos contextos como o médico (pacientes), militar (soldados) e desporto (atletas), ou na monitorização posicional de pessoas, de uma forma quase transparente para o utilizador final. A comodidade, facilidade de utilização e obtenção de dados automatizada potenciam o seu desenvolvimento nestes campos [7][8].
- Meio ambiente - monitorização da natureza através de sensores dedicados para a medição do calor, pressão, humidade, proximidade, aplicados a vários aspetos da vida selvagem, como a deteção de inundações e fogos florestais, estado de vulcões, glaciares, elevando assim o sentido de prevenção em relação a eventos naturais cuja deteção por vezes peca por tardia. O custo cada vez menor de cada nó e a grande facilidade de implantação, mesmo que em áreas adversas, tornam cada vez mais viáveis a sua utilização [9][10].
- Urbanismo e infraestruturas - redes urbanas inteligentes, gestão de tráfego, monitorização do estado de conservação e segurança estrutural das infraestruturas como edifícios, pontes, viadutos e túneis, casas inteligentes, permitem um maior

estado de consciencialização (*awareness*) em relação ao estado de operação de diversos elementos estruturais [11].

- Indústria e agricultura - proporcionam a utilização de soluções com vista à gestão inteligente de energia, controlo do estado e comportamento de equipamentos, monitorização posicional de gado, sistemas de rega automática, entre outros [12] [13] [14].

Transversalmente surgiu o conceito de *Internet of Things*, Internet das Coisas, que visa proporcionar ao público geral aplicações mais práticas destas tecnologias e como podem ser incorporadas no nosso dia a dia, nos equipamentos que usamos nas nossas redes corporais, na vigilância das nossas casas, no nosso lazer e até mesmo no controlo de forma inteligente dos nossos eletrodomésticos.

2.1.2 Desafios e condicionantes

Os desafios e condicionantes das WSN são um superconjunto dos existentes para as redes sem fio convencionais e redes *ad hoc*. As redes WSN são de uma forma geral, muito mais dinâmicas pois os seus nós podem ficar sem energia, ser destruídos pelo ambiente em que se inserem ou simplesmente levados, o que condiciona todo o seu ciclo de vida, desde a sua conceção, à obtenção dos seus objetivos (muito mais restrito).

Levantam-se de seguida algumas das problemáticas inerentes a estas redes:

- Especificidade das aplicações: dada a abrangência da tecnologia, tornando possível a sua utilização nos mais variados contextos, as soluções aplicacionais e técnicas têm de ser criadas especificamente de modo a corresponder aos requisitos de cada cenário e objetivo. A grande variância nos campos de aplicação torna-se também por isso, num dos maiores entraves colocados à proliferação da tecnologia visto que existe pouca reutilização entre cenários.
- Tipo de interação com o ambiente: dada a natureza de muitos dos cenários de aplicação, o tipo de tráfego gerado é atípico, podendo a rede estar inativa durante longos períodos e reportar com grandes taxas de transferência perante a ocorrência de eventos.
- Escala: dependendo da precisão pretendida para o cenário, redes com grande densidade de nós podem ser implementadas. Para tal os protocolos, além de garantirem eficiência em relação ao consumo de energia, também têm de garantir robustez e um grau de performance adequado.
- Tolerância a falhas, redundância: as WSN estão sujeitas a falhas derivadas de dano físico dos próprios ambientes nos quais estão implantados, devido a falhas no seu hardware ou simplesmente por exaustão das suas baterias. A rede tem de ser robusta de modo a ser capaz de superar estes problemas, garantindo ao mesmo tempo o seu propósito.

- Energia: dada a incapacidade de interação com os nós em determinados cenários após a sua implantação, não existe a oportunidade de recarregar baterias pelo que os nós têm de ser o mais eficientes possível. Esta característica tem impacto profundo quer na arquitetura do próprio nó, quer na da rede. Os protocolos utilizados são definidos de acordo com esta restrição, assim como a seleção dos sensores e qualquer tipo de hardware adicional.
- Auto-configuração: o grande número de nós e a inacessibilidade dos mesmos após a sua implantação, fazem com que estes tenham de ter a capacidade de se configurar automaticamente e de se ajustar perante alterações no cenário, sejam ambientais ou mesmo ao nível do tráfego gerado.
- Processamento centrado em dados: a redundância dos dados é característica neste tipo de redes, sendo mais importante o resultado da informação processada por um conjunto de nós que deteta um evento, do que os dados de um só nó que deteta esse mesmo evento. Esta mudança de paradigma tem repercussões em toda a arquitetura da rede.
- Múltiplos tipos de tráfego: dependendo dos objetivos das redes de sensores sem fios, estas podem apresentar tráfego periódico, envio de dados para a estação base em intervalos fixos, baseado nos eventos detetados (aleatórios) ou nas consultas efetuadas.
- Ferramentas de suporte para o *design*, implantação e gestão das WSN: existe a necessidade de ferramentas de software que possibilitem a otimização das soluções, desde a criação de planos de implantação que minimizem os seus custos, mantendo as restrições definidas para o tempo de vida da rede, grau de redundância, latência e área de cobertura, até aos simuladores que permitem a análise do tráfego e dos protocolos a utilizar.
- Segurança e privacidade: os problemas desta natureza nas redes sem fio (*wireless*) convencionais são conhecidos e surgem em parte da utilização de *broadcasts* fazendo com que a informação transmitida seja facilmente capturada. Vários esquemas de criptografia e protocolos de autenticação foram desenvolvidos para cumprir metas de segurança como confidencialidade, integridade ou disponibilidade e para proteger a rede contra ameaças como espionagem, impersonificação ou sabotagem.
- Campos de teste: falta de condições para simular cenários de grande escala, com por exemplo, milhares de nós.

Mesmo perante estas dificuldades é importante apontar que a comunidade académica está ativa e já apresenta algumas soluções para estas problemáticas. No entanto, o nível de maturidade necessário para transformar as ferramentas disponíveis de forma compatível com as necessidades profissionais, ainda não foi alcançado.

Desafios	Mecanismos necessários
Escassez de recursos	Uso eficiente de recursos
Condições ambientais extremas e dinâmicas	Operações de rede ajustáveis
Redundância de dados	Fusão de dados e processamento localizado
Comunicações sem fios confiáveis	Garantia de confiança
Ausência de indentificadores globais (ID) para nós sensores	Paradigma de comunicações centradas em dados
Suscetível a falhas de nós	Tolerância a falhas
Implementações em grande escala	Sensores de custo baixo, pequenos, auto-configuráveis e com capacidade de se integrar na rede

Tabela 1: Desafios e mecanismos necessários para os superar, traduzido de [15].

A tabela seguinte apresenta orientações/mecanismos que podem ser utilizados na abordagem aos desafios enunciados.

2.1.3 Arquitetura de um nó

Cada nó de uma rede WSN consiste numa combinação de dispositivo de comunicação, microcontrolador/processador, memória, sensores e fonte de energia.

O transmissor/recetor sem fio (*wireless*) executa modulação, transmissão/receção e desmodulação de dados digitais através do canal *wireless*. Uma vez mais, os transmissores não são escolhidos de forma a otimizar o débito de dados ou as taxas de erro dos pacotes, mas sim para minimizar o consumo de energia. A maioria dos transmissores neste contexto têm uma potência de saída na ordem de 1 mW, enquanto que, por exemplo, a IEEE 802.11 WLAN usa 100 mW e um telemóvel pode usar ainda mais. Além disso, as taxas de dados permitidas por estes transmissores estão limitadas ao intervalo das dezenas a centenas de kbits por segundo.

O principal objetivo do microcontrolador/processador é processar (executar) instruções referentes à detecção, comunicação e auto-organização da rede, juntamente com uma memória não volátil (geralmente uma memória *flash* interna) que armazena as instruções do programa, e uma memória ativa que armazena temporariamente os dados detectados. O microcontrolador executa o *software* aplicacional e a *stack* de protocolos. Em geral, os microcontroladores usados em cenários onde os nós sensores tendem a ser otimizados para o consumo de energia e não para o desempenho. Por outro lado, se os objetivos da monitorização forem bem definidos e não variarem ao longo do tempo de vida útil da rede, os processadores podem ser utilizados pois são muito eficientes em termos de consumo de energia e para a maioria das tarefas simples de detecção. No entanto, como estes não são processadores genéricos (*off the shelf*), o

processo de design e implementação pode ser complexo e caro [16]. Caso os cenários proporcionem alterações nos alvos de monitorização ou alterações frequentes, como a alteração do *software* executado nos nós sensores sem fio ou configurações remotas, isto é, requerem instalações e atualizações de forma dinâmica, o uso de microcontroladores pode ser mais adequado. Estas tarefas requerem uma quantidade considerável de recursos, espaço e computação pelo que não são adequadas para processadores.

A informação processada por estes provém dos circuitos dos sensores/atuadores que contêm transdutores e conversores analógico-digital/digital-analógico. Estes são capazes de monitorizar por exemplo a temperatura, luz, humidade e movimento.

Por fim, a fonte de alimentação fornece energia necessária ao nó sensor, sendo a bateria a fonte primária mais frequentemente utilizada. As baterias têm uma capacidade de armazenamento de energia finito, embora alguns tipos de baterias se recarreguem parcialmente de acordo com certos processos químicos, só o podendo fazer quando o utilizador não consome energia por um determinado período de tempo. Várias investigações abordam modos alternativos de recarregar baterias extraíndo energia do ambiente, como por exemplo, de vibrações ou gradientes de temperatura. Estas técnicas de recolha de energia [17], quando combinadas com estratégias de gestão de energia ao nível do nó, podem prolongar o tempo de vida do nó de forma significativa, [16]. Em muitas aplicações de WSN é impraticável ou mesmo impossível substituir ou recarregar as baterias dos nós. Consequência direta do armazenamento de energia finito é um tempo de vida finito do nó.

2.1.4 Arquitetura de uma rede

A energia consumida nas transmissões é proporcional ao quadrado da distância² (1), pelo que transmissão direta de dados para o nó recetor irá resultar numa morte precoce dos nós mais afastados deste. Este tipo de transmissão apenas é viável para os nós mais próximos do recetor. Para contornar esta limitação, é utilizada a arquitetura *multi-hop*, baseada em transmissões mais curtas de nó para nó, até chegar ao recetor, reduzindo assim o consumo de energia.

$$P_r = \left(\frac{P_t \times G_t \times G_r \times \lambda^2}{(4 \times \pi \times R)^2} \right) \quad (1)$$

Em redes *ad hoc* não há nós distintos (com identificação única), estes comunicam com outros nós de forma arbitrária. A situação é um pouco diferente nas redes de sensores: muitas vezes existem algumas estações base presentes (ou nós coletores), para os quais outros nós sensores (ou fontes de dados) transmitem os seus dados. As estações base também podem configurar e controlar as operações entre os nós sensores [18].

As estações base são frequentemente mais poderosas que os nós de sensores comuns, podendo ser até *laptops* ou computadores de mesa. Uma estação base é muitas vezes

²Fórmula de transmissão de *Friis*.

o ponto onde um utilizador interage com a rede de sensores, executando consultas e visualizando resultados. É também o ponto de transição natural entre a rede de sensores e outras redes como a Internet [19].

- Topologia em estrela: uma rede em estrela é uma topologia de comunicações onde uma única estação base pode enviar e/ou receber mensagens para um determinado número de nós remotos. Por sua vez, estes não podem enviar mensagens entre si. As vantagens deste tipo de rede para redes de sensores sem fio incluem a simplicidade, a capacidade de manter o consumo de energia dos nós remotos restrito ao mínimo e a realização de comunicações com baixa latência entre o nó remoto e a estação base. As desvantagens desta topologia são que a estação de base deve estar ao alcance de transmissão de rádio de todos os nós individuais (a implantação nem sempre pode ser controlada), não é tão robusta como outras redes devido à sua dependência de um único nó para gerir a rede (ponto de falha único) e não é apropriada para redes com grande número de nós (centenas ou milhares).
- Topologia em malha: uma rede em malha permite transmitir dados entre nós da rede que estejam dentro do seu alcance de transmissão de rádio. Isto permite o que é conhecido como comunicações *multi-hop*, ou seja, se um nó deseja enviar uma mensagem para outro nó que está fora do seu alcance, ele pode usar um nó intermédio para encaminhar a sua mensagem até ao nó desejado. Esta topologia de rede tem a vantagem de redundância e escalabilidade. Se um nó individual falhar, um nó remoto ainda pode comunicar com qualquer outro nó com a ajuda de nós intermediários. Além disso, o alcance da rede não é necessariamente limitado pelos alcances individuais dos nós, podendo ser aumentado simplesmente através da adição de mais nós ao sistema. Uma das desvantagens deste tipo de rede é o consumo de energia para os nós que implementam as comunicações *multi-hop*, geralmente mais elevados do que para os nós que não têm essa capacidade, limitando muitas vezes a vida útil da bateria desses nós. Além disso, à medida que aumenta o número de saltos na comunicação para o destino, o tempo para entregar a mensagem também aumenta, especialmente se a operação de baixa potência dos nós for um requisito.
- Topologia em estrela híbrida: um híbrido entre a rede em estrela e em malha origina um tipo de rede de comunicações robusta e versátil, mantendo a capacidade de manter o consumo de energia nos nós de sensores no mínimo. Nesta topologia, os nós sensores com menor potência geralmente não têm a capacidade de encaminhar mensagens. Isso permite que o consumo mínimo de energia seja mantido. No entanto, outros nós na rede possuem esta capacidade de *multi-hop*, permitindo-lhes encaminhar mensagens de nós sensores para outros nós na rede. Geralmente, os nós com a capacidade de *multi-hop* têm maior potência e, se possível, são muitas vezes ligados diretamente à linha de energia elétrica.

2.2 Arquitetura

As comunicações sem fio apresentam alguns desafios interessantes como por exemplo a largura de banda limitada, o alcance de transmissão limitado e o facto de estarem sujeitas a muitos erros devido a, por exemplo, interferências e atenuação do sinal. A presente secção pormenoriza por isso, o funcionamento dos nós e a sua interação, abordando as características e funções relevantes nas camadas físicas, de acesso ao meio e de rede nas WSN.

2.2.1 Camada física

A camada física é responsável pela seleção da frequência, *carrier frequency generation*, deteção de sinal, modulação e cifragem de dados. Quando é necessário transmitir uma mensagem esta é processada e convertida num fluxo binário (*source encoding*) de forma a responder eficientemente às restrições da largura de banda e da potência do sinal a utilizar. O sinal é depois codificado e modulado de modo a ser transmitido com robustez em relação ao ruído e às interferências. Do lado do recetor, quando a transmissão sem fios é recebida, esta é descodificada (com uma certa probabilidade de erro), com base na relação entre a potência de sinal e a potência do ruído do canal (SNR).

Enquanto que o processo de codificação permite por exemplo, controlar a deteção e correção de erros, a modulação tem impacto direto na transmissão/receção de sinais com antenas curtas, visto que quanto mais curto é o comprimento de onda do sinal transmitido, mais curto é o comprimento da antena, o que permite transmissões com pouca perda e dispersão de sinal na forma de ondas eletromagnéticas.

Como podemos controlar a potência de transmissão e por consequência a probabilidade de ocorrer erros (diminui com o aumento da potência), esta torna-se um dos pontos de otimização possíveis nesta camada visto que, é essencial alcançar um ponto de equilíbrio entre o número de retransmissões aceitáveis, e a energia que estamos dispostos a consumir. Ainda neste contexto o gasto da energia nas transmissões acontece essencialmente devido ao canal e ao custo fixo de operação dos circuitos de transmissão e receção. Como mencionado anteriormente, a perda associada ao canal é proporcional ao quadrado da distância enquanto que a associada às transmissões aumenta linearmente com o número de saltos. Assim, outro foco de atenção é a otimização entre a potência de transmissão e o número de saltos a realizar de modo a minimizar (novamente) a energia consumida e, por consequência, a probabilidade de ocorrer erros.

Como resultado destas considerações, os principais focos de pesquisa nesta camada são a criação de estratégias de gestão eficiente da energia com impacto direto no design do hardware, estratégias para superar os efeitos sobre a propagação do sinal e a criação de esquemas de modulação mais eficientes.

2.2.2 Camada de acesso ao meio

O papel principal desta camada é coordenar o acesso e transmissão sobre um meio, sem fios, comum a vários nós. A interferência e as colisões entre pacotes são as principais causas do insucesso nas comunicações. Existe por isso, a necessidade de haver compensação com o recurso a mecanismos de retransmissão. Tradicionalmente, tal é alcançado otimizando o acesso ao meio, os métodos de transmissão e retransmissão de pacotes, o comprimento destes, utilizando esquemas de modulação/codificação e controlando a potência de transmissão, entre outros [20].

Os protocolos MAC tradicionais não têm como princípios a minimização da energia dispendida, nem da necessidade de estabelecer ligações de comunicação entre milhares de nós enquanto fornece à rede a capacidade de auto-organização e adaptação às mudanças de topologia devido às falhas e dos nós e à sua redistribuição. Por isso os esforços têm-se centrado no desenvolvimento de esquemas de economia de energia no design de hardware, na gestão da topologia [21] [22], em protocolos de camada MAC [23] [24] e em protocolos de roteamento na camada de rede [25] [26]. Projetar protocolos MAC eficientes energeticamente é uma das técnicas que prolongam a vida útil da rede. Além da eficiência energética, latência e *throughput* são também características importantes para consideração no projeto do protocolo MAC para WSNs. Um protocolo MAC devidamente concebido permite que os nós acedam ao canal de uma forma que economiza energia e suporta QoS.

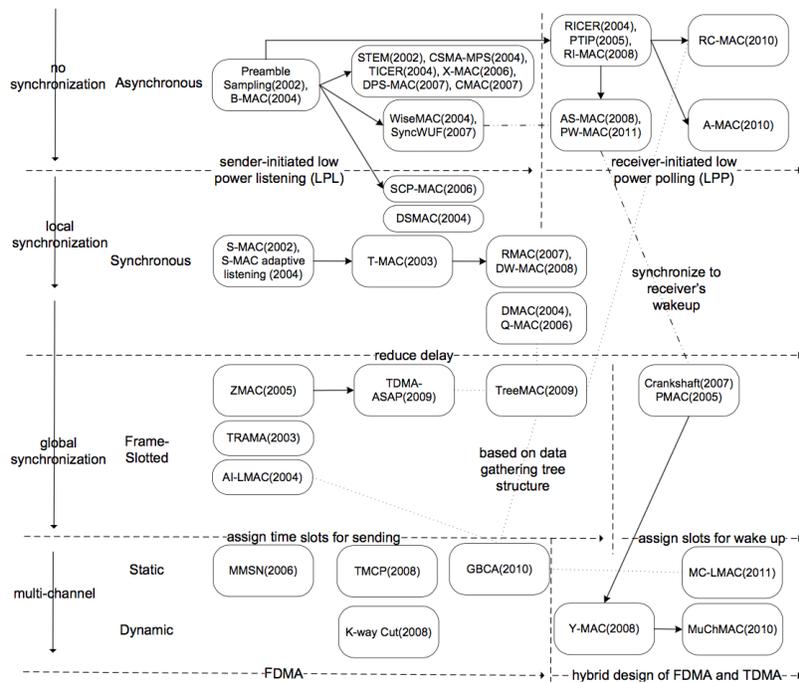


Figura 1: Evolução de protocolos MAC [1]

A evolução destes protocolos está documentado no trabalho de [1], estando os mesmos divididos em quatro grupos: síncrono, assíncrono, *frame-slotted* e multicanal, de acordo com os problemas que procuram resolver (Figura 1). Por exemplo, síncrono e assíncrono estão relacionados com o mecanismo de *duty cycling* em WSN. Para economizar energia, a alteração concertada de estados é amplamente adotada nas WSN. Nesta técnica, cada nó alterna entre os estados ativo e inativo. Dois nós podem comunicar entre si somente quando ambos estão ativos.

Em protocolos MAC síncronos, nós vizinhos (*clusters*) são configurados para acordar ao mesmo tempo. Desta forma, a comunicação é facilitada e o foco está na redução da latência e na melhoria das taxas de transferência. Os protocolos MAC assíncronos, por outro lado, focam-se no estabelecimento com eficiência da comunicação entre dois nós que têm diferentes ciclos (ativo/inativo). Este tipo de protocolo melhora as taxas de transferência com a introdução da sondagem de baixa potência reduzindo também o custo do lado do recetor.

Para obter melhores taxas de transferência, mecanismos de alocação de parcelas de tempo (*time slots*) garantem que nenhum nó dentro de uma mesma vizinhança de comunicação transmite ao mesmo tempo, proporcionando-se assim um ambiente de transmissão de dados sem colisões. No entanto, uma grande preocupação neste contexto é que, a utilização do canal é baixa quando poucos nós têm dados para enviar pois os intervalos de tempo atribuídos aos seus vizinhos são desperdiçados. O foco destes protocolos é o de melhorar a utilização do canal sob baixa contenção. Uma abordagem mais recente aloca as parcelas de tempo não para as transmissões, mas sim às receções de mensagens.

O grupo final, multicanal foca-se em aumentar ainda mais a capacidade da rede através de transmissões paralelas, atribuindo para tal nós a diferentes canais. A atribuição destes e a comunicação eficiente entre canais são dois grandes desafios dos protocolos MAC multicanal.

A tendência futura é combinar o Acesso Múltiplo por Divisão de Tempo (TDMA) com o Acesso Múltiplo por Divisão de Frequência (FDMA) para resolver a deficiência de comunicação entre canais, [28].

As tabelas 2 e 3 apresentam as características de alguns protocolos, nomeadamente no que respeita ao tipo de aplicação nas redes, sejam para consulta do estado das redes, para a deteção de eventos ou para a análise periódica (histórico ou tendências), Tabela 2, e em relação às suas meta-caraterísticas, como a que camada se aplicam, com que objetivos foram criados e as métricas utilizadas para o seu desempenho, Tabela 3.

2.2.3 Camada de rede

A escolha do protocolo a implementar depende em grande parte do interesse do utilizador sobre a rede. Os protocolos de roteamento tendem a ser hierárquicos, pois são utilizados em redes com topologia em malha ou estrela híbrida, mais eficientes em relação à energia e mais escaláveis quando comparados com os desenvolvidos para redes em estrela (a energia dispendida é proporcional ao quadrado da distância de

Protocolo	Tipo de Aplicação		
	Consulta	Evento	Periódico
SPIN	X		
Directed Diffusion	X		
Shah et al.			X
Rumor Routing	X		
CADR	X		
COUGAR	X		
ACQUIRE	X		
GBR	X		
O(1)-Reception Routing Protocol		X	
EMPR	X		
LEACH		X	
EAD		X	
TinyDB	X		
PEGASIS		X	
TEEN		X	
APTEEN			X
UCR		X	
BCDCP		X	
GAF		X	X
MECN		X	
GEAR	X		
GOAFR		X	
MBR		X	
GMREE		X	
Zhao et al. Randomly Shifted Anchors		X	X
Chang et al.			X
Kalpakis et al.		X	X
Minimum Cost Forwarding		X	
SAR		X	X
Energy-Aware QoS Routing Protocol			X
EAD General		X	
SPEED	X		
GET			X

Tabela 2: Classificação dos protocolos de acordo com a sua utilização nas WSN, adaptado de [27].

Protocolo	Camada	Abordagem	Avaliação	Topologia	Objetivo	Métricas
ECPS	MAC, Rede	Modelo Matemático: programação probabilística dinâmica	Experimental	Aleatória (Estática)	Maximizar a probabilidade de enviar o pacote para o seu D em n transmissões	Energia
E2LA	MAC, Rede	Modelo Matemático: programação probabilística dinâmica	Experimental	Aleatória (Estática)	Minimizar a energia	Energia
MAC CROSS	MAC, Rede	Heurística	Hardware de simulação experimental	Aleatória (Estática)	Maximizar o tempo de inatividade	Energia
O-Aloha	Física, MAC	Heurística	Simulação	Aleatória	Maximizar o débito de transmissão	Débito de transmissão
POWER	Física, MAC, Rede	Heurística		Uniforme (Estática)	Maximizar o número de clusters	Energia
Weilian Su	Física, MAC, Rede	Framework	Experimental	Aleatória	Otimizar a performance da WSN	
Shunguang Cui	MAC, Rede	Modelação como otimização de problema	Analítica	Aleatória	Maximizar a vida útil da rede	Tempo de vida útil da rede
Sense-Sleep Trees	MAC, Rede	Heurística	Simulação	Malha	Maximizar a vida útil da rede e a área de cobertura	Tempo de vida útil da rede e energia consumida
Game Theoretic Approach	Física, Aplicação	Teoria de Jogos	Analítica	Aleatória	Minimizar a distorção total	Cobertura da distorção
In Yeup Kong	Física, MAC, Rede	Matemática	Analítica	Aleatória	Maximizar a vida útil da rede	
Cross-Layer Scheduling	MAC, Rede	Heurística	Simulação	Aleatória	Maximizar a vida útil da rede	Tempo de vida útil da rede
Cross-Layer design for cluster	Física, MAC, Rede	Heurística	Simulação	Distribuição uniforme	Maximizar a vida útil da rede	Tempo de vida útil da rede

Tabela 3: Características de protocolos aplicados em multi camadas, adaptado de [27].

transmissão) e podem apresentar características de acordo com o propósito das suas aplicações, isto é, o encaminhamento pode ser reativo, pro ativo, ou uma combinação entre ambos, o híbrido [18].

Os primeiros protocolos apresentavam uma abordagem de *Flooding*, baseados na simples disseminação dos dados por *multicast*, sendo os dados encaminhados em forma de *broadcast* até atingirem o nó de receção. São simples, sem recurso a tabelas de roteamento e com baixos atrasos nas transmissões. Por outro lado, se todos os nós participarem no encaminhamento dos pacotes, existirá um consumo exagerado de energia, reduzindo drasticamente o tempo de vida útil da rede. Introduzem ainda o chamado *data implosion*, isto é, um nó receber o mesmo pacote múltiplas vezes. Para eliminar este problema apareceram os protocolos baseados em *Gossiping* que, em vez de utilizarem o *broadcast* para alcançarem todos os seus vizinhos, escolhem apenas um aleatoriamente, sendo que os dados chegam ao nó de receção, eventualmente. Obviamente esta abordagem pode induzir grandes atrasos na propagação dos dados, especialmente se estivermos perante uma rede de grandes dimensões. Para responder a estes dois problemas, surgiram os protocolos de roteamento centrado nos dados ³, nos quais as

³Data centric routing protocol

rotas são determinadas antes da transmissão de dados, sendo negociadas com base nas *queries* efetuadas pelo nó de receção à rede, as quais especificam os dados pretendidos. Apenas nós com os dados pretendidos prosseguem com a transmissão, escolhendo para tal a rota mais adequada. A principal desvantagem destes protocolos é que dependem da natureza da aplicação em causa, isto é, são adequados a aplicações que se baseiam em *queries* à rede, por oposição, por exemplo, a aplicações de monitorização. Outra desvantagem é que, em redes grandes, a fase de negociação e as rotas com muitos saltos induzem atrasos consideráveis na transmissão de dados, limitando a escalabilidade das redes.

De modo a ultrapassarem estas dificuldades, apareceram os protocolos que implementam hierarquias a estas arquiteturas *multi-hop*. A mudança de paradigma surge, não dos avanços tecnológicos, mas sim devido aos requerimentos das aplicações comerciais orientadas às WSN. De pequenas com nós móveis, o foco comercial passou para grandes redes onde maior parte do tráfego de dados é convergente para alguns *sink nodes*, arrastando com isto os esforços no desenvolvimento e na standardização dos protocolos [29].

2.3 Gestão da rede e dos nós

O funcionamento dos nós e das redes de sensores sem fios rege-se de forma similar às redes Wi-Fi (802.11). No entanto, são os cenários e objectivos característicos das WSN que causam repercussões vincadas na sua morfologia, permitindo que esses mesmos objectivos sejam alcançados. A presente secção aborda estas características nos contextos da implantação e cobertura, gestão de energia, sincronização temporal, localização e segurança das redes.

2.3.1 Implantação e cobertura

As redes de sensores sem fios podem ser implantadas de uma forma pré-determinada, sendo os nós colocados manualmente e de forma eficiente, na área a cobrir. Minimiza-se assim, não só a interferência entre os nós, como o *overlap* da área a monitorizar. No entanto, este tipo de implantação muitas vezes não é possível. Em ambientes inacessíveis ou hostis é necessário o recurso a métodos de implantação *ad-hoc*. Como não é possível garantir uma distribuição eficiente dos nós na área a cobrir, recorre-se à utilização de um número de nós superior ao que de outra forma seria necessário, obtendo redes de grande densidade e sobredimensionadas [18]. Visto que estas redes têm de ser autoconfiguráveis, especial atenção tem de ser prestada a problemas específicos desta forma de implantação, tais como a determinação das posições dos nós sensores de modo a obter a cobertura desejada, a conexão entre nós e a eficiência no consumo de energia, usando o menor número de nós possível [30].

Eventos que ocorrem numa área não monitorizada não serão registados, enquanto que, eventos em áreas com grande densidade de nós provocarão congestionamentos e

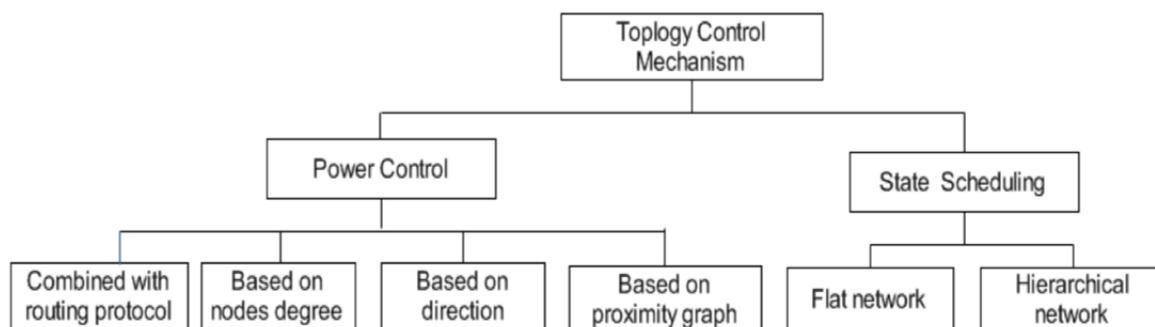


Figura 2: Taxonomia dos mecanismos de controle de topologia [2].

atrasos na transmissão dos dados. Assim sendo, uma implantação ótima proporcionará qualidade de serviço, maior vida útil da rede e poupança nos custos [31].

2.3.2 Gestão da energia

A vida útil de uma rede depende do seu contexto e deve tomar em consideração tanto a sua conectividade, como a sua cobertura. Definições genéricas baseadas no número de nós ativos não se adequam a cenários de redes densas e com redundância, outras, baseadas em percentagens dos nós ativos não têm em consideração que, os nós mortos podem impedir por exemplo, a comunicação com a estação base, e as baseadas no tempo até a falha de um *cluster* ignoram o facto que estes são alternados de modo a balancear o consumo de energia. Por outro lado, definições que consideram apenas a cobertura pretendida não refletem as possíveis falhas ao nível das comunicações. Outra das abordagens é do ponto de vista das aplicações, caso para o qual a vida útil de uma rede é o período durante o qual esta satisfaz os requerimentos da aplicação [32].

A gestão da energia tem obviamente um impacto direto na vida útil da rede e na capacidade de esta desempenhar as suas funções. A energia de cada nó da rede é consumida essencialmente por três componentes: pelo dispositivo de comunicação, na transmissão de dados e como estes são encaminhados para o destino, pelo controlador, na computação dos dados (dependendo diretamente da complexidade dos algoritmos usados) e pelos sensores. Contudo, como a computação consome menos energia que as comunicações, esta pode ser mais complexa, desde que conduza a uma poupança de energia no encaminhamento dos dados. Dependendo da combinação real entre o controlador e dispositivo de transmissão, o controlador pode executar várias centenas ou mesmo milhares de instruções com os mesmos custos de energia necessários para transmitir um bit no canal sem fio, sendo por isso a computação mais económica que a comunicação. Resulta deste facto que, é importante ter um modo eficiente de encaminhar os dados dos nós até ao recetor, mantendo atrasos na transmissão aceitáveis [18]. Como visto anteriormente, numa arquitetura *multi-hop*, mesmo que o caminho percorrido pelos pacotes desde um nó sensor até ao recetor seja maior que uma transmissão direta, existe uma maior poupança de energia ao logo das várias

transmissões pois a energia dispendida é proporcional ao quadrado da distância entre nós. Este facto é uma das grandes vantagens deste tipo de arquitetura.

Também pode ocorrer a situação em que o transmissor está à escuta no canal mas sem realmente receber algo, *idle state*. A energia consumida nesse estado é muitas vezes uma fração significativa (50% ou mais) da energia consumida ao receber um pacote, [16]. Como este tipo de rede (WSN) é propício à ocorrência de falhas de *hardware* e danos acidentais, também é possível que certos nós fiquem sem bateria devido a um uso desequilibrado da sua energia. Estas falhas de energia nos nós podem propiciar ocorrências de falhas na cobertura da rede, sendo que mecanismos de controle da energia e agendamento dos estados dos nós são necessários para equilibrar e otimizar o tempo de vida útil da rede [33]. Os maiores gastos de energia verificam-se nas seguintes situações:

- Colisões, quando um nó recebe mais de um pacote ao mesmo tempo dá-se uma colisão e estes têm de ser descartados, forçando uma nova transmissão;
- Escuta excessiva, quando um pacote é enviado para um nó e todos os outros nós no alcance da transmissão processam esse pacote, mesmo não sendo o seu destinatário;
- *Broadcast* dos pacotes de controlo, isto é, dos pacotes necessários para o estabelecimento dos canais de transmissão;
- Escuta ativa, quando um canal está constantemente à escuta num canal vazio à espera de possível tráfego;
- Interferências, nos nós localizados entre os raios de transmissão e de interferência, pois recebem um pacote mas não o conseguem decodificar.

Por outro lado, as técnicas para a redução de energia podem classificar-se em: redução dos dados transmitidos, redução dos *broadcasts* dos protocolos, encaminhamento eficiente em relação à energia dispendida, ciclos de estados de escuta e controlo de topologia.

Embora o consumo de energia nos nós se deva a diversos fatores como as tarefas de monitorização, receção, transmissão e processamento de dados, é ao nível das transmissões onde estes gastos são maiores. De acordo com a sua implantação e propósito, os nós podem ter potência de transmissão fixa ou variáveis, possuindo mecanismos de ajuste da potência conforme a distância ao próximo nó. Estas decisões têm de ser devidamente ponderadas e tomadas atempadamente.

2.3.3 Sincronização dos relógios

Um dos requisitos das redes de sensores sem fios, com especial incidência nas que apresentam uma grande área de cobertura, é a sincronização do tempo. São, de um modo geral, redes que possuem um elevado número de nós (tendo cada qual o seu

relógio), pelo que tem de ser criada uma linha temporal comum, de modo a cumprir os seus objetivos.

No entanto, para redes as redes WSN, os standards atuais provaram-se desadequados. Tecnologias como o *Global Positioning System* (GPS), que implicam a utilização de antenas dispendiosas e de grande dimensão em relação ao nó, ou o *Network Time Protocol* (NTP), protocolo complexo e com consumo de energia desapropriado a estas realidades, só são utilizados em casos muito particulares, pelo que novas soluções, mais eficientes e adequadas foram pesquisadas [34] [35]. A sincronização de relógios surge como ferramenta de apoio às funções de agregação de dados. Nos casos, em que os dados recolhidos em diferentes nós são processados em colaboração mútua, antes de serem enviados para os nós de receção (evitando redundância nas transmissões), tempos precisos nas partilhas são necessários para obter informações apuradas em relação a um evento detetado. Outro exemplo do aproveitamento desta sincronização, prende-se com conservação de energia inerente à utilização de ciclos precisos de *sleep/wake up* dos nós da rede, evitando períodos excessivos e desnecessários de escuta do meio.

Na camada de acesso ao meio, são vários os protocolos (como por exemplo o já descrito TMDA), cujo o propósito é permitir o acesso de múltiplos nós, a um meio comum de comunicação. Para tal, todos têm de partilhar uma linha temporal, requerendo para tal a sincronização global dos relógios. Evitam-se assim colisões e novas transmissões, prolongando a vida útil da rede.

A sincronização também é requerida para suportar a correta localização de nós numa rede (ver capítulo seguinte). Várias técnicas utilizam tecnologias de estimação de distâncias, recorrendo para tal, à medição do tempo de transmissão, implicando por isso, a sincronização da rede.

De realçar também, que o próprio processo de sincronização tem de ser otimizado para consumir o mínimo de energia possível, recorrendo por isso a um número mínimo de trocas de mensagens entre nós. Além disso, o facto de o meio associado a este tipo de redes ser imprevisível, de performance variável, seja por estar sujeito às adversidades climáticas, a interferências associadas às redes com elevada densidade de nós, ou até pela existência de nós móveis que promovem constantes mudanças na conectividade e topologia, faz com que os protocolos sejam concebidos também com robustez e capacidade de reconfiguração em mente.

A sincronização propriamente dita é feita com o recurso ao *broadcast de mensagens*. Esta pode ser efetuada entre pares de nós e, pela repetição do processo, para a totalidade da rede. Estas mensagens, como quaisquer outras comunicações neste meio, podem ser afetadas pelos níveis de energia dos nós e do estado e tráfego da rede, originando-se atrasos nas comunicações. Estes assumem valores aleatórios e têm de ser tomados em consideração, podendo ser descritos do seguinte modo [35]:

- *Send Time*: tempo dispendido na criação da mensagem na camada de aplicação.
- *Access Time*: tempo de espera para aceder ao canal de transmissão na camada de acesso ao meio.

- *Transmission time*: tempo de transmissão da mensagem na camada física.
- *Propagation time*: tempo da transmissão da mensagem do nó emissor para o recetor, sobre um meio sem fios.
- *Reception time*: tempo necessário para a receção da mensagem na camada física.
- *Receive time*: tempo para reconstruir a mensagem recebida e enviá-la para a camada de aplicação no nó recetor.

2.3.4 Localização

A localização dos nós é fundamental para reportar a origem de um evento monitorizado, na assistência à resolução de consultas (*queries*) feitas à rede, no encaminhamento (quando baseado em localização geográfica), e para averiguar a cobertura da rede. Quando a implantação de uma rede é feita de um modo *ad-hoc*, ou perante a existência de nós móveis, a solução óbvia de pré-determinar a posição dos nós e guardar essa informação nos mesmos, nem sempre é viável. Por outro lado, e como referido no ponto anterior, soluções como o recurso ao GPS não são exequíveis para muitos destes cenários pois o consumo de energia, o custo, o tamanho da antena e até as condições do local de implantação (obstáculos à comunicação) não são complacentes com a realidade de um grande número de aplicações de WSN. Nestes casos, a alternativa é deixar a própria rede organizar-se e determinar as posições dos seus nós, em tempo real, acrescentando assim robustez a alterações no ambiente de monitorização.

Uma das abordagens ao problema da localização passa pela estimativa da distância entre nós. As técnicas de localização baseadas em alcances requerem que, os sensores dos nós sejam capazes de monitorizar características como a intensidade do sinal recebido nas comunicações *wireless*, ou o intervalo de tempo entre a chegada de impulsos de ultrasons.

- *Time of Arrival, ToA*: a distância entre dois nós é calculada com base no tempo de propagação do sinal entre os dois, e no conhecimento da velocidade de propagação do mesmo. Como estas velocidades⁴ requerem medições muito precisas, o grau de sincronização dos relógios tem de ser muito elevada, o que é um custo importante para o tempo de vida útil da rede.
- *Time Difference of Arrival*: usa a mesma técnica do ToA mas com dois tipos de sinal com velocidades diferentes, o que implica o recurso a outro tipo de hardware, sendo uma desvantagem para a utilização desta técnica. Por outro lado esta não requer que os relógios dos nós envolvidos estejam sincronizados e consegue obter estimativas muito precisas.

⁴velocidade da luz: 300km/h, objeto a 10m implica uma medição de 30ns

- *Angle of Arrival*: técnica que usa um conjunto de antenas ou microfones para determinar a direção da propagação do sinal o ângulo de chegada com base numa direção de referência. Consequentemente, a adição de antenas ou microfones pode tornar a solução inapropriada ao cenário, tanto por ser um custo adicional, como por aumentar o tamanho do nó.
- *Received Signal Strength*: baseia-se no decaimento do sinal com a distância percorrida.

De um modo geral, os nós descobrem a sua localização com o recurso a um número limitado de nós da rede cuja localização é pré-determinada ou que, têm mesmo a capacidade de calcular a sua posição global, chamados nós de referência⁵.

2.3.5 Segurança

Todas as soluções de segurança baseiam-se em algoritmos criptográficos simétricos ou assimétricos para cifrar pacotes ou gerir certificados. Os algoritmos criptográficos frequentemente requerem chaves criptográficas que têm de ser distribuídas pelos nós e, cuidar do seu ciclo de vida é parte essencial dos protocolos de gestão de chaves. Efetivamente, a gestão destas chaves acaba por ser a parte mais complexa dos protocolos de segurança. As características especiais das redes de sensores sem fio tornam o planeamento e a implementação adequada dos protocolos de segurança especialmente desafiante e uma importante área de pesquisa.

Os nós sensores são muitas vezes implantados em grandes quantidades, sem haver preocupação de proteger cada nó individualmente do acesso físico. É preciso prever que um invasor será capaz de aceder fisicamente aos nós, ou seja, ler sua memória ou influenciar os seus sensores. Segurança especial nos dispositivos de memória seria necessária para impedir que o invasor acesse à memória, no entanto, estes raramente existirão em nós sensores baratos. As restrições quanto à memória e às capacidades computacionais são também um obstáculo para a implementação de algoritmos criptográficos, especialmente no que respeita à criptografia de chave assimétrica pois é muito exigente computacionalmente para pequenos processadores. Um outro problema é que, a integridade da mensagem é verificada por campos que necessitam estar presentes nos pacotes para garantir a autenticação e integridade da mensagem. Estes campos têm um comprimento mínimo razoavelmente elevado, 16 bytes sendo um valor típico. No entanto, os valores do sensor são muitas vezes de ordem muito menor e a maioria do pacote seria sobrecarga. A capacidade finita de energia dos nós abre outra linha particularmente atraente de ataques: um invasor pode forçar os nós sensores a esgotarem as suas baterias. Uma análise mais detalhada de possíveis ataques de negação de serviço pode ser encontrada em [36].

Tal como na secção anterior, as dificuldades associadas às redes de sensores sem fios condicionam os processos de gestão deste tipo de redes. Em especial, o consumo

⁵termo técnico inglês: anchor node

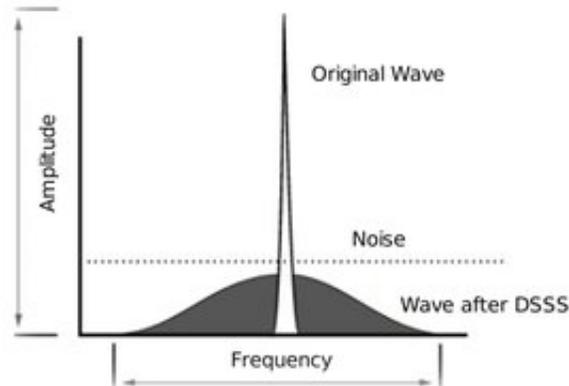


Figura 3: Efeito do DSSS.

de energia, por ter um impacto significativo no tempo de vida útil da rede, é por isso um tema importante de pesquisa desta temática.

2.4 IEEE 802.15.4

O IEEE 802.15.4 permite que as WSN operem com baixas taxas de transferência de dados, baixa complexidade e baixo custo, sendo o ideal para aplicações que requeiram implantação *ad-hoc*, com longa vida útil e com quantidade de nós. Este *standard* suporta as duas topologias mais comuns nestas redes, em estrela e ponto a ponto e modos de poupança de energia suportando estados de *sleep* e *idle*. Garante também atrasos muito baixos a cada salto, permitindo a transmissão de dados em arquiteturas *multi-hop*, evitando assim perdas de energia acentuadas devido a propagação de dados de longo alcance, [18].

2.4.1 O padrão 802.15.4

Uma das vantagens desta norma é a utilização da *Direct Sequence Spread Spectrum*, DSSS, na modulação a informação antes de ser enviada para a camada física. Basicamente, cada *bit* de informação a ser transmitido é modulado em 4 diferentes sinais, processo este que faz com que o total de informação a ser enviado passe a ocupar uma largura de banda maior, usando no entanto, um espectro de densidade de potência menos acentuado (para cada sinal). Este mecanismo causa menos interferência nas bandas de frequência usadas e melhora o rácio sinal/ruído (SNR) no recetor, visto que é mais fácil detetar e decodificar a mensagem transmitida pelo emissor⁶.

Outra das vantagens é a utilização do CSMA-CA, *Carrier Sense Multiple Access with Collision Avoidance* e o GTS *Grant Time Slot*, para evitar que todos os nós comecem a emitir ao mesmo tempo, e da verificação do nível de energia no canal (PLME-ED *request*), averiguando-se assim a quantidade de energia (atividade/ruído/interferências)

⁶<http://www.libelium.com/802-15-4-vs-zigbee>

que existe num canal (ou vários), antes de começar a usá-lo. Deste modo, poupa-se energia escolhendo canais livres aquando da construção da rede, utilizando um dos seguintes três modos:

- Energia: analisa os canais e reporta a energia encontrada. Não interessa se esta foi causada por outro nó, outra tecnologia ou ruído, isto é, reporta apenas se o espectro está a ser usado. Só quando o valor recebido estiver abaixo de um certo limite é que esse nó poderá transmitir.
- CCA *Carrier Sense*: analisa o meio e reporta se houver outras transmissões 802.15.4. Só quando o canal estiver livre é que poderá transmitir.
- Energia+CCA: analisa o meio e reporta se houver transmissões 802.15.4 acima de um limite de energia especificado. Se não houver, poderá transmitir.

O 802.15.4 consome pouca energia pois está preparado para trabalhar em pequenos ciclos de atividade (*low-duty cycle*). Isto quer dizer que o dispositivo de transmissão está inativo a maior parte do tempo (em média, até 99%), enquanto que as tarefas de envio e receção de dados podem ser coordenadas de modo a consumir apenas uma pequena parte da bateria do nó, dependendo esta percentagem do modelo de comunicação utilizado. Por exemplo, se forem usados *beacons* (redes PAN ou em estrela) o mínimo tempo para a transmissão/receção aumenta, pelo que o tempo de utilização do dispositivo de transmissão também.

2.5 Simuladores e Emuladores

A simulação é uma ferramenta/técnica baseada, normalmente, em software, para melhor compreender e analisar uma rede de sensores, e, comparativamente a outras técnicas, encontra-se entre a análise com modelos analíticos, uma abordagem mais teórica e matemática, e o desenvolvimento de rede quase reais, designadas *testbeds*, com equipamentos e materiais similares ao da rede real.

Para efetuar uma simulação é necessário um modelo adequado baseado em pressupostos sólidos e uma *framework* adequada de modo a facilitar a sua implementação. Visto que, os resultados das simulações dependem muito do cenário de estudo e dos pressupostos feitos sobre o hardware e camada física, que de um modo geral não são suficientemente apurados para descrever o comportamento real da rede, a credibilidade dos resultados nem sempre é a desejada. No entanto, modelos muito detalhados conduzem a problemas de escalabilidade e performance da rede, pelo que o compromisso entre precisão e escalabilidade será sempre um dos problemas a resolver [37][38].

A presente secção introduz brevemente os conceitos de simuladores e emuladores, as suas diferenças (Tabela 4), e apresenta uma lista de *softwares*, a partir da qual se seleccionaram os candidatos para a análise à rede implementada no Museu da Baleia.

Simuladores	Emuladores
Sistema capaz de imitar parcialmente outro sistema.	Sistema que imita de forma exata o comportamento de outro sistema.
Pode não seguir todas as regras inerentes do sistema que está sendo simulado	Obedece rigorosamente aos parâmetros e regras do sistema emulado.
Modela aplicações e eventos.	Copia o comportamento do sistema.

Tabela 4: Diferenças entre simuladores e emuladores.

Trabalhos semelhantes podem ser encontrados em [39][40]

2.5.1 Simuladores

Os simuladores são adequados a situações onde se pretende averiguar a rede num nível mais alto, como o efeito do encaminhamento, dos protocolos, topologia ou da agregação de dados. Uma das desvantagens dos simuladores é que o utilizador fica limitado a uma única plataforma, de software ou hardware, e uma linguagem de programação, tipicamente Mica Motes e o TinyOS/NesC. O facto de estes estarem-se a tornar os *standards* das redes de sensores sem fios ajuda no entanto a mitigar este problema [41][38].

- Simuladores

NS-2 é um simulador de eventos discretos orientado a objetos, escrito em C++, com um interpretador OTcl como *front-end*. Utiliza scripts em OTcl para

a configuração do simulador, definição da topologia da rede, especificação de cenários, gravação de simulações, entre outras funcionalidades. O facto de apresentar uma abordagem modular permite que seja facilmente extensível e popular. A simulação ocorre ao nível dos pacotes pelo que, permite a obtenção de resultados detalhados. Como principais desvantagens da utilização deste simulador identifica-se a inexistência de um *sensing model*, os formatos dos pacotes e dos protocolos MAC são diferentes dos utilizados tipicamente nas redes de sensores sem fios e o modelo de energia utilizado é bastante simples [42]. O facto de ser orientado a objetos, embora traga vantagens ao nível da extensibilidade e organização, faz com que não tenha uma boa performance em redes com grande número de nós. Como cada nó é um objeto e pode interagir com todos os outros nós da rede, implica que um elevado número de dependências seja verificado a cada passo da simulação, criando uma relação n^2 [38]. Os simuladores seguintes são extensões do NS-2.

J-Sim é um ambiente de simulação baseado no NS-2, sendo desenvolvido em Java. Ao abandonar os conceitos de orientação a objetos em prol de uma abordagem por componentes, resolve o problema da escalabilidade do NS-2. Melhora também o modelo de energia e a capacidade de simular a utilização de sensores para captar os fenómenos em causa. Permite ainda simular aplicações e suporta ligações entre o hardware dos sensores reais ao simulador[38]. A biblioteca gráfica utilizada é o Jacl. Uma das principais vantagens é o facto de suportar uma grande variedade de protocolos, incluindo também uma *framework* com um modelo de WSN bastante detalhado e implementação de algoritmos de localização, encaminhamento e difusão de dados. No entanto, e como muitos outros construídos sobre simuladores universais, o J-Sim apenas disponibiliza o 802.11 ao nível dos protocolos MAC. Apresenta também algumas ineficiências, como por exemplo um *overhead* no modelo de inter-comunicação e pela utilização do Java, quando comparado a muitas outras linguagens como o C++[38].

OMNeT++ é um simulador modular de eventos discretos implementado em C++, simples e de aprendizagem rápida. Apresenta uma biblioteca gráfica rica para animação, *tracking* e análise de erros. Uma das desvantagens deste simulador é o facto de disponibilizar poucos protocolos, embora a sua recente popularidade e contributo da comunidade académica tenham mitigado este problema. No entanto, o facto que de a maioria dos módulos disponíveis terem sido desenvolvidos por grupos de pesquisa independentes, tornam-nos incompatíveis visto que não partilham uma interface comum[41].

Castalia, desenvolvida para redes de dispositivos de baixa potência como nós sensores sem fios. As suas principais vantagens (declaradas) são os modelos avançados utilizados para a simulação do meio, baseados em dados medidos empiricamente, e modelos de rádio criados com base em rádios reais de comunicação com baixa potência e monitorização do consumo de energia [43].

O ns-3 é utilizado para modelar a evolução de um sistema em rede através de eventos discretos no tempo. O ns-3 é um projeto com licença GNU GPLv2, escrito em C ++, com ligações (do inglês bindings) disponíveis para Python, usando um programa chamado PyBindGen para gerar ligações Python para todas as bibliotecas. Útil para o estudo do desempenho da rede ou na operação de protocolos. Com a capacidade de criar código em C ++ ou Python é possível criar os módulos específicos, existindo já muitos disponíveis visto que se trata de um projeto muito ativo, não sendo no entanto compatível com ns-2⁷.

MiXiM é outra framework de modelação e oferece modelos detalhados de propagação de ondas rádio, estimação da interferência, consumo de energia nos transceivers rádio e protocolos MAC para redes sem fios [44].

2.5.2 Emuladores

A ferramenta, que utiliza *firmware* bem como *hardware* para executar uma simulação, é chamada de emulador, [45]. A emulação pode combinar a implementação de software e hardware em nós reais, podendo por isso representar um desempenho mais preciso. Normalmente, o emulador tem alta escalabilidade, pelo que pode emular vários nós sensores ao mesmo tempo. Se pretendermos uma análise mais específica, como a interação entre nós ou aperfeiçoar o desempenho da rede ou dos seus algoritmos, devemos optar por emuladores visto que, estes correm o próprio código das aplicações.

- Emuladores

TOSSIM é um emulador de eventos discretos, extensível, desenhado de raiz, cujo objetivo é estudar o comportamento de aplicações desenvolvidas sobre o sistema operativo TinyOS com o suporte a nós "MicaZ Motes", a emulação de interfaces rádio e os conversores analógico para digital e EEPROM. Foi criado sobre quatro conceitos: a escalabilidade, o sistema tem de suportar milhares de nós com configurações diferentes, a completude, para capturar o comportamento da rede é necessário simular o máximo de interações de sistema, fidelidade, para que os testes sejam o mais apurados possível, até as interações mais subtis devem ser capturadas, e a validação dos algoritmos [46][47].

O TOSSIM suporta C++ e Python e tira vantagem da arquitetura baseada em componentes do TinyOS, sendo que para o compilar não são necessárias modificações ao código fonte. Assim, as aplicações podem correr em PCs, facilitando o recurso a outras ferramentas de análise, permitindo testes não só aos algoritmos como também às suas implementações. Posteriormente o código pode ser introduzido nos nós sem qualquer alteração. Para representar o ambiente recorre a um modelo probabilístico de erros. Utiliza uma ferramenta de interface gráfica chamada TinyViz para visualizar, monitorizar, controlar e analisar simulações.

⁷<https://www.nsnam.org/docs/ns-3-overview.pdf>

Outra ferramenta, SimX, permite controlar a velocidade da simulação, a manipulação da topologia e o controlo das variáveis. Quando perante restrições importantes ao nível do consumo de energia, é possível recorrer à extensão PowerTOSSIM.

Entre as principais desvantagens referidas pela comunidade aponta-se os factos de que não é apropriado para o controlo de métricas referentes a protocolos, não modela o consumo de energia (embora como referido se possa recorrer ao PowerTOSSIM) e que todos os nós têm de correr o mesmo código, pelo que não é apropriado para redes heterogéneas [48].

ATEMU é uma plataforma de emulação para redes de sensores, utilizando a linguagem de programação C, distinguindo-se por ter a capacidade de emular hardware de nós sensores a um nível extremamente baixo. Suporta nós MICA2 mas pode ser facilmente extensível a outros tipo de hardware, permitindo por exemplo testar sistemas operativos e aplicações sem ser o TinyOS e, simular redes heterogéneas com diferentes sensores, tendo como desvantagens uma fraca escalabilidade e elevados requisitos de processamento. Utiliza uma interface gráfica, XATDB, capaz de depurar e observar a execução do código. ATEMU é adequado para cenários que necessitem de um alto nível de fidelidade, mesmo que para tal se recorra a simulações mais longas, apresentando no entanto resultados mais refinados que o TOSSIM. Como desvantagens apresenta, um *overhead* envolvido na descodificação instrução a instrução e, como o modelo do rádio é simulado, cada transmissão de rádio afeta todos os outros nós da rede, criando assim um algoritmo de ordem n^2 . Apesar de ser 30 vezes mais lento que o TOSSIM e não ser escalável, é o emulador para redes de sensores mais preciso ao nível da instrução. Informação mais detalhada pode ser encontrada em [49].

Avrora⁸ é um conjunto de ferramentas de análise e simulação para programas feitos para micro controladores AVR, suportando plataformas de sensores como os Mica2 e MicaZ. Sendo implementado em Java, oferece grande portabilidade e flexibilidade visto que a simulação do código máquina é independente do sistema operativo. Cada nó é simulado pela sua tarefa (*thread*), correndo o código real do nó Mica. O Avrora é o meio termo entre o TOSSIM e o Avrora pois, embora opere o código instrução a instrução, não realiza uma sincronização no final de cada, alcançando uma maior escalabilidade, até aos 10000 nós, sem no entanto perder precisão [45][50].

A principal desvantagem deste emulador prende-se com o facto de não ter interface gráfica, sendo a interação feita por linha de comandos. Não modela *clock drift* (fenómeno no qual os nós ficam com frequências de relógio ligeiramente diferentes devido a tolerâncias de fabrico, temperatura e performance da bateria), nem mobilidade.

⁸<https://sourceforge.net/projects/avrora/>

2.5.3 Conclusão

Com base neste levantamento sobre simuladores e emuladores e nas características observadas referentes ao cenário em causa (Museu da Baleia), foram escolhidos os simuladores Castalia, NS-3 e o MiXiM. No capítulo seguinte expande-se o cenário de implantação e escolhe-se, entre estes, o simulador que serve de base ao estudo da rede implementada com vista à obtenção de orientações relativas à topologia e dimensionamento da rede do Museu da Baleia.

3 Museu e contextualização das simulações

O Museu da Baleia e a implementação do projeto WISE-MUSE são o objeto de estudo da presente dissertação. Para alcançar os objetivos definidos, a criação de orientações pertinentes à execução e gestão de projetos baseados em WSN *indoor*, optou-se por recorrer a simulações em detrimento de estudos analíticos ou da análise de dados reais. A flexibilidade obtida através da utilização de um simulador e de um modelo, criado com o conhecimento prévio do hardware utilizado (fidedigno), tornaram esta opção a mais viável e adequada.

O museu e o projeto são descritos na primeira secção do presente capítulo e servem de introdução e contextualização ao problema em mãos. A segunda secção aborda as condicionantes apresentadas pelo museu, tanto ao nível do *hardware* utilizado como da solução utilizada no projeto. Na terceira procede-se à análise e escolha do simulador, ponderando para tal, as características do *hardware*, a disposição espacial da rede e as condições inerentes aos ambientes *indoor*.

3.1 Museu e o projeto WISE-MUSE

O Museu da Baleia, situado na vila do Caniçal no arquipélago da Madeira, é o objeto de estudo do presente trabalho. Este visa preservar o património e o conhecimento histórico sobre a caça à baleia (baleação) na Madeira, e gerar e divulgar o conhecimento sobre os cetáceos e o meio marinho ⁹. A baleação no arquipélago da Madeira, atingiu o seu auge nos anos 50, vindo a cessar de forma voluntária, em 1981. As instalações atuais do Museu, fundadas em 2011, são compostas por uma variedade de espaços que vão desde amplos pavilhões, onde se podem visitar diversas exposições (permanentes e temporárias), baleeiras, réplicas à escala real de baleias e golfinhos, até salas de conferência e bibliotecas que servem de suporte, não só às ações de sensibilização ambiental promovidas pelo Museu, como também aos projetos científicos em que este participa.

Dadas as características do museu, surge a necessidade natural de conservar e monitorizar, tanto os artefatos expostos como as próprias instalações, e ainda garantir a sua segurança. A resolução destas necessidades alinha-se na perfeição com um outro projeto. Este consiste na monitorização de parâmetros, ambientais e estruturais, com o recurso a redes de sensores sem fios, de nome WISE-MUSE. Este, desenvolvido inicialmente para o museu de arte contemporânea localizado na Fortaleza de São Tiago, recorre às WSN para monitorizar e controlar automaticamente parâmetros ambientais, como o grau de humidade, os níveis de CO e CO₂, luminosidade e temperaturas, e até mesmo o estado das portas de emergência (aberto, fechado e emergência) [51] [52].

Os objetivos deste projeto são alcançados através de:

- a criação de redes de sensores sem fios capazes de monitorizar os parâmetros ambientais mais críticos nos museus (incluindo os poluentes);

⁹<http://www.museudabaleia.org>

- proporcionar aos visitantes e aos próprios gestores dos museus a possibilidade de consultar dados em tempo real, em diferentes formatos (gráficos, tabelas, gradientes) e a consulta de históricos de dados quando necessário;
- capacidade de análise dos dados de modo a estar em conformidade com as regras de preservação a implementar. Se houver uma variação significativa, ou um certo valor atingir um determinado limite, um alerta tem de ser enviado automaticamente, seja por mensagem para um telemóvel (SMS) ou *e-mail*, de modo a aumentar a eficiência do sistema de monitorização.
- controlo automático das condições ambientais manipulando os sistemas de ar condicionado e do grau de humidade do museu;
- visualização dos dados e dos componentes do WSN integrados num modelo 3D do edifício;
- instalação de um sistema de video vigilância wireless com a capacidade de envio de mensagens em casos de intrusão, aumentando a segurança do museu;
- monitorizar remotamente os estado de abertura das portas de emergência;
- solução de baixo custo e não intrusiva.

3.2 Contextualização do modelo

A implementação do projeto WISE-MUSE ao Museu da Baleia, apresenta características resultantes dos problemas clássicos deste tipo de cenário e outras específicas ao próprio museu. Algumas destas definem não só a topologia do próprio projeto como também se refletem na formulação do modelo a utilizar no capítulo 4. De entre estas, consideraram-se relevantes as seguintes:

- A colocação dos nós sensores a 2m de altura para evitar danos acidentais provocados por visitantes, embora o ideal para CO seja colocar ao nível dos joelhos e do CO₂ junto ao teto.
- Necessidade de controlar a perda de pacotes nas transmissões devido às interferências provocadas por outros sistemas já existentes (audiovisuais ou mesmo equipamentos tipo microondas)
- Condicionamento da posição de alguns nós sensores para zonas com tomadas elétricas próximas, dada a opção do Museu pela alimentação direta dos nós a partir da sua rede elétrica.
- A estação base, que tem de estar ligada ao servidor, foi colocada numa pequena divisão sob umas escadas e coberta por um acabamento em madeira, o que condiciona receção do sinal.

- Os sensores que detetam as mudanças no estado das portas de emergência (aberto, fechado, emergência) estão localizados dentro dos próprios bloqueadores das portas. Como estes são alimentados por corrente elétrica, condicionam, por causa da voltagem, a escolha das tecnologias de transmissão de sinal a utilizar.
- Dada a natureza das portas de emergência (acessos para o exterior), estas situam-se nas zonas periféricas do Museu, que, de uma forma geral, são os pontos mais distantes em relação à estação base, situada numa zona mais central. O hardware utilizado para os sensores nestas portas, XBee Series 2, têm um alcance de transmissão inferior (que os XBee PRO, utilizados nas restantes situações), o que condicionou a localização dos nós em redor destas, por forma a manter a conectividade com o resto da rede.
- Existe incompatibilidade entre o *hardware* do XBee PRO e a sua colocação em portas de emergência.

Assim, da implementação do projeto, aliada à necessidade de minimizar o seu impacto visual, de controlar os efeitos das interferências e de utilizar de forma condicionada a localização de alguns nós e do seu hardware, resultou uma rede topológica densa e com elevado grau de redundância. É sobre esta rede, que após a seleção de um simulador adequado e da criação de um modelo, se procederá ao estudo da sua topologia, dimensionamento e comportamento sobre diferentes condições de carga.

3.2.1 Avaliação e seleção do simulador

O campo da simulação de redes apresenta, como mencionado em 2.5, uma grande variedade de soluções, cada qual com as suas especificidades e limitações.

No presente contexto, uma grande condicionante para a escolha do simulador adequado, é a sua capacidade para simular redes de sensores sem fios em ambientes *indoor*. Nestes, os problemas genéricos das WSN são exacerbados, verificando-se por exemplo que, a existência de obstáculos físicos como paredes, mobiliário ou pessoas, é mais comum e variável. Outros desafios a ultrapassar são a perda de potência de sinal devido a fenómenos como a reflexão, penteração, difração e dispersão, assim como a existência de interferências causadas por outros dispositivos operando na mesma largura de banda, sendo um requisito para o simulador a capacidade de modelar estes parâmetros.

Por outro lado, a rede já implementada no Museu utiliza tecnologia ZigBee, pelo que a *stack* protocolar, nas camadas física e de acesso ao meio, se baseia na norma 802.15.4, condicionando por sua vez a seleção do simulador pois este tem de suportar esta norma. A condicionante final neste processo foi ser *open source*. Este facto permite, não só a livre utilização do *software*, como também promove a participação da comunidade interessada e, por consequência, o melhoramento do próprio simulador.

De referir, outro factor que é normalmente importante neste tipo de simulação, a capacidade de modelar o comportamento da bateria. Dado que o Museu alimenta

a sua rede de nós sensores diretamente da sua rede elétrica, esse problema torna-se irrelevante para esta escolha.

Da triagem feita aos simuladores identificados em 2.5, usando como principais fatores de seleção os critérios mencionados, *open source*, poderem simular cenários *indoor* e suportarem a norma IEEE 802.15.4, foram selecionados três, o Castalia ¹⁰, o NS-3 ¹¹ e o MiXiM ¹², sendo que tanto o Castalia como o MiXiM são *frameworks* do OMNeT++ ¹³. Estes foram instalados e testados, apresentando-se de seguida as ilações tiradas dessas experiências.

Castalia

O Castalia foi o primeiro simulador a ser testado e por isso teve a curva de aprendizagem mais lenta. Foram seguidos os exemplos providenciados pelo *software* e procedeu-se à expansão desses ao cenário do Museu. Mesmo não utilizando qualquer interface gráfica, a sua utilização acabou por revelar-se simples, sendo necessário modificar apenas o ficheiro de configuração.

As simulações efetuadas limitaram-se à utilização de protocolos e *hardware* cujas implementações eram providenciados com o simulador, que, de uma forma geral se mostraram suficientes e bem organizadas (estrutura de diretórios).

Como aspetos menos positivos, destaca-se a incapacidade de simular redes com *multi-hop*, o que torna a utilização do Castalia inadequada ao cenário do museu. A documentação, além do manual de utilizador, pareceu limitada, tanto na página oficial como ao nível de projetos efetuados por terceiros com o simulador. Este último aspeto é no entanto ligeiramente mitigado pela comunidade Google Groups, realçando-se que o próprio líder do projeto demonstrou interação na resolução de dúvidas, tendo respondido pessoalmente nos fóruns, nomeadamente à questão do *multi-hop*.

Como resultado destas simulações, afere-se que o Castalia está mais orientado para o estudo de transmissões nó para nó, e não tanto para o estudo do comportamento de redes mais complexas.

MiXiM

O MiXiM funciona com um GUI baseado no Eclipse ¹⁴ e a foi a única plataforma gráfica testada. Dada a experiência prévia com o Castalia, a curva de aprendizagem foi mais rápida visto que as duas *frameworks* utilizam o OMNeT++, partilhando assim a mesma estrutura do ficheiro de configuração.

O simulador apresenta várias ferramentas gráficas como o *Simulation Launcher*, *Sequence Chart View* e o *Result Analysis Tool*. O primeiro permite observar a simulação

¹⁰<http://castalia.npc.nicta.com.au/>

¹¹<https://www.nsnam.org>

¹²<http://mixim.sourceforge.net/>

¹³<http://www.omnetpp.org/>

¹⁴<http://mixim.sourceforge.net>

em tempo simulado até ao nível do pacote mas não escala bem com o aumento do número destes, especialmente quando são *broadcasts*, nem com o do número de nós. É no entanto uma ferramenta bastante útil em cenários pequenos. O *Sequence Chart View* permite depurar sequencialmente o fluxo das transmissões e o *Result Analysis Tool* permite averiguar os indicadores disponibilizados e criar gráficos com estes. Todas estas ferramentas tornam os processos de aprendizagem e depuração mais acessíveis, e a boa seleção de exemplos disponibilizados, torna fácil a adaptação destes para o cenário pretendido (museu).

Os protocolos apresentados são aplicáveis a uma variedade assinalável de campos aplicativos, tendo-se complementado com outra *framework*, INET ¹⁵, de modo a incluir os necessários para o presente estudo.

As maiores dificuldades registadas com este simulador foram a necessidade de entender com algum detalhe o modelo de comunicação utilizado (*Connection Manager*, *Decider* e modelos analógicos, ver o apêndice D) e a interpretação de alguns resultados obtidos, que aparentemente se contradizem (ver capítulo 4). Outras pequenas falhas resultaram de uma utilização mais exaustiva do simulador, tais como a incapacidade de simular a falha de um nó, ou simplesmente desligá-lo, e alguns *crashes* utilizando *Flooding*.

NS-3

Não partilhando a *framework* OMNeT++, a experiência NS-3 foi muito diferente.

A definição da topologia e a gestão das configurações para as simulações são efetuadas de modo mais explícito (quando comparado com os ficheiros de configuração), recorrendo-se para tal a instruções em C++. Como por exemplo, é necessário definir explicitamente a *stack* protocolar a utilizar ou mesmo definir que atributos da simulação pretendemos obter resultados, o que, no caso das *frameworks* OMNeT++, é feito de forma transparente e automática.

Tal como com o Castalia, foi apenas utilizada a linha de comandos embora estejam disponíveis interfaces gráficas, como por exemplo o NetAnim. A documentação disponível *online* está bem estruturada e com bom detalhe, tendo sido efetuada com o recurso a uma ferramenta chamada Doxygen.

Para corresponder às condições impostas pelo cenário em questão, utilizou-se uma *framework* mais adequada, 6LowPAN, mas verificou-se que, à data das simulações, o suporte à norma IEEE 802.15.4 não era o adequado, existindo *bugs* ao nível das funções *helper*. Descobriu-se também que a *framework* apenas permitia encaminhamento estático, o que, tendo em conta a natureza do problema, tornou inadequada a utilização deste simulador.

¹⁵<https://inet.omnetpp.org>

Conclusão

No âmbito do presente trabalho, e de acordo com a experimentação efetuada com os 3 simuladores, foi escolhido o MiXiM. Embora haja a necessidade de dominar tanto o OM-NeT++ como o próprio MiXiM, as ferramentas e opções disponibilizadas adequam-se às características do nosso cenário (*indoor*) e aos objetivos definidos para as simulações (orientações para situações específicas). Por outro lado, a ausência de *multi-hop* (Castalia) e o suporte inadequado à norma 802.15.4 (NS-3) revelaram-se condicionantes importantes para a escolha do simulador. De realçar ainda que a comunidade Google Groups referente aos simuladores e a informação prestada por estes, se revelou bastante útil no processo de aprendizagem.

Caraterística	Castalia	MiXiM
<i>Log Normal Shadowing</i>	Sim	Sim (usando a fórmula de Friis)
Extensibilidade do modelo do canal sem fios	É necessário alterar o código fonte	É possível criar externamente
Modelação do ruído	Sim	Sim
Configurações BER e PER	Técnicas de modelação e SNR	Técnicas de modelação e SNR
Documentação	Excelente	Fraca
Processo de aprendizagem	Rápida	Lenta
Exemplos facultados	Suficientes	Suficientes
MAC 802.1.4	Parcial	Sim
Suporte CC2420	Sim	Sim
Suporte a energia	Sim	Sim
Extensibilidade de funções internas	Boa	Boa
Tempo relativo consumido	Rápido	Lento

Tabela 5: Comparação entre Castalia e MiXiM.

Trabalho similar no que respeita à avaliação de simuladores, embora mais pormenorizada, foi feita em [19], assinalando a tabela 5 as conclusões do autor. Em relação às simulações efetuadas, considera-se que a documentação atual não é escassa, nem que o processo de aprendizagem não é tão lento como referido (ultrapassar dúvidas). Por outro lado, a referência aos tempos lentos para as simulações no MiXiM parece adequada visto que, algumas configurações utilizadas, produziram simulações muito demoradas.

3.2.2 Hardware

A rede de sensores sem fios do museu é composta por dois tipos de dispositivos, o XBee S2 e o XBee PRO (S2B), num total de 84 nós e uma estação base. As suas principais características, de acordo com o presente contexto, são as apresentadas na tabela 6.

	XBee S2	XBee PRO
Transmit Power	2mW	63mW
Range (Indoor/Outdoor)	40m/120m	90m/1500m
Receiver Sensitivity	-96dBm	-102dBm
Max. Data Rate	250kbps	
Frequency Band	2,4GHz	

Tabela 6: Principais características do hardware utilizado

Uma comparação técnica entre estes dois tipos de hardware pode ser consultada no Apêndice A, primeira e terceira colunas respetivamente.

A Figura 4, mostra a distribuição espacial destes dispositivos de acordo com as suas funcionalidades. Os nós instalados nas portas, para o controlo dos bloqueadores destas, são do tipo XBee S2. Estes estão representados por ícones com um símbolo de uma porta, estando neste caso 5 sensores a reportar portas fechadas, a vermelho, e um a verde, porta aberta. De realçar que, esta informação é reportada para a estação base a cada 90 segundos. Todos os restantes nós são XBee PRO. Estes possuem sensores que desempenham uma variedade de funções, desde o controlo da ventilação de gases em posições estratégicas (nomeadamente dispositivos de corta-fogo a disparar em casos de incêndio), ícones com o símbolo de uma chama, a amarelo na planta em questão, ao controlo da luminosidade em determinados expositores, até aos já referidos parâmetros atmosféricos, ícones com um símbolo de três ondas, a verde na mesma planta. As arcas frigoríficas encontram-se no piso -2 e estão representadas por ícones com um símbolo semelhante a um floco de neve a azul (consultar restantes plantas do Apêndice B). Todos estes nós enviam mensagens de controle com uma periodicidade de 600 segundos (10 minutos).

De referir ainda que todos sensores, independentemente do tipo de dispositivo, estão preparados para, no caso de eventos excecionais, enviar assincronamente mensagens de alerta para a estação base. Estes acontecem sempre que certos limites pré-estabelecidos, de acordo com o tipo de sensor e com a posição do nó sejam ultrapassados, como por exemplo, se os sensores instalados nos frigoríficos detetarem uma temperatura acima de um determinado limite.

3.2.3 Disposição espacial

De acordo com o levantamento anterior, o modelo foi criado e ajustado de modo a servir os propósitos deste trabalho, nomeadamente complementar analiticamente a rede de

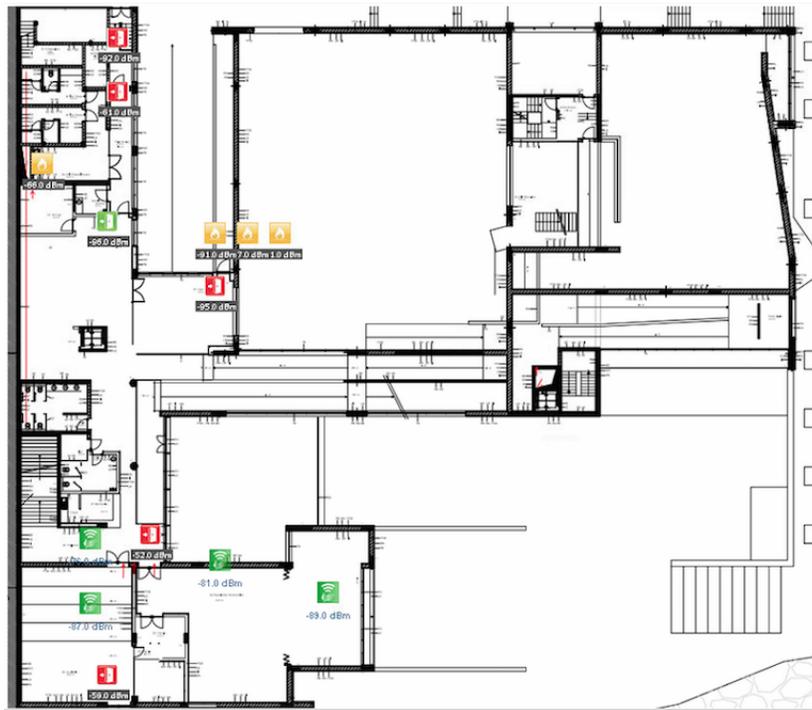


Figura 4: Planta do piso 1

sensores sem fios implementada no Museu da Baleia.

O primeiro passo deste ajustamento consistiu na criação da topologia física da rede, sendo a sua disposição espacial obtida de acordo com o material fornecido pelo grupo responsável pela sua implantação. Este material consistia em mapas no formato .pdf com os respetivos nós, diferenciados por cores de acordo com as suas funcionalidades (conforme descrito em 3.2.2), e as plantas do museu no formato .dwg, utilizado pelo conhecido AutoCad, da AutoDesk¹⁶. Com estes recursos foi possível obter um mapeamento 3D dos nós pertencentes à rede, posteriormente utilizado como base das simulações. A Figura 5 demonstra este processo, estando as plantas relacionadas disponíveis no Apêndice B. Como pequena simplificação considerou-se que, todos os nós num mesmo piso, tinham a mesma cota, isto é, a mesma coordenada Z (num eixo ordenado x, y, z), sendo esta obtida de acordo com os valores apresentados nas plantas. Exceção a esta consideração foram alguns nós que, situados no pavilhão principal se encontravam colocados no teto.

Assim, os nós, 85 na sua totalidade, distribuídos numa área de dimensões aproximada de 65x60x25 metros, foram inseridos individualmente no ficheiro de configuração base, de acordo as instruções da Listagem 1.

```
**playgroundSizeX = 65m
**playgroundSizeY = 60m
```

¹⁶<http://www.autodesk.com/education/free-software/all>

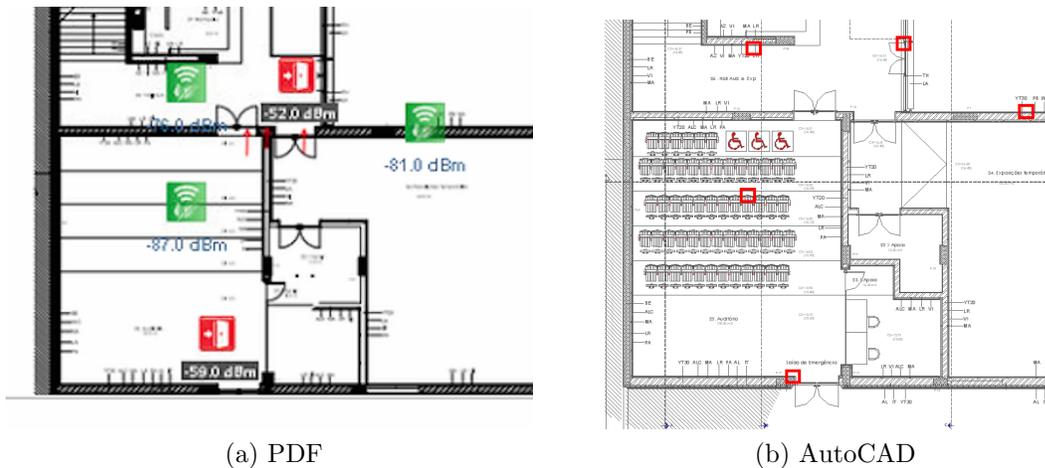


Figura 5: Mapeamento de acordo com o material fornecido

```

**.playgroundSizeZ = 25m
**.numHosts = 85

**.node[0].mobility.initialX = 0m
**.node[0].mobility.initialY = 43m
**.node[0].mobility.initialZ = 21m

```

Listing 1: Disposição espacial

A localização de todos os nós referentes à rede, pode ser consultada na transcrição do referido ficheiro, "template.ini" (ponto de partida de todas as simulações executadas), no Apêndice C.

O segundo passo, consistiu no ajustamento do modelo no que respeita às condições de transmissão e receção de mensagens entre os nós componentes da rede.

3.2.4 Condições de transmissão

Pelo facto de estarmos perante um ambiente indoor realizou-se uma análise espacial ao museu de modo a inferir o coeficiente para a perda de sinal adequado. Como referido anteriormente, o museu é composto por zonas díspares como os escritórios, caracterizados por uma grande densidade de paredes, até pavilhões de exposições e parques de estacionamento, onde prevalecem os espaços amplos. Foi então realizada uma classificação considerando-se zonas de baixa, média e elevada densidade de divisórias e obstruções. A medição destas áreas realizou-se com o recurso à ferramenta AutoCAD, sendo posteriormente feita uma ponderação das mesmas. Os resultados assim obtidos apresentam-se na tabela 7. Horizontalmente pode-se avaliar primeiro as áreas medidas por cada andar de acordo com as densidades definidas e, nas últimas linhas, as áreas totais e a sua ponderação percentual.

Piso	Baixa	Média	Alta	Total por piso [m ²]
2	984	743	672	2399
1	1939	449	109	2497
0	1758	418	221	2397
-1	322	175	386	883
Total por densidade [m ²]	5003	1785	1388	8176
Total por percentagem [%]	61,2	21,8	17	-

Tabela 7: Análise da densidade do museu

De acordo com estes valores, aproximadamente 61,2 % da área do museu possui baixa densidade de divisórias. Acrescenta-se ainda que, embora tenhamos 4 pisos, para efeitos de transmissão apenas temos 3, pois existem várias zonas em que o pé direito é duplo ou triplo. Para esta escolha (coeficiente de perda de sinal, α), foi também tomada em consideração a tabela 8, cujos valores são adotados por grande parte da literatura da área [53].

Posição	n	X _s [dB]
mesmo piso	2.76	12.9
através de um piso	4.19	5.1
através de dois pisos	5.04	6.5
através de três pisos	5.22	6.7

Tabela 8: Valores para o expoente de perda de sinal [53]

O valor final para α escolhido foi de 5.04, que de acordo com a mesma tabela, representa ambientes com transmissões através de dois pisos. Dada a baixa densidade de divisórias e a existência de espaços amplos, tal aproximação parece adequada, proporcionando de acordo com a fórmula da distância de transmissão, Fórmula 2 ($pMax$ é a potência máxima de transmissão na rede, α é o coeficiente de perda de sinal e $minRP$ é a potência mínima de receção), utilizada pelo simulador para o cálculo das distâncias de interferência de um sinal nos nós vizinhos, nos valores de 14,78m/38,57m, respetivamente para o modelo XBee S2 e XBee PRO.

$$Distância = \left(\frac{wavelength^2 \times pMax}{16 \times \pi^2 \times minRP} \right)^{\frac{1}{\alpha}} \quad (2)$$

3.2.5 Outras considerações

Outras considerações mais genéricas tomadas neste levantamento foram a fixação dos nós, assegurada pela instrução presente na Listagem 2 e a taxa de envio de *bits*,

optando-se por utilizar a máxima permitida pela tecnologia usada, 250kbps. No entanto a proposta pela *framework*, 15360bps, é um valor algo interessante pois seria um valor passível de ser utilizado neste tipo de rede, visto que não existem considerações muito restritas na utilização da bateria e, no cenário do Museu, estamos interessados na transmissão esporádica de valores. No entanto, a capacidade de atuar com taxas mais elevadas, reflete-se em períodos de transmissão e receção de pacotes mais curta, o que por sua vez se reflete em períodos de antena ativos mais curtos, logo em poupanças na bateria, tornando estas taxas desejáveis na maioria dos cenários.

```
**node[*].mobilityType = "StationaryMobility"
```

Listing 2: Mobilidade

```
**node[*].nic.mac.bitrate = 250000bps
```

Listing 3: *Bitrate*

3.3 Conclusão

Todas estas considerações tiveram como propósito aproximar o nosso modelo à rede real implementada no museu no âmbito do projeto WISE-MUSE. Teve-se sempre em atenção que, quanto maior é o desejo de recriar as condições reais, maior é a dificuldade e complexidade impostas sobre o modelo, pelo que o detalhe obtido nesta secção resulta de um compromisso e ajusta-se aos objetivos definidos. O resultado final desta fase é o ficheiro de configuração denominado "template.ini", podendo este ser consultado no Apêndice C. Este ficheiro serve de ponto de partida a todas os cenários desenvolvidos no capítulo 4.

4 Análise à simulação e resultados

O primeiro passo na execução de uma simulação é a descoberta da rede. Para o selecionador selecionado, o processo baseia-se na utilização do protocolo *Wiseroute*. A análise do seu desempenho permitiu aferir sobre aspetos importantes como a conectividade da rede, a qualidade da cobertura das zonas e a posição ideal para a localização da estação base.

O *Wiseroute* é então um algoritmo que constrói a partir de um nó central (estação base), uma tabela de roteamento garantindo a inexistência de *loops*. Para tal, a estação inicia o processo emitindo uma mensagem para a criação das rotas em forma de *broadcast*. Cada nó que recebe esta mensagem, verifica o valor *Received Signal Strength Indicator*, RSSI, do pacote recebido e compara-o com um limite pré estabelecido para a rede. Se for superior, o nó marca esta fonte como sendo o nó destino, usando-o como próximo salto no envio de pacotes para a estação. Todos os outros *broadcasts* duplicados recebidos são automaticamente descartados. Cada nó que recebe esta mensagem reenvia-a, de modo a continuar o processo. Este procedimento maximiza a probabilidade de todos os nós se juntarem à rede e de evitar *loops*.

A utilização deste protocolo aliado a um conjunto de indicadores providenciados pelo OMNeT++, como por exemplo *nbRoutesRecorded*, *nbDuplicateFloodsReceived*, *nbDataPacketsReceived* e *latency:mean*, foram as principais fontes de dados e providenciaram o material necessário para alcançar os objetivos deste trabalho.

As secções do presente capítulo abordam as simulações efetuadas de modo a alcançar os objetivos propostos e estão organizadas da seguinte forma. A primeira secção aborda um estudo que consistiu na determinação de meta características referentes às simulações. A definição destas características, período de transição, período de estabilização e o número de repetições das simulações, foram necessárias de modo a otimizar os recursos disponíveis e a qualidade dos resultados, tendo sido utilizadas nas restantes simulações.

A secção seguinte estuda a WSN de acordo com a sua conectividade e apresenta um estudo sobre os modelos analógicos utilizados, determinando-se qual seria a posição mais adequada para a estação base, tendo como principal indicador a conectividade global da rede.

A terceira secção, analisa o sobredimensionamento, visando a manipulação do número de nós da rede com vista à exploração do grau de sobredimensionamento, abordando novamente questões relacionadas com a conectividade e qualidade da cobertura prestada pela rede.

A secção final reúne as conclusões obtidas ao longo das secções, resumindo-as.

4.1 Estudo prévio

No mundo das simulações, os resultados podem ser facilmente enviesados originando conclusões precipitadas, incoerentes ou mesmo simplesmente erradas. Uma das ob-

servações iniciais feitas durante o processo de aprendizagem do OMNeT++ foi que, um só ciclo de *broadcasts* do protocolo *WiseRoute* podia não ser suficiente para que todos os nós da rede fossem descobertos. Sendo estes ciclos repetidos de 1200 em 1200 segundos (valor *default*), qualquer troca de pacotes nos instantes iniciais da simulação não seria feita na presença de todos os nós da rede, produzindo deste modo dados sem interesse no contexto da maioria das simulações. Por outro lado observou-se também que, tornar as simulações longas poderia conduzir a um desperdício de recursos, especialmente de tempo, sem que isso se traduzisse numa melhoria dos resultados obtidos.

De modo a evitar este tipo de inconsistências, é necessário um espírito crítico e imparcial, como tal, procedeu-se aos estudos descritos de seguida.

4.1.1 Período de transição

No presente contexto, foi definido como período de transição o número de ciclos *Wiseroute* necessários para que a rede atingisse conetividade total, ou seja, que todos os nós da rede fossem descobertos, e criassem uma rota para a estação base. O objetivo da determinação deste valor era o de suprimir o envio de quaisquer pacotes pertinentes à simulação antes do término desse período, garantindo deste modo que todos os nós da rede participariam na simulação.

Inicialmente a simulação foi executada com um tempo equivalente a 5 ciclos *Wise-route*. A análise do parâmetro *nbRoutesRecorded*¹⁷ permitiu-nos depois saber se, todos os nós eram descobertos e possuíam uma rota definida para a estação. Se todos os nós fossem descobertos, a simulação era repetida nas mesmas condições, mas agora num período de apenas 1 ciclo. Caso contrário, o número de ciclos era aumentado para 10 de modo a atingir uma taxa de descoberta de 100%. Os resultados obtidos inicialmente mostraram uma conetividade total ao fim de 5 ciclos, todavia, para um número inferior de ciclos realizada tal conetividade não se verificou, pelo que foi definido o período de transição para 5 ciclos.

Explorando mais o cenário em causa, foram escolhidos diversos nós tendo em consideração a sua distribuição espacial ao longo dos 4 pisos do museu. Esta escolha foi feita de modo a obter amostras sobre comportamentos possíveis do protocolo *Wise-route* na rede. Uma das observações feitas foi que, dependendo da posição escolhida para a estação base, o processo de descoberta da rede conseguia, em certos casos, definir as rotas para todos os nós com cinco iterações apenas do *WiseRoute*, enquanto que noutros a topologia completa não é criada por mais iterações que o processo realizasse. Este cenário é novamente abordado na secção seguinte podendo os resultados destas simulações ser consultados na tabela 9. Esta tabela apresenta, para cada nó testado como estação base, se a descoberta da rede era total (ok) e quantos ciclos eram necessários (1, 3, 5, 10 ou 50).

¹⁷*nbRoutesRecorded* é parâmetro binário apresentando o valor 1 caso o nó em causa consiga definir uma rota para a estação base, e 0 caso contrário.

ID do nó	1	3	5	10	50
2	-		-	ok	
3			-	ok	
4	-		ok		
6	-		ok		
9					ok
11			x	x	x
12			x	x	x
16				x	ok
18			x	x	x
21				x	ok
24				x	ok
26			x	ok	
30			x	x	x
40	x		ok	ok	
42			x	x	x
46			x	ok	
48	x		ok		
49			x	x	ok
51				x	x
62				x	ok
73				x	ok
78			x	ok	
80	x		ok		

Tabela 9: Resultados da análise da criação da topologia da rede

4.1.2 Período de estabilização

Como de uma forma geral apenas se pretende analisar a performance do modelo quando este está estabilizado, definiu-se período de estabilização como sendo o número de ciclos *Wiseroute* que decorrem desde o início da simulação até que esta produza resultados estáveis, isto é, independentes do grau de conectividade da rede e do número de ciclos realizados. Para tal foi realizado um estudo visando garantir essa característica e ao mesmo tempo aferir sobre o tempo total adequado para estas, de modo a não incorrer em simulações morosas.

Tendo em conta esta definição, o estudo consistiu na realização da mesma simulação, fazendo variar o número de ciclos *WiseRoute* para a criação das rotas (1, 2, 3, 4, 5, 10, 25, 50 e 100 ciclos). Posteriormente procedeu-se à análise da correlação dos vários resultados com uma simulação designada padrão, na qual foram repetidos 500 ciclos, um valor arbitrário considerado infinito no contexto do presente estudo. Esta

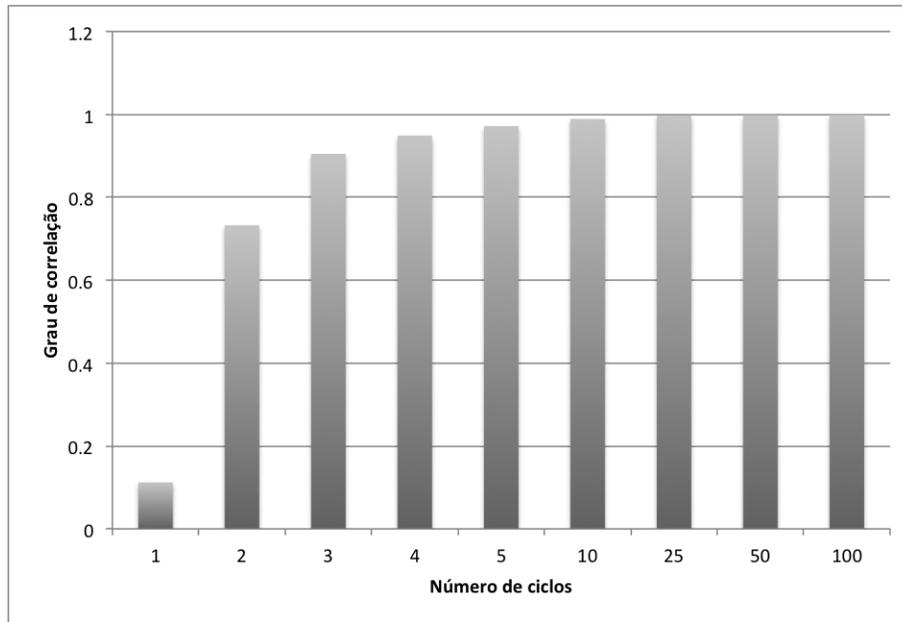


Figura 6: Grau de correlação com 500 ciclos

correlação foi calculada com o recurso à fórmula de Pearson e os resultados podem ser consultados na figura 6. Ao analisá-los constatou-se que existe uma variação acentuada nos resultados das simulações com poucos ciclos (1 a 5). Esta situação espelha o comportamento expectável do período de transição e é também a razão de se eliminar a sua influência nos resultados das simulações. Esta característica aponta também para a importância da escolha de um número de ciclos relativamente mais elevado para a execução das simulações.

Outra constatação foi que as simulações com a execução de 10 ciclos apresentam resultados com um grau de correlação de 0,99 em relação à definida como padrão. Tendo em conta que o número de ciclos é relativamente baixo no âmbito dos objetivos propostos neste trabalho e que o aumento deste não produz melhorias acentuadas nos resultados das simulações, este valor reflete um bom compromisso entre a correlação dos resultados obtidos e os padrão, e o uso eficiente dos recursos disponíveis.

4.1.3 Número de repetições da simulação

No presente contexto (simuladores), executar uma simulação múltiplas vezes produz sempre o mesmo resultado, por isso, para uma determinada configuração, o simulador atribui internamente um número aleatório, *seed*, e utiliza-o sempre que esta for executada. Um dos corolários deste mecanismo é que os resultados obtidos podem ser severamente condicionados pela *seed* gerada. De modo a ultrapassar este facto e a obter dados independentes, repetiu-se a mesma simulação com diferentes *seeds* e foram analisados os seus resultados, pretendendo-se desta forma determinar as condições necessárias para obter resultados com um grau de confiança adequado aos objetivos.

O procedimento constou na configuração de uma simulação simples e na observação do processo de descoberta da rede. A experiência foi então repetida 10 vezes, forçando-se 10 *seeds* diferentes. Posteriormente, os valores médios, as diferenças com os valores simulados, as suas variâncias e os intervalos de confiança para os nós da rede foram calculados para um grau de confiança inicial de 90%. Uma análise a estes resultados revelou que, em média, o valor de flutuação do intervalo era da ordem dos 65% em relação aos valores medidos. Para melhorar este valor, foi calculado o número de repetições necessárias para obter os referidos 90% de confiança. A média dos valores obtidos para essa configuração foi de 133 simulações, o que se considerou como sendo excessivo. As figuras 7, 8 e 9 demonstram o efeito da consolidação dos dados de múltiplas simulações e a relevância de um grau de confiança suficientemente alto. A simulação utilizada para esta demonstração, apenas pretendia analisar o processo de descoberta da rede, mostrando no eixo-x os nós da rede, e no eixo-y o número de pacotes de descoberta duplicados recebidos por cada nó.

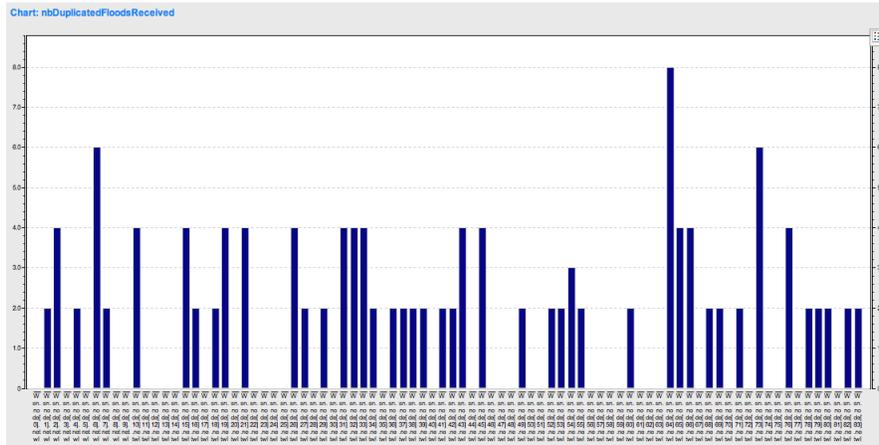


Figura 7: Uma repetição

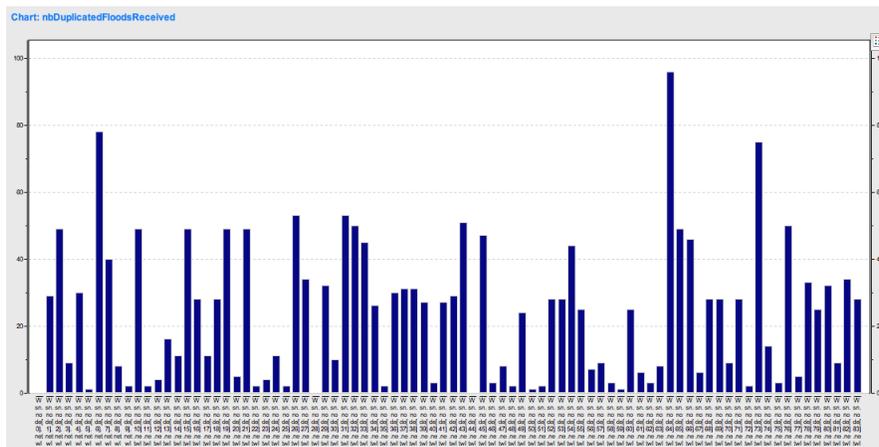


Figura 8: Vinte e cinco repetições

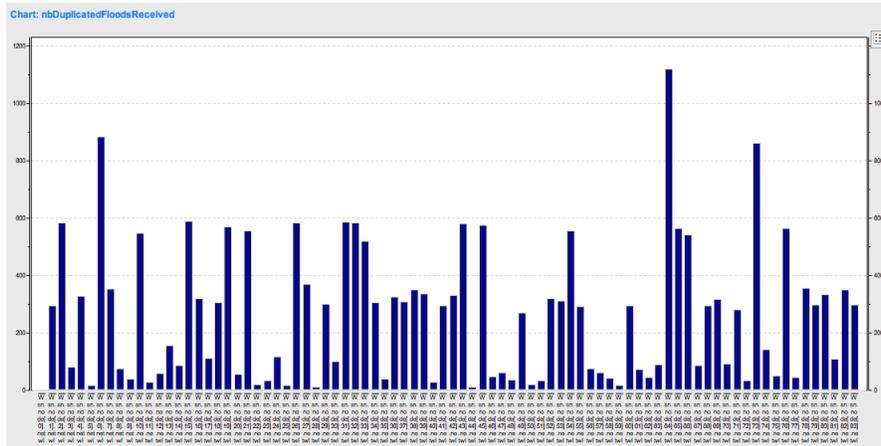


Figura 9: Quinhentas repetições

O procedimento foi repetido para diferentes graus de confiança, podendo consultar-se os resultados finais na tabela 10.

Grau de confiança [%]	50	60	70	80	90	95	98	99
Número de repetições	1	2	5	19	133	812	7894	41908

Tabela 10: Relação entre o grau de confiança dos resultados e o número de repetições

De acordo com a referida tabela, de modo a obter um compromisso entre um bom grau de confiança e o número de repetições exequível, optou-se por considerar apenas 80% de confiança, pelo que as simulações das secções seguintes seriam repetidas 19 vezes (com *seeds* diferentes). No entanto, para facilitar a interpretação e tratamento dos resultados das simulações, arredondou-se este número para 20.

Conclusões

Deste estudo resultaram as seguintes ilações, sendo consideradas sempre que aplicáveis à simulação em causa e aos seus objetivos:

- Considerou-se que o período de transição do modelo corresponde ao intervalo de tempo que ocorre desde o primeiro ciclo de *broadcasts* do protocolo *WiseRoute*, e o quinto. De modo a evitá-lo nas simulações, não foram enviados pacotes pertinentes ao objetivo destas durante esse período.
- Para assegurar que a simulação é executada durante um período de tempo condacente com os objetivos propostos e que os resultados estabilizam, as simulações foram executadas durante um período mínimo equivalente a 10 ciclos *WiseRoute* (incluindo o período de transição). Os resultados apresentam assim um grau de correlação de 99% com simulações 50 vezes mais longas.

- Por forma a garantir que os resultados não são condicionados pelas *seeds* geradas pelo simulador , cada simulação foi repetida 20 vezes (com *seeds* diferentes), para um grau de confiança ligeiramente superior a 80%.

Desta forma pretende-se não só melhorar os resultados das futuras simulações, como também desprovi-los da influência destes fatores inerentes à metodologia usada para a simulação.

4.2 Análise à topologia da rede

Após o estudo das características do *hardware* utilizado e da conceptualização do modelo, o primeiro aspeto a ter em conta ao abordar um projeto desta natureza é a sua conectividade. Entende-se como conectividade da rede, a capacidade desta em estabelecer rotas de comunicação entre a estação base e os restantes nós. A descoberta destes por parte da estação deve ser realizada num intervalo de tempo considerado adequado tendo em conta o âmbito da rede e dos cenários definidos.

Visto que a topologia física da rede já estava definida, a conectividade foi controlada pela configuração de parâmetros no modelo, tais como o limite RSSI de cada nó e o coeficiente de perda de sinal. A análise efetuada nesta secção é feita com base não só nesses parâmetros, como também com o recurso a alguns dos modelos analógicos propostos pelo simulador, validando-se o seu impacto na conectividade da rede e posteriormente na otimização da posição da sua estação base.

4.2.1 Conetividade

Sabendo que a conectividade dos nós é imperativa para a gestão, preservação e segurança do património exposto no Museu, alcançá-la foi considerado como objetivo primário para o estudo desta WSN. Assim, como ponto de partida foi criado um ficheiro com as configurações base, "template.ini", transcrito no apêndice C, prestando especial atenção às ponderações tomadas na secção 4.1, evitando deste modo conclusões baseadas em casos particulares e sem grande importância no cômputo geral, otimizando ao mesmo tempo as simulações.

Neste estudo particular, a métrica utilizada para a avaliação da conectividade da rede foi *nbRoutesRecorded*, podendo esta registar valores binários, isto é, de valor 0 em nós que não criam rota para a estação base, e 1 no caso oposto. A sua sensibilidade foi estudada em relação aos parâmetros *rssThreshold*, que é o limite configurado para cada nó ao RSSI dos *broadcasts* enviados pelo protocolo *Wiseroute*, e em relação ao coeficiente de perda de sinal, α . Como mencionado na análise exposta na secção 3.2.4, havendo alguma subjetividade na escolha deste último parâmetro, inerente ao facto do museu ser composto por espaços de grande heterogeneidade, considerou-se que havia alguma flexibilidade e oportunidade para o melhoramento deste coeficiente.

Para complementar este estudo foram utilizados dois dos modelos analógicos providenciados pela *framework* MiXiM nomeadamente, o *Simple Path Loss* (SPL) e o *Log Normal Shadowing* (LNS), com os quais se procedeu à análise da sensibilidade da rede com vista à concretização da conectividade total entre nós. Ao nível da *framework* utilizada, estes modelos são os responsáveis pela modelação da atenuação do sinal e, juntamente com o módulo *Decider* (Apêndice D), determinam a potência do sinal (combinado) das várias transmissões e se estas são recebidas corretamente nos nós de destino. Em conjugação com o módulo *Connection Manager* (Apêndice D), que controla a definição da distância máxima sobre a qual uma transmissão causa interferência, torna-se possível a análise do comportamento dos rádios utilizados na rede,

modelando assim a camada física dos cenários em questão.

Simple Path Loss

O SPL é um modelo analógico que caracteriza a perda de potência de sinal ao longo da sua propagação. Esta atenuação pode ter origem em vários efeitos como por exemplo a difração, reflexão e refração, podendo ainda ser influenciada pelos obstáculos que encontra, pela distância entre emissor e recetor ou até mesmo pelas características das antenas em jogo. Por ser um modelo simples, é o escolhido para servir de referência na presente secção.

Tendo como objetivo a conectividade total da rede, as combinações testadas com o modelo previamente criado variaram desde as configurações base (*default*), até às limitadas pelo hardware aplicado na rede real.

Numa primeira abordagem, o controlo da receção de mensagens nos nós foi realizada através dos valores impostos nestes para o *rssThreshold*. A primeira configuração testada consistiu na utilização do parâmetro *rssThreshold* a -50 dBm (valor *default*), e do coeficiente de perda de sinal α a 5,04 conforme definido na secção 3.2.4. O intuito desta configuração era o de analisar a facilidade de descoberta da rede, sabendo que o *rssThreshold* era pouco penalizador, considerando as características da rede e que α era uma aproximação.

A segunda configuração utilizada foi uma situação quase limite, onde todos os nós da rede tinham o *rssThreshold* a -96 dBm. Com esta definição específica pretendia-se analisar a importância da decisão tomada pelo Museu em utilizar diferentes tipos de hardware no estabelecimento da conectividade da rede. Esta configuração é equivalente à utilização do hardware XBee S2 em todos os nós da rede. A configuração final desta abordagem representa o limite possível de acordo com as características do hardware utilizado e permite aferir sobre a qualidade da conectividade alcançada.

Numa segunda abordagem foi testada a sensibilidade do modelo ao coeficiente de perda de sinal α . O objetivo consistia em averiguar o valor limite deste parâmetro para o estabelecimento da conectividade e compará-lo com o estimado. Para tal, e para ambos os modelos analógicos, foram sucessivamente testados diferentes valores de modo a alcançar o objetivo.

Os resultados obtidos com as configurações iniciais definidas para o modelo analógico SPL constam na figura 10.

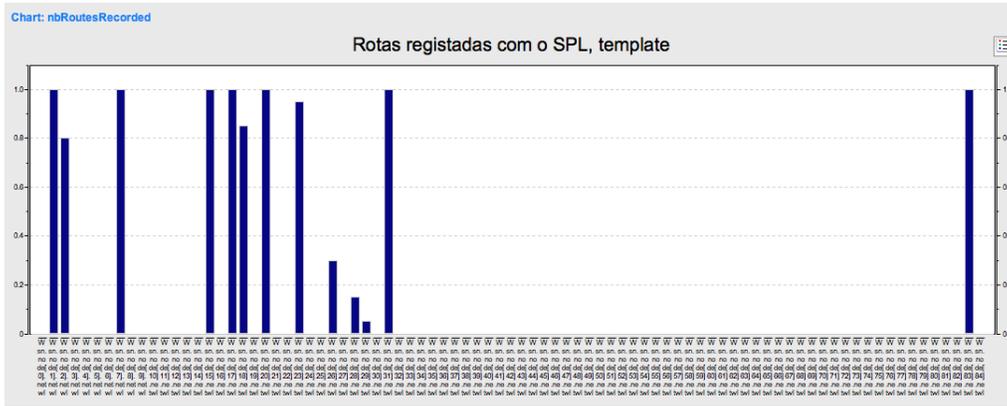


Figura 10: Simulação *template* com o SPL

Como se pode constatar, estes não são indicativos do comportamento pretendido nesta secção, apenas alguns nós, com uma distribuição espacial aparentemente aleatória, conseguem ligar-se à estação base com sucesso. Conclui-se então que a combinação dos parâmetros testados, aliados ao modelo criado, caracterizam um cenário penalizador para a transmissão do sinal. Na prática, tal comportamento poderia muito bem ser representativo de um cenário interior e bastante heterogéneo, como o do Museu da Baleia.

```
**node[*].nic.phy.analogueModels = xmldoc("SPL_config.xml")
```

Listing 4: Utilização do modelo analógico SPL

A segunda configuração foi obtida fazendo variar o limite da potência do sinal recebido das mensagens de criação das rotas, *rssThreshold*, cujo valor definido por defeito é de -50dBm. O estudo da conectividade da rede foi realizado até aos limites do *hardware* utilizado que, dependendo do nó em questão, tinham sensibilidades máximas de -96dBm e -102dBm (Tabela 6).

Os resultados obtidos ao realizar as simulações com o valor -96dBm em todos os nós, estão patentes na figura 11. Da sua análise conclui-se que o valor utilizado está no limiar do que seria o comportamento pretendido, isto é a conectividade total em todas as simulações. Ao analisar os nós em falha, números 36 e 46, observa-se que estes criam rotas em 85 e 90% das simulações, respetivamente. Constata-se ainda que, estes são nós XBee PRO, pelo que a sua sensibilidade máxima é de -102dBm, deixando assim em aberto a capacidade do modelo obter a conectividade total.

A terceira configuração para esta abordagem, impondo os limites de acordo com o *hardware* de cada nó, listagem 5, revelou então uma conectividade perfeita ao longo das 20 repetições, no entanto, é de realçar o facto de que esta conectividade é obtida nos limites das características do *hardware*.

```
**node[0].netwl.rssiThreshold = -96dBm
**node[8].netwl.rssiThreshold = -96dBm
```

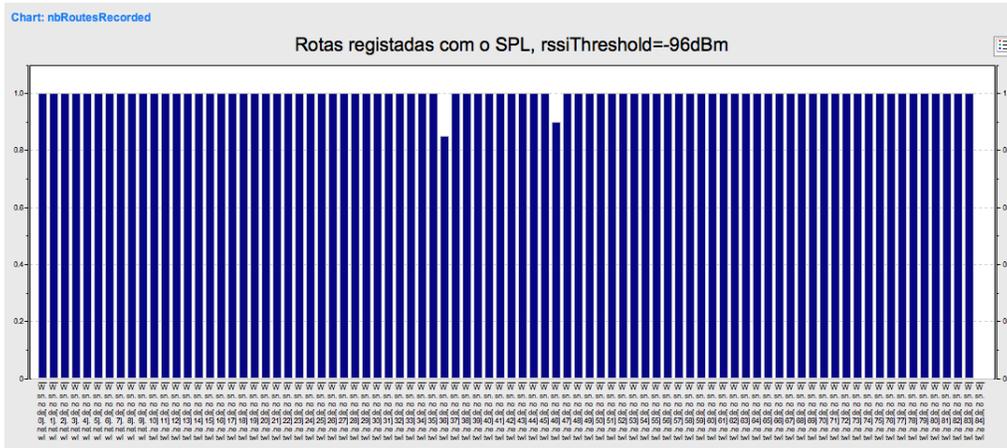


Figura 11: Simulação *template* com o SPL e rssiThreshold=-96dBm, global

```

**.node[13].netwl.rssiThreshold = -96dBm
**.node[14].netwl.rssiThreshold = -96dBm
**.node[20].netwl.rssiThreshold = -96dBm
**.node[22].netwl.rssiThreshold = -96dBm
**.node[24].netwl.rssiThreshold = -96dBm
**.node[33].netwl.rssiThreshold = -96dBm
**.node[34].netwl.rssiThreshold = -96dBm
**.node[43].netwl.rssiThreshold = -96dBm
**.node[44].netwl.rssiThreshold = -96dBm
**.node[45].netwl.rssiThreshold = -96dBm
**.node[50].netwl.rssiThreshold = -96dBm
**.node[54].netwl.rssiThreshold = -96dBm
**.node[56].netwl.rssiThreshold = -96dBm
**.node[57].netwl.rssiThreshold = -96dBm
**.node[59].netwl.rssiThreshold = -96dBm
**.node[80].netwl.rssiThreshold = -96dBm
**.node[*].netwl.rssiThreshold = -102dBm

```

Listing 5: Limite do RSSI máximo de acordo com o hardware

Ainda de acordo com a segunda abordagem definida, a sensibilidade do modelo ao coeficiente α foi verificada. Para tal, várias simulações foram executadas, diminuindo-se o coeficiente (cujo valor inicial era 5,04) em busca da conectividade total limite da rede. As simulações terminaram quando, mesmo para um valor de α igual a 4, que de acordo com a tabela 8 presente na secção 3.2.4, corresponde a ambientes com um só piso, não se descortinou qualquer melhoramento na conectividade, como se pode comprovar pela comparação das figuras 10 e 12.

Uma análise a estes resultados revela a possibilidade da implementação do modelo analógico SPL não apresentar qualquer ligação com o parâmetro α nos limites testados, o que é, no mínimo, de estranhar.

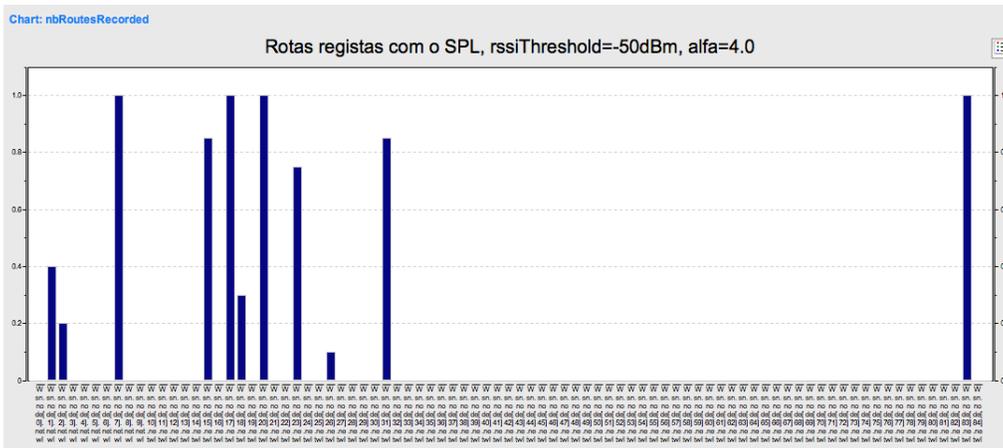


Figura 12: Simulação com o SPL e $\alpha=4.0$

Tendo em conta a primeira das abordagens, conclui-se que a alteração do valor limite *rssiThreshold* faz sentido no presente contexto, visto que vai de encontro a um parâmetro definido pelo hardware, sendo por isso utilizado doravante. Por outro lado, a manipulação do coeficiente α , que não revelou qualquer influência na conectividade da rede, aparenta ser resultado de uma implementação incompleta do modelo SPL, pelo que não será tomado em consideração nas simulações que utilizem este modelo no presente trabalho (será sempre utilizado o valor 5,04 com o SPL).

Log Normal Shadowing

Dada a heterogeneidade do Museu, devido à existência de zonas com elevada densidade de divisões (paredes), grandes corredores e espaços abertos, a perda de sinal para pontos à mesma distância varia consideravelmente. Para a reduzir esta discrepância entre os valores medidos e estimados, foi estudado também o modelo *log-normal shadowing*, LNS, que na sua construção tem em atenção estes desvios.

O processo utilizado para o modelo analógico anterior foi desta vez repetido para o LNS. Assim, e de acordo com a primeira abordagem definida, começou-se por utilizar o modelo LNS com as configurações *default* definidas no ficheiro "template.ini" obtendo os resultados demonstrados na figura 13. Desta disposição, podemos desde já concluir que, para a utilização deste modelo analógico, a rede apresenta uma barreira à conectividade, isto é, uma zona com pior cobertura por parte dos nós implementados. A configuração testada não é capaz de transmitir diretamente ou indiretamente (com recurso a saltos), as mensagens para todos os nós da rede.

```
**.*.node[*].nic.phy.analogueModels = xmldoc("LNS_config.xml")
```

Listing 6: Utilização do modelo analógico LNS

Continuando com as configurações preconizadas, foi analisada a sensibilidade da

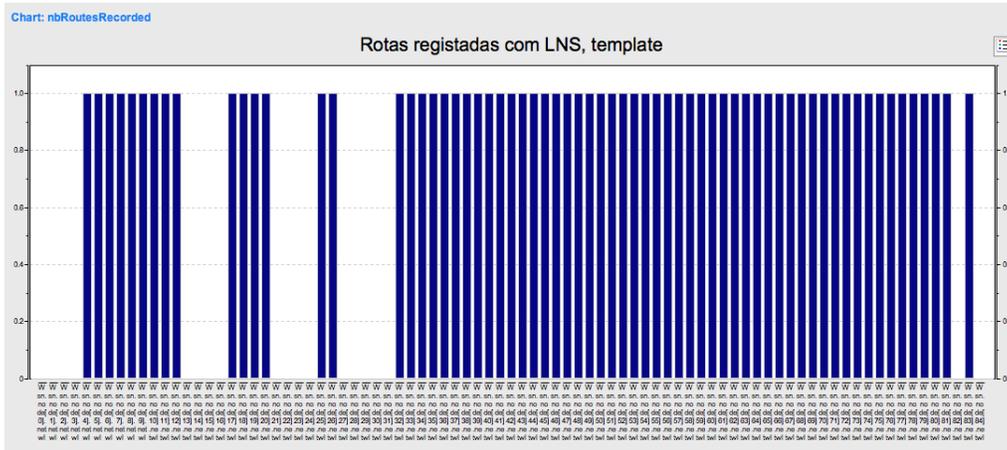


Figura 13: Simulação *template* com o LNS

rede ao parâmetro *rssThreshold*. Esta revelou que, mesmo para a configuração limite definida anteriormente, isto é, com os limites utilizados iguais aos do *hardware* de cada nó, os mesmos nós continuam sem criar a rota para a estação base. Constatase que, para o modelo analógico LNS, o parâmetro em causa não tem qualquer influência, tal como entre o coeficiente α e o SPL.

A segunda abordagem, baseada na manipulação do parâmetro α produziu resultados mais interessantes. Como anteriormente, procedeu-se à realização de várias simulações, diminuindo sempre este valor. Com α igual a 4,70 e como demonstra a figura 15, apenas os nós 0, 2, 23, 24, 27 e 28 permaneciam sem rotas. A conectividade total é depois atingida para o valor de 4,60 sendo que, os últimos dois nós a registarem as suas rotas foram os 0 e 24.

Recorrendo à tabela 8 presente na secção 3.2.4, constata-se que, este valor representa ambientes com comportamento próximo ao de dois pisos, estando em relativa concordância com a estimativa inicial de 5,04 pois existem zonas muito amplas, até com triplo pé direito¹⁸.

Tendo concluído que estes, 0 e 24, eram os nós com pior cobertura por parte da rede (para esta configuração), o comportamento inverso foi avaliado, isto é, aumentou-se o valor do coeficiente. O objetivo desta variante era o de determinar que zona tinha melhor cobertura, através da constatação de quais seriam as últimas a manter os nós ligados à rede.

As primeiras simulações mostraram ainda uma boa resiliência por parte da rede, pois poucos nós perderam conectividade, como demonstrado na figura 16, no entanto, para valores mais altos de α o modelo parece perder a validade. A figura 17 mostra a mesma simulação utilizando o valor de 5.45. Como se pode constatar, inexplicavelmente, alguns nós passam a obter conectividade perfeita ao longo de todas as repetições, como por exemplos os nós 0, 1 e 2, e, em duas das repetições, todos os nós chegam

¹⁸Distância entre o pavimento e o teto.

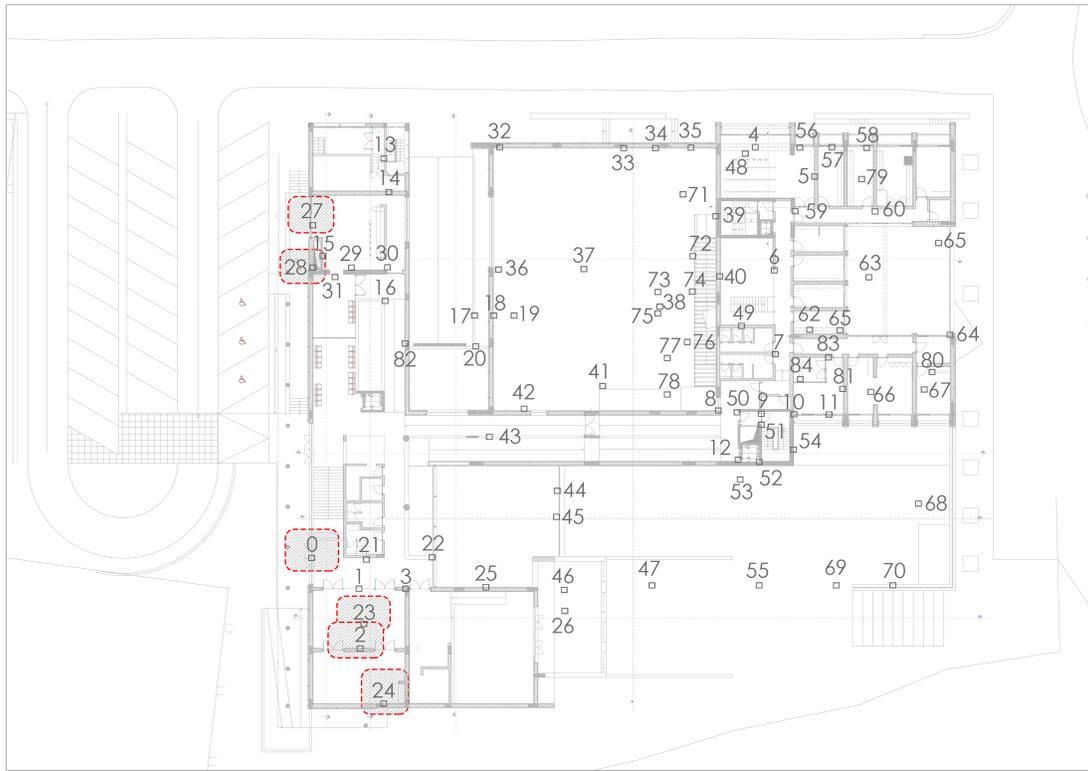


Figura 14: Últimos nós a serem descobertos com o módulo LNS

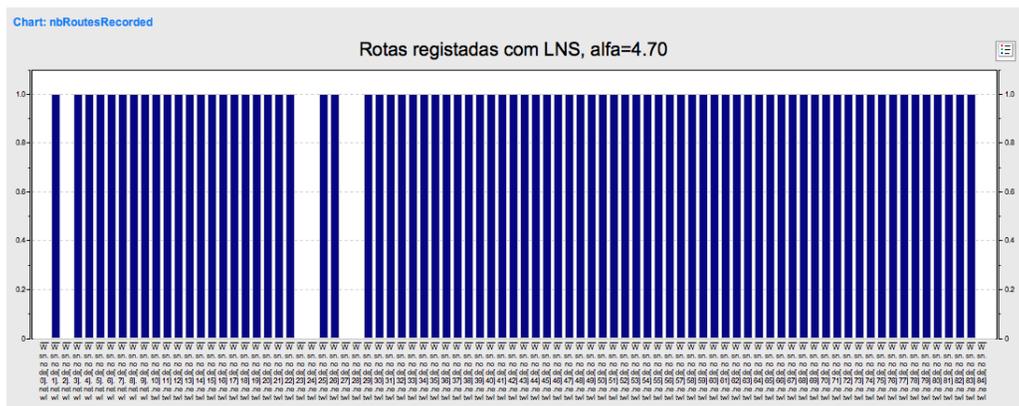


Figura 15: LNS com $\alpha=4.70$

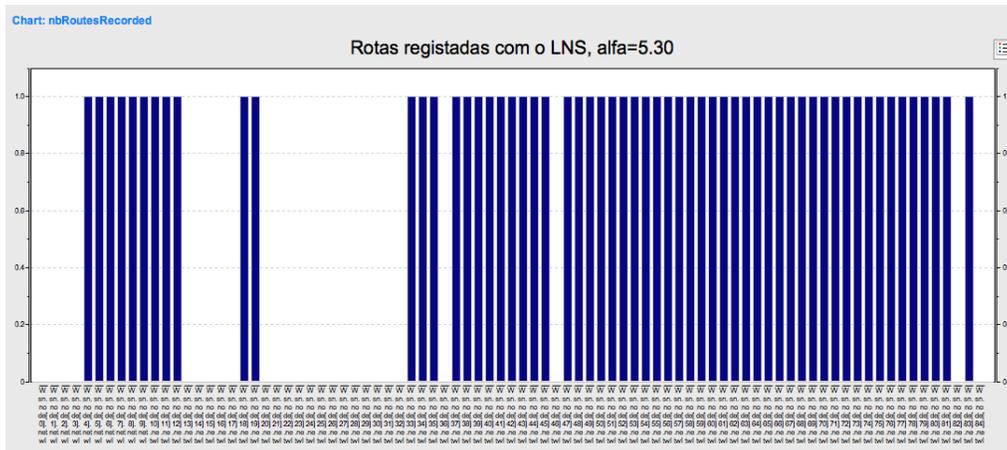


Figura 16: LNS com $\alpha=5.30$

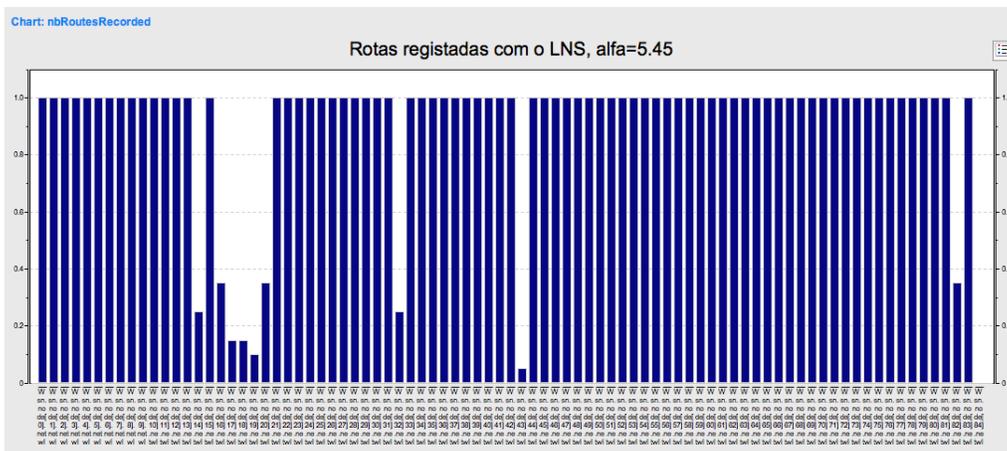


Figura 17: LNS com $\alpha=5.45$

a criar rota para a estação base, contrariando o comportamento expectável para o aumento de α (ver figuras 15 e 16).

Conclusão

Como resultado destas análises, verifica-se um comportamento oposto entre ambos os modelos considerados em relação aos indicadores escolhidos. Aquando da utilização do SPL, a rede mostra-se sensível ao parâmetro *rssThreshold* e insensível ao α , enquanto que, para o modelo LNS, a rede reage de modo inverso. Uma possível explicação para estes comportamentos reside na possibilidade destes modelos terem sido implementados por diferentes equipas, logo com diferentes objetivos e pressupostos.

Da análise às duas situações em que se conseguiu conectividade total conclui-se ainda que:

- O SPL apresenta um comportamento que poderá ser indicativo de um modelo muito penalizador para zonas de grande densidade de nós, isto é, com grandes níveis de interferência. De acordo com a figura 10 os nós que se ligam mais facilmente não apresentam características semelhantes a não ser que estejam em zonas com pouca densidade de nós (excecionalmente os nós 7 e 8 que estão mesmo ao pé da estação).
- O comportamento do modelo LNS é facilmente explicado visto que, os nós mais afastados têm maiores dificuldades em integrar-se na rede. As distâncias de interferência obtidas pela fórmula 2 para esta situação são de 26 e 55 metros, respetivamente para os nós S2 e PRO, um aumento em relação aos 19 e 39 metros obtidos para a situação inicial (valores *default*). Este aumento é relevante pois aumenta a probabilidade de um nó (PRO), instalado na zona central do museu, ter alcance para todos os outros.

4.2.2 Otimização da posição para a estação base

Estando perante um museu com grande heterogeneidade de espaços, a escolha da posição para a estação base pode não ser uma decisão trivial, mesmo quando condicionados ao facto de só se poder utilizar um coeficiente α para a caracterização global do ambiente. Por providenciar melhor cobertura, esta posição será sempre central, mas depende ainda das características particulares desse espaço. Posições em caixas de escadas ou em vitrines por exemplo, não são adequadas à sua localização.

Para abordar este problema, fez-se uma primeira análise visual de modo a seleccionar, das 85 posições definidas para os nós implementados na rede, quais as potenciais candidatas a serem usadas como estação. De acordo com o já referido, os principais critérios usados para esta seleção foram a sua localização central e a preferência por espaços reservados.

Posteriormente, cada nó candidato exerceu as funções de estação base numa simples simulação. Esta consistia no envio de pacotes dos restantes nós para a estação. A

ID do nó	Modelos	
	SPL	LNS
5	0,623	0,022
6	0,488	0,005
10	0,513	0
12	0,548	0
40	0,525	0
48	0,895	0,019
74	0,310	0
78	0,367	0
81	0,766	0,024
83	0,642	0,006
84	0,691	0

Tabela 11: Número médio de saltos dos pacotes

métrica usada para a análise dos resultados foi *meanNbHops*, que indica o número médio de saltos que os pacotes tiveram até alcançar a estação base. De referir que, a implementação do OMNeT++ faz a contagem dos saltos depois de verificar se o nó que recebe o pacote é o destino final do mesmo, pelo que, os pacotes que são transmitidos diretamente para o nó recetor não contabilizarão nenhum salto.

O mesmo procedimento foi repetido para ambos os modelos analógicos, e de acordo com as conclusões tiradas na secção anterior, em simulações com 20 repetições, estando os resultados obtidos presentes na tabela 11. A contagem apresentada pelo simulador não tem em consideração saltos diretos, isto é, se um pacote de origem em A com destino a B chega a este através de uma transmissão direta, para o cálculo deste indicador o número de saltos será zero.

Numa primeira análise aos resultados obtidos para o modelo SPL, constata-se que os nós que obtiveram melhor resultado foram os 74 e 78. Ambos localizam-se no piso -1 e possuem, de entre as posições testadas, as coordenadas mais centrais (a rede está implementada numa área aproximada de 63mx57m, em planta).

```

...
# Node: 74
**.node[74].mobility.initialX = 38m
**.node[74].mobility.initialY = 17m
**.node[74].mobility.initialZ = 8m
...
# Node: 78
**.node[78].mobility.initialX = 35m
**.node[78].mobility.initialY = 27m
**.node[78].mobility.initialZ = 8m

```

Listing 7: Posições dos nós 74 e 78

Ao analisar os nós do piso 2, nomeadamente, 5, 6, 10, 12, 83 e 84, os que apresentaram melhores resultados foram os 6, 10 e 12, também eles, os mais centrais do grupo. Destes, o nó 6 apresenta o melhor resultado, situando-se numa área com menor densidade de nós.

De uma forma geral conclui-se que centralidade da posição tem um grande impacto, verificando-se que, ao colocar a estação base em posições um pouco menos centrais, o número de saltos médio aumenta consideravelmente, como se pode constatar dos resultados obtidos para os nós 5, 48 e 81. Por outro lado, o nó 84, a estação base real, não apresenta um valor muito favorável, especialmente quando comparado com os obtidos nos nós 74 e 78. Estes dois últimos situam-se num piso sem acesso ao público pelo que seriam, de acordo com a análise feita, as melhores posições para a estação base.

No que se refere ao modelo LNS, as condições de transmissão proporcionam uma maior facilidade na criação das rotas. A utilização do modelo analógico LNS por seu lado não permite obter conclusões sobre a melhor posição da estação base mas, com a exceção do nó 84, os nós que apresentam os melhores resultados, também o demonstraram com o SPL.

4.2.3 Conclusões sobre a topologia da rede

Os indicadores utilizados nesta abordagem foram o *rssThreshold* e o parâmetro α e, como ambos os modelos analógicos testados apresentam comportamentos inconsistentes com estes indicadores, os dois modelos são utilizados e avaliados de forma independente.

O modelo SPL apresenta o comportamento expectável em relação à configuração do *rssThreshold*, melhorando a conectividade da rede com a sua diminuição. Por outro lado, o mesmo mostra-se indiferente ao parâmetro α dando a entender que não foi implementado no SPL e penaliza as zonas com maior densidade de nós.

O modelo analógico LNS apresenta resultados baseados na distância de interferência, alcançando a conectividade total da rede com a calibração do coeficiente de perda de potência do sinal transmitido α , para o valor 4,6. Por seu lado, não mostra qualquer ligação ao indicador *rssThreshold*.

Na análise à posição ótima para a estação base, de acordo com as posições definidas pela solução implementada, conclui-se que a zona central do Museu é a mais adequada e que as posições 74 e 78 seriam as ideais. Estas, além de estarem em posições centrais, estão num piso de acesso restrito.

4.3 Análise à sobrecarga da rede

Estando perante um número elevado de nós e sensores, o desempenho da WSN do museu pode facilmente deteriorar-se devido à quantidade de dados a ser transmitida. Esta deterioração é exacerbada ainda pelas condições específicas dos cenários interiores, cujo elevado número de obstáculos, nas suas mais diversas formas, conduzem a inúmeras retransmissões de dados.

Por conseguinte, o estudo das condições de transmissão, especialmente em relação à sua periodicidade (variável dada a natureza dos vários tipos de sensores) e o conhecimento dos limites de desempenho aceitáveis da rede revestem-se de importância, não só para esta rede, como também para redes similares.

Os objetivos das seguintes simulações são os de propor procedimentos para a análise da sobrecarga da rede e a identificação de medidas para a sua correção. Para tal analisa-se a resposta da rede quando exposta a condições de stress, tendo em atenção indicadores como o número de pacotes recebidos na estação base, a latência e o número de retransmissões (*backoffs*). Uma retransmissão acontece quando uma tentativa de transmissão de dados é abortada por se verificar que o meio está ocupado, existindo algoritmos que determinam posteriormente quanto tempo depois dessa tentativa uma nova deverá ser realizada.

Inicialmente o estudo foi só previsto para o protocolo *Wiseroute* mas, sabendo que nos casos de falha da estação base, como por exemplo avaria, a rede compensa adotando um comportamento do tipo *flooding*, um protocolo deste tipo foi analisado. Deste modo foi também estudada a performance do próprio simulador visto que protocolos deste tipo consomem quantidades elevadas de recursos.

Quando uma rede é configurada com um protocolo de encaminhamento como é o caso do *Wiseroute*, as transmissões são efetuadas por *broadcast* especificando nos pacotes não só a identificação desse nó final, como também a do próximo salto. Qualquer nó vizinho ao emissor que receba este pacote, após a verificação destes identificadores, procede novamente ao *broadcast* do pacote ou ao seu descarte. O processo repete-se até os pacotes chegarem ao seu destino.

Por outro lado, uma rede com um protocolo do tipo *flooding* transmite os pacotes em forma de *broadcast*, especificando apenas o destino final. Os nós que recebem estas mensagens repetem o processo uma vez, sendo que, no final do todos os nós da rede recebem a mensagem, independentemente da existência de um destinatário específico na mensagem em questão. A exceção a este comportamento é feita pelo nó destinatário final da mensagem, que não realiza o *broadcast*. Visto que todos (menos um) os nós repetem o *broadcast* este tipo de comportamento é obviamente ineficiente provocando geralmente degradação do comportamento da rede (se usado de forma descontrolada).

Os cenários de estudo destas condições baseiam-se no envio de pacotes por parte dos nós emissores (todos os nós da rede exceto a estação base) para a estação base de modo compreender o comportamento da rede perante as configurações selecionadas. O envio destes pacotes é feito tanto de forma sincronizada pelos nós emissores (sendo este o cenário mais penalizador) como de forma não sincronizada. A análise à sobrecarga é

ainda complementada com o estudo de configurações onde número de pacotes enviados pelos emissores e a sua taxa de transmissão são alterados. Deste modo, pretende-se então compreender a sensibilidade do modelo em relação a estas características e balizar os limites de eficácia da rede.

Os indicadores utilizados foram o número de pacotes recebidos na estação base, *nbDataPacketsReceived*, a latência média medida na receção dos pacotes, *latency:mean* e o número de retransmissões, *nbBackoffs*. Sobre estes indicadores infere-se então o comportamento da rede sobre as condições de stress implementadas, nomeadamente em relação aos pacotes perdidos na rede, aos atrasos e sobre as zonas mais congestionadas.

De realçar que, os resultados apresentados de seguida refletem sempre a agregação das 20 repetições efetuadas para cada configuração. Como em pontos anteriores, ambos os modelos analógicos foram testados assim como a sua combinação.

4.3.1 Wiseroute

Os indicadores estudados estão de uma forma ou outra, relacionados com o grau de saturação suportável pela rede e refletem a sua eficácia, pelo que, várias configurações foram testadas. Estas, englobam o envio de pacotes na rede com uma cadência periódica, com taxas de envio de 1p/1s (1 pacote por segundo), 1p/500ms e 1p/100ms, sendo o seu envio primeiro efetuado de forma sincronizada e posteriormente, de forma dessincronizada, entre todos os nós. Embora como referido na secção 3.2.2 as taxas de envio reais sejam maiores do que as aqui testadas, serve a presente análise para demonstrar um procedimento possível, para avaliar o seu impacto numa qualquer rede de sensores sem fios em estudo. Para simular o envio de pacotes não sincronizados pelos nós, iniciou-se de forma aleatória o seu envio, tendo sido utilizados valores dentro de intervalos de 1, 2 e 3 minutos. Foram ainda testados os modelos analógicos SPL, LNS e uma combinação entre ambos. Como não é do âmbito desta dissertação a afinação destes modelos, foram utilizados como providenciados no simulador. Apesar disso, os procedimentos aqui utilizados continuam válidos para configurações diferentes dos modelos ou mesmo com a utilização de outros que se julguem mais adequados.

1 pacote

A primeira abordagem centrou-se no estudo da rede perante a introdução de um pacote por parte dos nós emissores (84 no total). Sem entrarmos em cálculos para determinar os tempos de envio dos pacotes de modo a chegarem todos ao mesmo tempo à estação base ou de modo a originarem estrangulamentos, a combinação mais crítica é o envio simultâneo destes, pelo que foi a primeira configuração testada.

Posteriormente, de modo a amenizar o pico de tráfego assim criado, foi introduzido um intervalo de tempo no qual as aplicações de cada nó são inicializadas de forma aleatória (não sincronizada). Esta medida visa também aproximar o modelo de tráfego, ao presente na rede do museu.

As instruções presentes na listagem 8 refletem a criação desse desfasamento, sendo

Protocolo: Wiseroute			
M. Analógico	Envio	Pacotes Recebidos	Eficácia
SPL	Sincronizado	65,4	77,86%
	Não sincronizado	63,8	75,95%
LNS	Sincronizado	83,8	99,76%
	Não sincronizado	83,9	99,88%

Tabela 12: Pacotes recebidos na estação base nas simulações com o envio de 1 pacote por nó, num total de 84, com envio sincronizado e aleatório no intervalo de 1 minuto.

as aplicações de cada nó, responsáveis pelo envio destes pacotes, inicializadas com uma probabilidade seguindo uma distribuição uniforme, entre os 5 e 65 segundos.

```
**node[*].apl.initializationTime = intuniform(5,65)*1s
```

Listing 8: Dessincronização

Os resultados destas simulações estão presentes na tabela 12. Desta constatamos que a implementação utilizada do modelo SPL é mais penalizadora para o envio de pacotes, do que o modelo LNS, apresentando uma eficácia muito inferior. Por outro lado, o espaçamento implementado no envio dos pacotes também não apresentou efeitos visíveis pelo que, para obter melhorias ao nível da receção dos pacotes com o SPL de acordo com os resultados obtidos, as medidas teriam de ser feitas ao nível das configurações do próprio modelo analógico, ou da concetualização do modelo da rede WSN, alterando por exemplo o coeficiente de perda de sinal α , no entanto estas medidas não se enquadram no âmbito da presente avaliação.

5 pacotes

A segunda abordagem à sobrecarga da rede passou por elevar o número de pacotes injetados por cada nó emissor para 5, totalizando 420 pacotes. Deste modo já foi possível testar as diferentes taxas de transmissão estando os resultados expostos nas tabelas 13, 14 e 15.

Sobre as configurações com envio sincronizado, tabela 13, destaca-se a perda de eficiência da rede perante taxas de transmissão de 1p/100ms. Esta perda é mais acentuada na configuração com o modelo analógico LNS, obtendo-se resultados similares nas outras combinações de modelos analógicos, denotando um limite físico da rede, isto é, de saturação, independente da combinação de modelos utilizada. O mesmo comportamento pode ser confirmado ao analisarmos a latência, tabela 15, duplicando para o LNS ao aumentarmos a taxa de transmissão de 1p/500ms para 1/100ms. Ao comparar os resultados do modelo SPL com a sua combinação com o LNS, conclui-se que este é o que mais influência tem, pois os resultados são similares.

Ainda de acordo com os dados expostos nas tabelas 13 e 14, confirma-se que pe-

Protocolo: Wiseroute							
Indicador	M. Analógico	Taxa de transmissão			Eficácia[%]		
		1seg	500ms	100ms	1seg	500ms	100ms
Pacotes Recebidos	SPL	252	258	95.7	60,00	61,43	22,79
	LNS	401	376	94	95,48	89,52	22,38
	SPL + LNS	252	253	87.3	60,00	60,24	20,79

Tabela 13: Resultados para os pacotes recebidos nas simulações com o envio sincronizado de 5 pacotes (num total de 420).

Protocolo: Wiseroute							
Indicador	M. Analógico	Taxa de transmissão			Eficácia[%]		
		1seg	500ms	100ms	1seg	500ms	100ms
Pacotes Recebidos	SPL	272	279	278	64,76	66,43	66,19
	LNS	419	376	417	99,76	89,52	99,29
	SPL + LNS	336	333	329	80,00	79,29	78,33

Tabela 14: Resultados para os pacotes recebidos nas simulações com o envio não sincronizado num intervalo de 1 minuto de 5 pacotes (num total de 420).

Protocolo: Wiseroute							
Indicador	M. Analógico	Envio de 5 pacotes					
		Sincronizados			Não sincronizados		
		1seg	500ms	100ms	1seg	500ms	100ms
Latência	SPL	0,0090	0,0101	0,0150	0,0086	0,0087	0,0086
	LNS	0,0057	0,0065	0,0139	0,0052	0,0065	0,0053
	SPL + LNS	0,0090	0,0112	0,0162	0,0097	0,0096	0,0097

Tabela 15: Resultados para as latências nas simulações com o envio de 5 pacotes, com e sem sincronização num intervalo de 1 minuto.

rante as condições iniciais idealizadas, o modelo SPL apresenta uma grande perda de pacotes, mesmo quando o início da transmissão é realizada de forma aleatória no intervalo de 1 minuto, revelando uma eficácia de 66,19% para a taxa de transmissão de 1p/100ms. As mesmas simulações foram repetidas posteriormente para intervalos de inicialização de 2 e 3 minutos de modo a analisar com maior pormenor a sua sensibilidade a este parâmetro. Os resultados obtidos encontram-se transcritos na tabela 16 aferindo-se destes um aumento da eficácia para 78,09% perante um espaçamento inicial de 2 minutos e de uma ligeira quebra aos 3 minutos. Outro pormenor relevante prende-se com o facto dos pacotes recebidos serem constantes para estes intervalos, isto é, independentes da taxa de transmissão. Este facto reforça a existência de um limite a partir do qual não existem ganhos ao espaçar as transmissões. No caso particular desta rede, tal limite seria aos 2 minutos.

Ainda sobre as mesmas simulações, a figura 18 mostra a comparação entre simulações com taxas de transmissão de 1p/100ms com o envio sincronizado e não sincronizado num intervalo de 1 minuto. Ao espaçar-se o envio de pacotes reduz-se retransmissões garantindo deste modo uma maior eficácia da rede. Em cenários onde a vida útil de um nó é pertinente e é necessário controlar os gastos de bateria, reduzir o número de retransmissões é fundamental.

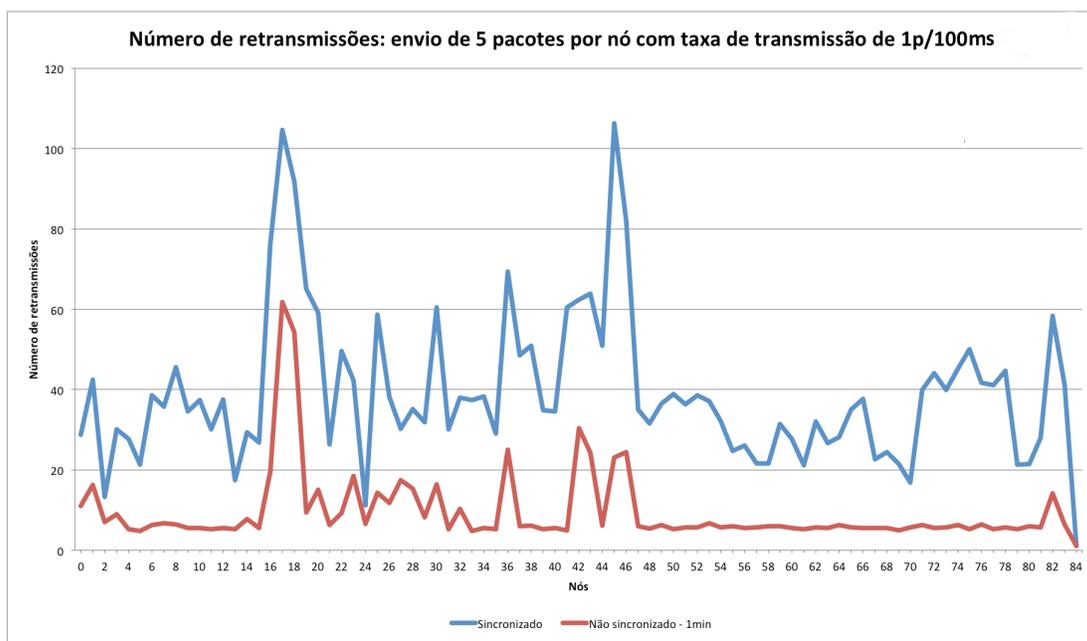


Figura 18: Número de retransmissões no cenário de envio de 5 pacotes sincronizados e não sincronizados num intervalo de 1 minuto, com taxa de transmissão de 1p/100ms.

Dada a natureza da transmissões presentes na rede do Museu, esta abordagem já está implementada visto que, dependendo da função específica de cada sensor, estes têm taxas de transmissão bem diferentes, como já exposto no 3.2.2, tornando o risco de sincronização das transmissões muito reduzido.

Protocolo: Wiseroute							
Indicador	Taxa	Não sincronizado			Eficácia[%]		
		1 min	2 min	3 min	1 min	2 min	3 min
Pacotes Recebidos	1s	272	328	321	64,76	78,09	76,43
	500ms	279	328	320	66,43	78,09	76,19
	100ms	278	328	318	66,19	78,09	75,71

Tabela 16: Resultados para o envio de 5 pacotes espaçados em intervalos de 1, 2 e 3 minutos.

A figura 19 apresenta uma amostra dos resultados obtidos para o indicador associado ao número de retransmissões, *nbBackoffs*, no caso específico das simulações com intervalo de transmissão de 100ms entre pacotes, cujo início das transmissões foi gerado de forma sincronizada com a combinação de ambos os modelos. Como se pode observar, o número de retransmissões é semelhante ao longo dos nós havendo no entanto algumas exceções. O levantamento aos nós com maior número de *nbBackoffs* para as simulações efetuadas para a combinação dos dois modelos analógicos revela que os nós 17, 18, 36, 42, 43 e 45 apresentam consistentemente os valores mais elevados de retransmissões.

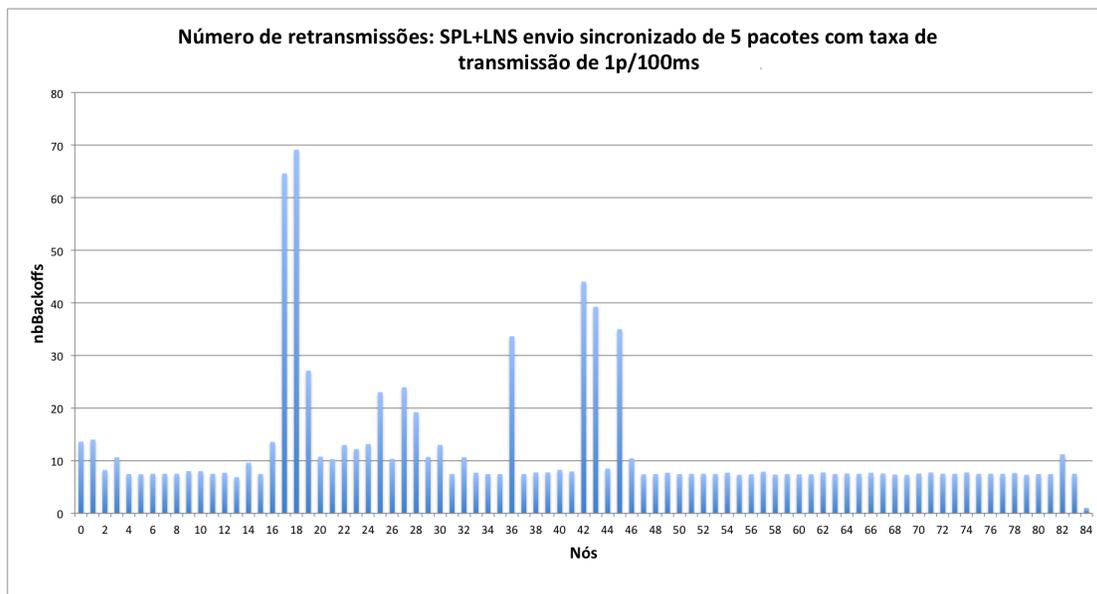


Figura 19: Número de retransmissões para simulações com o envio de 5 pacotes por nó de forma sincronizada utilizando uma combinação de ambos os modelos analógicos e uma taxa de transmissão de 1p/100ms.



Figura 20: Pisos sobrepostos com a localização dos nós identificados com maior número de retransmissões

Estes dados indicam a existência de um congestionamento devido às condições nos nós a jusante, pontos críticos para a passagem dos pacotes em rota para a estação base. Como se pode constatar da figura 20, ao sobrepor-se a localização de todos os nós num mesmo plano é possível identificar uma barreira à transmissão dos pacotes. De modo a compreender a interação entre o protocolo *Wiseroute* e o modelo LNS, realizou-se um pequeno estudo, apresentado no Apêndice E, com o objetivo de mostrar a fraca escalabilidade do mesmo.

4.3.2 Flooding

Dada a natureza dos protocolos como o *Flooding* só em situações com topologias muito particulares, é que nem todos os nós recebem um pacote enviado por *broadcast*, revelando essas situações, a existência de zonas críticas da rede. Um exemplo desse tipo de configuração topológica é a figura 21 no qual, se o nó 0 efetuar um *broadcast* com destino ao nó 1, este nunca será recebido pelo nó 2 pois o 1 não dará continuidade ao processo. Será por isso de esperar um protocolo muito robusto e redundante mas ineficiente no que respeita à energia dispendida nas transmissões.

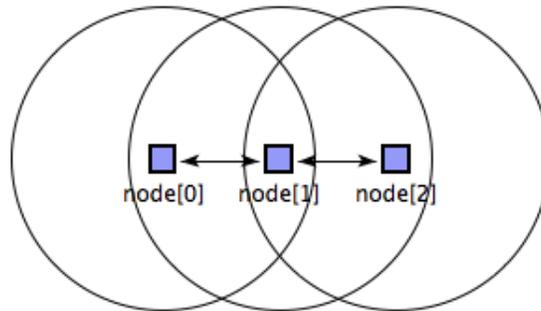


Figura 21: Caso particular onde nem todos os nós recebem os pacotes enviados por broadcast: envio de uma mensagem do nó 0 para o nó 1.

```

**node[*].networkType = "Flood"
...
**node[*].appl.broadcastPackets = true
**node[*].appl.trafficType = "periodic"
**node[*].appl.trafficParam = 30 s

```

Listing 9: Exemplo de Flooding

Reutilizando a abordagem dos cenários anteriores, foram efetuadas as simulações cujos resultados se encontram expressos nas tabelas 17 e 18.

Protocolo: Flooding			
Modelo	Envio	Pacotes Recebidos	Eficácia
SPL	Sincronizado	399	95,00%
	Não sincronizado	400	95,24%

Tabela 18: Pacotes recebidos nas simulações com o envio de 5 pacotes (em 420 possíveis).

Protocolo: Flooding			
Modelo	Envio	Pacotes Recebidos	Eficácia
SPL	Sincronizado	80,0	95,23%
	Não sincronizado	80,0	95,23%
LNS	Sincronizado	83,7	99,64%
	Não sincronizado	83,8	99,76%

Tabela 17: Número de pacotes recebidos nas simulações com o envio de 1 pacote (em 84 possíveis).

A primeira observação feita é o aumento da eficiência do modelo SPL nestas condições, que anteriormente era 77,86% para pacotes enviados de forma síncrona e de 75,95% para pacotes enviados aleatoriamente num intervalo de 1 minuto, tendo subido com a utilização do protocolo *Flooding* para os 95,23% em ambos os casos. Outra observação é que o espaçamento não aparenta ter efeito na eficácia da rede. Em qualquer altura um *broadcast* pode atingir a estação base tendo este tipo de protocolo um grau de redundância extremamente elevado.

Ainda devido à sua natureza, o número de transmissões cresce exponencialmente com o número de nós, o que faz com que estes tenham de processar muito mais informação, mesmo descartando a sua maioria. Este excesso de processamento traduz-se num maior consumo de bateria, incomportável na maioria dos cenários WSN e num aumento dos níveis de interferência (figura 22).

No que respeita à latência, visto que todos os nós recebem pacotes (embora a maioria seja descartada posteriormente) é possível estudar os seus níveis, e não apenas na estação base. A figura 23 mostra os valores atingidos para a configuração de envio de 5 pacotes. Como seria de esperar, os nós mais distantes da estação base, 0, 1, 2, 3, 21, 22, 23 e 24, situados na zona mais a oeste do museu, são os que apresentam níveis mais altos para a latência.

Outra particularidade do protocolo é o que o seu tempo de execução da simulação e a quantidade de informação gerada pelo simulador são superiores. No caso específico do envio sincronizado de 5 pacotes, as 20 simulações tiveram um tempo real de execução aproximado de 4 horas e a informação gerada em disco foi de 22,80 GB.

Ao analisar estas simulações verifica-se que o modelo SPL mantém a sua eficácia.

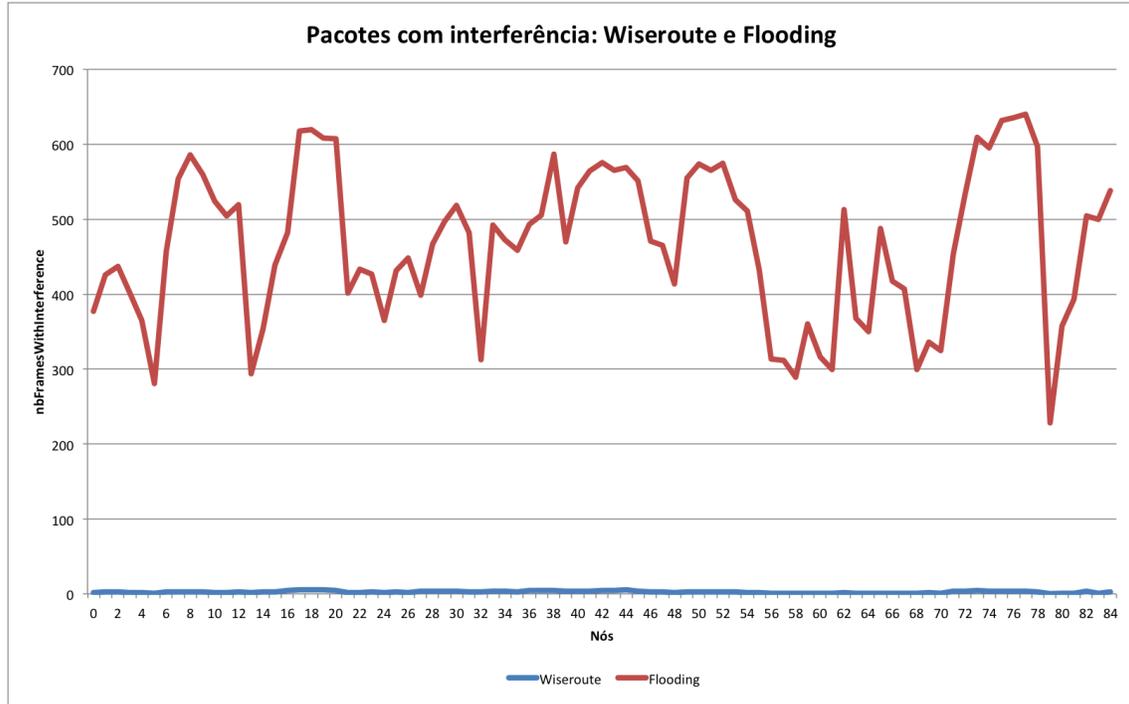


Figura 22: Comparação entre os níveis de interferência perante o envio sincronizado de 5 pacotes por nó entre os protocolos Wiseroute e Flooding.

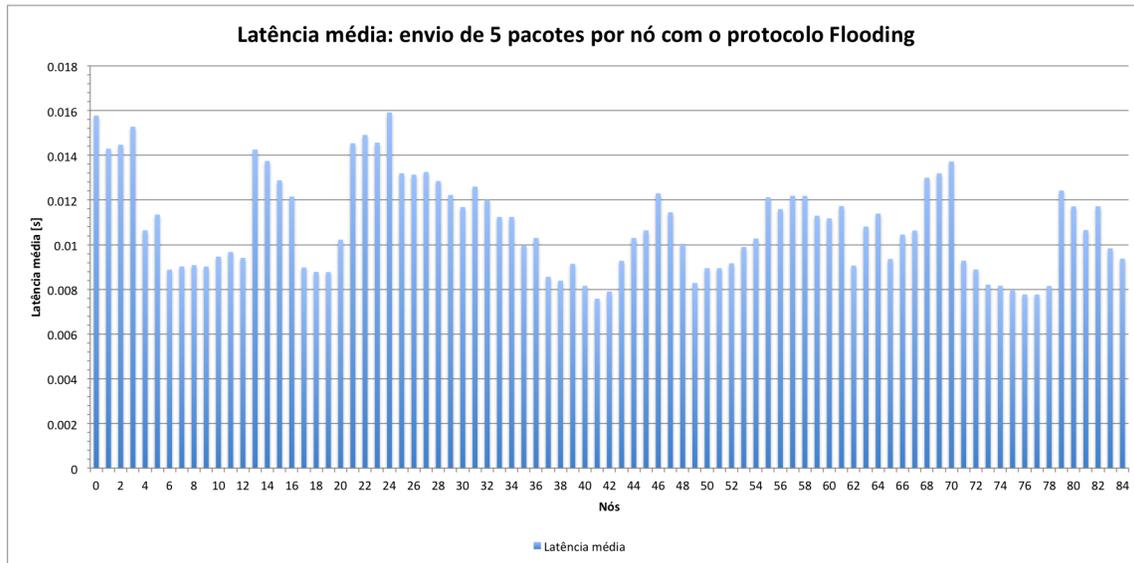


Figura 23: Níveis de latência perante o envio sincronizado de 5 pacotes por nó com o protocolo Flooding.

4.3.3 Conclusões sobre a sobrecarga da rede

Nas simulações efetuadas, independentemente do modelo analógico utilizado, deteta-se uma perda de eficácia acentuada para taxas de transmissão de 1p/100ms, existindo por outro lado um benefício visível ao espaçar o envio dos pacotes. Esse benefício está relacionado com a capacidade da rede em escoar os pacotes, tendo-se obtido o melhor resultado para o espaçamento aleatório no intervalo de 2 minutos, valor após o qual não se vislumbram melhorias.

Foram também identificados nós onde o número de retransmissões é consideravelmente superior. Por conseguinte, os nós a jusante são pontos de estrangulamento para o desempenho da rede. Estas situações podem ser amenizadas através da redução das taxas de transmissão de pacotes e evitando a sincronização no seu envio.

Não havendo restrições em relação à bateria dos nós pode ser benéfico, em situações de monitorização esporádica, utilizar um protocolo do tipo *Flooding* para a transmissão de informação. No entanto os níveis de interferência no meio aumentam significativamente.

5 Conclusões e Trabalho Futuro

O presente trabalho tinha como objetivos definidos, a criação de orientações pertinentes à implementação e gestão de projetos baseados em WSN *indoor*, optando-se por recorrer a simulações em detrimento de estudos analíticos ou da análise de dados reais. A flexibilidade obtida da utilização de um simulador e de um modelo, criado com o conhecimento prévio do *hardware* utilizado (fidedigno), tornam esta opção a mais viável e adequada para o estudo da rede em questão.

Através do estudo realizado aos simuladores concluiu-se que, nesta área de investigação, ainda são muito específicos (logo pouco abrangentes) e aparentam alguma imaturidade. A escolha do simulador mais adequado aos desafios do Museu recaiu sobre o Mixim. De notar que, o Castalia foi excluído por falta de suporte a topologias com *multi-hop* e, no caso do NS-3, por não suportar o protocolo 802.15.4. Para evitar a utilização de dados incoerentes, derivados de descobertas incompletas da rede, da sua utilização em períodos instáveis ou mesmo da realização de simulações curtas, as análises estatísticas realizadas mostraram que, para o presente modelo, topologia e recursos disponíveis, não se deveriam gerar pacotes antes do quinto ciclo do protocolo *Wiseroute*, que as simulações teriam num período mínimo de 10 ciclos e a mesma simulação seria executada 20 vezes. Os resultados apresentados desta forma estão, de acordo com os compromissos assumidos, isentos da influência de fatores derivados da metodologia usada na simulação. Esta validação estatística do modelo é uma importante contribuição da presente dissertação, pois a sua aplicabilidade é transversal a esta problemática (facilmente aplicada noutros contextos sobre simulações), permitindo ajustar os recursos disponíveis à precisão desejada.

Como identificadas, as condicionantes referentes à implementação do projeto, aliadas à necessidade de minimizar o seu impacto visual, de controlar os efeitos das interferências, dos condicionalismos da localização de alguns nós e do seu hardware, resultou numa rede topológica densa e com elevado grau de redundância.

As contribuições ao nível das simulações realizadas no contexto da conectividade prendem-se com a demonstração de que os modelos utilizados apresentam comportamentos incompletos em relação aos parâmetros em estudo, revelando imaturidade dos mesmos. Os resultados obtidos foram, por vezes, contraditórios, dificultando a sua interpretação. Identificaram no entanto que, a escolha de uma posição central ao cenário de atuação é um fator fulcral para a determinação da posição ótima para a estação base. As simulações revelaram as posições 74 e 78 como as melhores localizações, ambas centrais e no piso inferior ao da estação base real.

As simulações relacionadas com a carga da rede identificaram uma 'barreira' onde o número de retransmissões era substancialmente maior. A área identificada funciona como um estrangulamento ao tráfego que segue em direção à estação base. Constatou-se também que a rede apresenta uma perda de eficiência acentuada para taxas de transmissão na ordem de 1p/100ms. Por outro lado, espaçar os pacotes de forma aleatória num intervalo de 2 minutos produz melhorias ao nível do número de retrans-

missões. Este espaçamento, assim como o envio assíncrono de pacotes, amenizam a dificuldade de escoamento dos pacotes no congestionamento identificado previamente. Destas simulações concluiu-se ainda que, o protocolo *Flooding* pode ser benéfico em cenários que, não havendo limitações com as baterias, se procuram níveis de conectividade mais elevados e que não sejam requisitos os níveis de interferência da rede ou a latência. Outras importantes conclusões foram que, não existindo objetivos sobre as simulações previamente definidos, os tópicos abordados foram enviesados pelas características do simulador escolhido e a incapacidade de validar os resultados obtidos na rede implementada no Museu.

Assim, caso houvesse necessidade de implementar uma rede de sensores sem fios num cenário *indoor*, efetuar-se-ia um estudo ao modelo a utilizar, incluindo a definição do parâmetro α e a validação dos períodos de transição, estabilização e número de repetições da simulação. De acordo com um conjunto bem definido de objetivos e um levantamento espacial apurado, as simulações a executar seguiriam a lógica apresentada no presente trabalho. Inicialmente realizar-se-iam simulações para a determinação da posição ótima da estação base e, a partir desta e das condições de implantação definiriam-se as posições dos restantes nós (de acordo com as necessidades do projeto), sendo que as simulações seguintes centrar-se-iam na análise da conectividade e otimização espacial da rede. Uma vez satisfeitos com os resultados, proceder-se-ia à sua implantação.

Do presente trabalho, identificam-se ainda as seguintes oportunidades de desenvolvimento futuro. A simulação de falhas em nós críticos de modo a analisar a resposta da rede. Este tipo de cenários poderia identificar áreas do Museu com pouca redundância no encaminhamento de pacotes. Esse estudo permitiria também aferir a resiliência da rede, nomeadamente no que respeita ao número de nós necessários/desnecessários para manter determinados níveis de resposta da rede. Outro ponto a analisar seria a comparação com dados reais e a possibilidade de testar as conclusões obtidas na rede real. Sem a contribuição do Museu, outra alternativa seria a validação das conclusões com o recurso a um diferente simulador.

Os objetivos da presente dissertação foram alcançados, mas não foram explorados como desejado, diferentes cenários e abordagens à otimização da rede.

Referências

- [1] P. Huang, L. Xiao, S. Soltani, M. W. Mutka, and N. Xi, “The evolution of MAC protocols in wireless sensor networks: A survey,” vol. 15, no. 1, pp. 101–120.
- [2] Z. Gengzhong and L. Qiumei, “A survey on topology control in wireless sensor networks,” in *2010 Second International Conference on Future Networks*, IEEE, jan 2010.
- [3] www.elsevier.com/locate/comnet, “Computer networks 38 (2002),” 2002.
- [4] M. Tubaishat and S. Madria, “Sensor networks: an overview,” *IEEE Potentials*, vol. 22, pp. 20–23, apr 2003.
- [5] Y.-M. Huang, M.-Y. Hsieh, and F. E. Sandnes, “Wireless sensor networks and applications,” IEEE, 2008.
- [6] A. Survey, of Military, Applications, and of, “Meco 2012, bar, montenegro, june,” 2012.
- [7] A. Pantelopoulos and N. Bourbakis, “A survey on wearable sensor-based systems for health monitoring and prognosis,” vol. 40, no. 1, pp. 1–12.
- [8] B. Latré, B. Braem, I. Moerman, C. Blondia, and P. Demeester, “A survey on wireless body area networks,” *Wireless Networks*, vol. 17, pp. 1–18, nov 2010.
- [9] V. . o and March, “International journal of computer applications (0975 – 8887),” 0975.
- [10] W. SENSOR, NETWORKS, and A. peer-to-peer cross-layering approach, “A fire monitoring application for scattered,” 7200.
- [11] ISSN, Volume, Issue, and July, “International journal of soft computing and engineering (ijsce),” 2013.
- [12] A. A. K. S., K. Ovsthus, and L. M. Kristensen., “An industrial perspective on wireless sensor networks — a survey of requirements, protocols, and challenges,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1391–1412, 2014.
- [13] S. C, U. R, I. T, Y. A, N. D, P. I, V. C, Y. I, N. EM, E. R, G. I, N. G, and W. NETWORKS, “Buttyan layout 10/6/10 12:39 pm page,” in *44*, IEEE, 1536.
- [14] M. r. Akhondi, A. Talevski, S. Carlsen, and S. Petersen, “Applications of wireless sensor networks in the oil, gas and resources industries,” pp. 941–948, IEEE.
- [15] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, “Wireless sensor networks: a survey on recent developments and potential synergies,” vol. 68, no. 1, pp. 1–48.

- [16] A. Willig, “Wireless sensor networks: concept, challenges and approaches,” vol. 123, no. 6, pp. 224–231.
- [17] S. Roundy, P. K. Wright, and J. M. Rabaey, “Energy scavenging for wireless sensor networks,” in *Norwell*, pp. 45–47, Springer, 2003.
- [18] Z. Manap, B. M. Ali, C. K. Ng, N. K. Noordin, and A. Sali, “A review on hierarchical routing protocols for wireless sensor networks,” vol. 72, no. 2, pp. 1077–1104.
- [19] “Wireless sensor networks - technology and protocols.”
- [20] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, “MAC essentials for wireless sensor networks,” vol. 12, no. 2, pp. 222–248.
- [21] A. Salhieh, J. Weinmann, M. Kochhal, and L. Schwiebert, “Power efficient topologies for wireless sensor networks,” in *International Conference on Parallel Processing, 2001.*, pp. 156–163, IEEE, 2001.
- [22] A. Chakrabarti, A. Sabharwal, and B. Aazhang, “Using predictable observer mobility for power efficient design of sensor networks,” in *Information Processing in Sensor Networks*, pp. 129–145, Springer, 2003.
- [23] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient mac protocol for wireless sensor networks,” in *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1567–1576, IEEE, 2002.
- [24] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, “Energy-efficient, collision-free medium access control for wireless sensor networks,” *Wireless networks*, vol. 12, no. 1, pp. 63–78, 2006.
- [25] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, “Protocols for self-organization of a wireless sensor network,” *IEEE personal communications*, vol. 7, no. 5, pp. 16–27, 2000.
- [26] K. Akkaya and M. Younis, “A survey on routing protocols for wireless sensor networks,” *Ad hoc networks*, vol. 3, no. 3, pp. 325–349, 2005.
- [27] T. A. Al-Khdour and U. Baroudi, “Literature review of mac, routing and cross layer design protocols for wsn,” in *Wireless Sensor Networks and Energy Efficiency: Protocols, Routing and Management*, pp. 70–118, IGI Global, 2012.
- [28] I. Demirkol, C. Ersoy, and F. Alagoz, “MAC protocols for wireless sensor networks: a survey,” vol. 44, no. 4, pp. 115–121.
- [29] T. Watteyne, A. Molinaro, M. G. Richichi, and M. Dohler, “From MANET to IETF ROLL standardization: A paradigm shift in WSN routing protocols,” vol. 13, no. 4, pp. 688–707.

- [30] Z. Bojkovic and B. Bakmaz, “A survey on wireless sensor networks deployment,” vol. 7, no. 12, pp. 1172–1181.
- [31] R. V. Kulkarni and G. K. Venayagamoorthy, “Particle swarm optimization in wireless-sensor networks: A brief survey,” vol. 41, no. 2, pp. 262–267.
- [32] R. Soua and P. Minet, “A survey on energy efficient techniques in wireless sensor networks,” in *Wireless and Mobile Networking Conference (WMNC), 2011 4th Joint IFIP*, pp. 1–9.
- [33] Z. Gengzhong and L. Qiumei, “A survey on topology control in wireless sensor networks,” pp. 376–380, IEEE.
- [34] S. M. Lasassmeh and J. M. Conrad, “Time synchronization in wireless sensor networks: a survey,” in *IEEE SoutheastCon 2010 (SoutheastCon), Proceedings of the*, pp. 242–245.
- [35] Y.-C. Wu, Q. Chaudhari, and E. Serpedin, “Clock synchronization of wireless sensor networks,” vol. 28, no. 1, pp. 124–138.
- [36] A. D. Wood and J. A. Stankovic, “Denial of service in sensor networks,” *computer*, vol. 35, no. 10, pp. 54–62, 2002.
- [37] H. Sundani, H. Li, V. K. Devabhaktuni, M. Alam, and P. Bhattacharya, “Wireless sensor network simulators a survey and comparisons,” vol. 2, no. 6, pp. 249–265.
- [38] C. P. Singh, O. P. Vyas, and M. K. Tiwari, “A survey of simulation in sensor networks,” pp. 867–872, IEEE.
- [39] W. largest Science, Technology, and Medicine, “Published by,”
- [40] Q. I., “Simulation framework of wireless sensor network (WSN) using MATLAB/SIMULINK software,” in *MATLAB - A Fundamental Tool for Scientific Computing and Engineering Applications - Volume 2*, InTech, sep 2012.
- [41] E. Egea-Lopez, J. Vales-Alonso, A. S. Martinez-Sala, P. Pavon-Marino, and J. García-Haro, “Simulation tools for wireless sensor networks,” in *Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS05)*, p. 24.
- [42] M. Jevtić, N. Zogović, and G. Dimić, “Evaluation of wireless sensor network simulators,” in *Proceedings of the 17th Telecommunications Forum (FOR 2009), Belgrade, Serbia*, pp. 1303–1306.
- [43] K. Ngo, T. Huynh, and D. Thu Huynh, “Simulation wireless sensor networks in castalia,” pp. 39–44, 02 2018.

- [44] R. K. Kodali and V. K. Malothu, "MIXIM framework simulation of WSN with QoS," in *2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, IEEE, may 2016.
- [45] M. Imran, A. M. Said, and H. Hasbullah, "A survey of simulators, emulators and testbeds for wireless sensor networks," in *Information Technology (ITSim), 2010 International Symposium in*, vol. 2, pp. 897–902.
- [46] J. Li and G. Serpen, "TOSSIM simulation of wireless sensor network serving as hardware platform for hopfield neural net configured for max independent set," *Procedia Computer Science*, vol. 6, pp. 408–412, 2011.
- [47] J. Li and G. Serpen, "Simulating heterogeneous and larger-scale wireless sensor networks with tossim tinyos emulator," *Procedia Computer Science*, vol. 12, pp. 374–379, 2012.
- [48] B. Musznicki and P. Zwierzykowski, "Survey of simulators for wireless sensor networks," vol. 5, no. 3, pp. 23–50.
- [49] J. Polley, D. Blazakis, J. McGee, D. Rusk, and J. S. Baras, "ATEMU: a fine-grained sensor network simulator," in *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pp. 145–152.
- [50] A. Kellner, K. Behrends, and D. Hogrefe, "Simulation environments for wireless sensor networks," p. 2.
- [51] L. R. Peralta, L. P. L. de Brito, B. T. Gouveia, D. G. de Sousa, and C. D. S. Alves, "Automatic monitoring and control of museums environment based on wireless sensor networks,"
- [52] L. M. R. Peralta, L. M. Brito, J. P. Santos, J. F. Santos, C. M. Francisco, C. M. Sousa, P. M. Moraes, and F. B. Gouveia, "Environmental monitoring platform based on a heterogeneous wireless sensor network," pp. 26–38.
- [53] T. S. Rappaport *et al.*, *Wireless communications: principles and practice*, vol. 2. prentice hall PTR New Jersey, 1996.

Appendices

A Especificações técnicas

Specifications of the XBee®/XBee-PRO® ZB RF Module

Specification	XBee	XBee-PRO (S2)	XBee-PRO (S2B)
Performance			
Indoor/Urban Range	up to 133 ft. (40 m)	Up to 300 ft. (90 m), up to 200 ft (60 m) international variant	Up to 300 ft. (90 m), up to 200 ft (60 m) international variant
Outdoor RF line-of-sight Range	up to 400 ft. (120 m)	Up to 2 miles (3200 m), up to 5000 ft (1500 m) international variant	Up to 2 miles (3200 m), up to 5000 ft (1500 m) international variant
Transmit Power Output	2mW (+3dBm), boost mode enabled 1.25mW (+1dBm), boost mode disabled	50mW (+17 dBm) 10mW (+10 dBm) for International variant	63mW (+18 dBm) 10mW (+10 dBm) for International variant
RF Data Rate	250,000 bps	250,000 bps	250,000 bps
Data Throughput	up to 35000 bps (see chapter 4)	up to 35000 bps (see chapter 4)	up to 35000 bps (see chapter 4)
Serial Interface Data Rate (software selectable)	1200 bps - 1 Mbps (non-standard baud rates also supported)	1200 bps - 1 Mbps (non-standard baud rates also supported)	1200 bps - 1 Mbps (non-standard baud rates also supported)
Receiver Sensitivity	-96 dBm, boost mode enabled -95 dBm, boost mode disabled	-102 dBm	-102 dBm
Power Requirements			
Supply Voltage	2.1 - 3.6 V	3.0 - 3.4 V	2.7 - 3.6 V
Operating Current (Transmit, max output power)	40mA (@ 3.3 V, boost mode enabled) 35mA (@ 3.3 V, boost mode disabled)	295mA (@3.3 V) 170mA (@3.3 V) international variant	205mA, up to 220 mA with programmable variant (@3.3 V) 117mA, up to 132 mA with programmable variant (@3.3 V), International variant
Operating Current (Receive)	40mA (@ 3.3 V, boost mode enabled) 38mA (@ 3.3 V, boost mode disabled)	45 mA (@3.3 V)	47 mA, up to 62 mA with programmable variant (@3.3 V)
Idle Current (Receiver off)	15mA	15mA	15mA
Power-down Current	< 1 uA @ 25°C	3.5 uA typical @ 25°C	3.5 uA typical @ 25°C
General			
Operating Frequency Band	ISM 2.4 GHz	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960 x 1.297 (2.438cm x 3.294cm)	0.960 x 1.297 (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)	-40 to 85° C (industrial)	-40 to 85° C (industrial)
Antenna Options	Integrated Whip Antenna, Embedded PCB Antenna, RPSMA, or U.FL Connector	Integrated Whip Antenna, Embedded PCB Antenna, RPSMA or U.FL Connector	Integrated Whip Antenna, Embedded PCB Antenna, RPSMA or U.FL Connector
I/O Interface	3.3V CMOS UART (not 5V tolerant), DIO, ADC	3.3V CMOS UART (not 5V tolerant), DIO, ADC	3.3V CMOS UART (not 5V tolerant), DIO, ADC
Networking & Security			
Supported Network Topologies	Point-to-point, Point-to-multipoint, Peer-to-peer, and Mesh	Point-to-point, Point-to-multipoint, Peer-to-peer, and Mesh	Point-to-point, Point-to-multipoint, Peer-to-peer, and Mesh
Number of Channels	16 Direct Sequence Channels	14 Direct Sequence Channels	15 Direct Sequence Channels
Channels	11 to 26	11 to 24	11 to 25
Addressing Options	PAN ID and Addresses, Cluster IDs and Endpoints (optional)	PAN ID and Addresses, Cluster IDs and Endpoints (optional)	PAN ID and Addresses, Cluster IDs and Endpoints (optional)
Agency Approvals			
United States (FCC Part 15.247)	FCC ID: OUR-XBEE2	FCC ID: MCQ-XBEEPRO2	FCC ID: MCQ-PROS2B

Figura 24: Especificações técnicas: Xbee S2 e XBee PRO(S2B)

B Plantas do museu

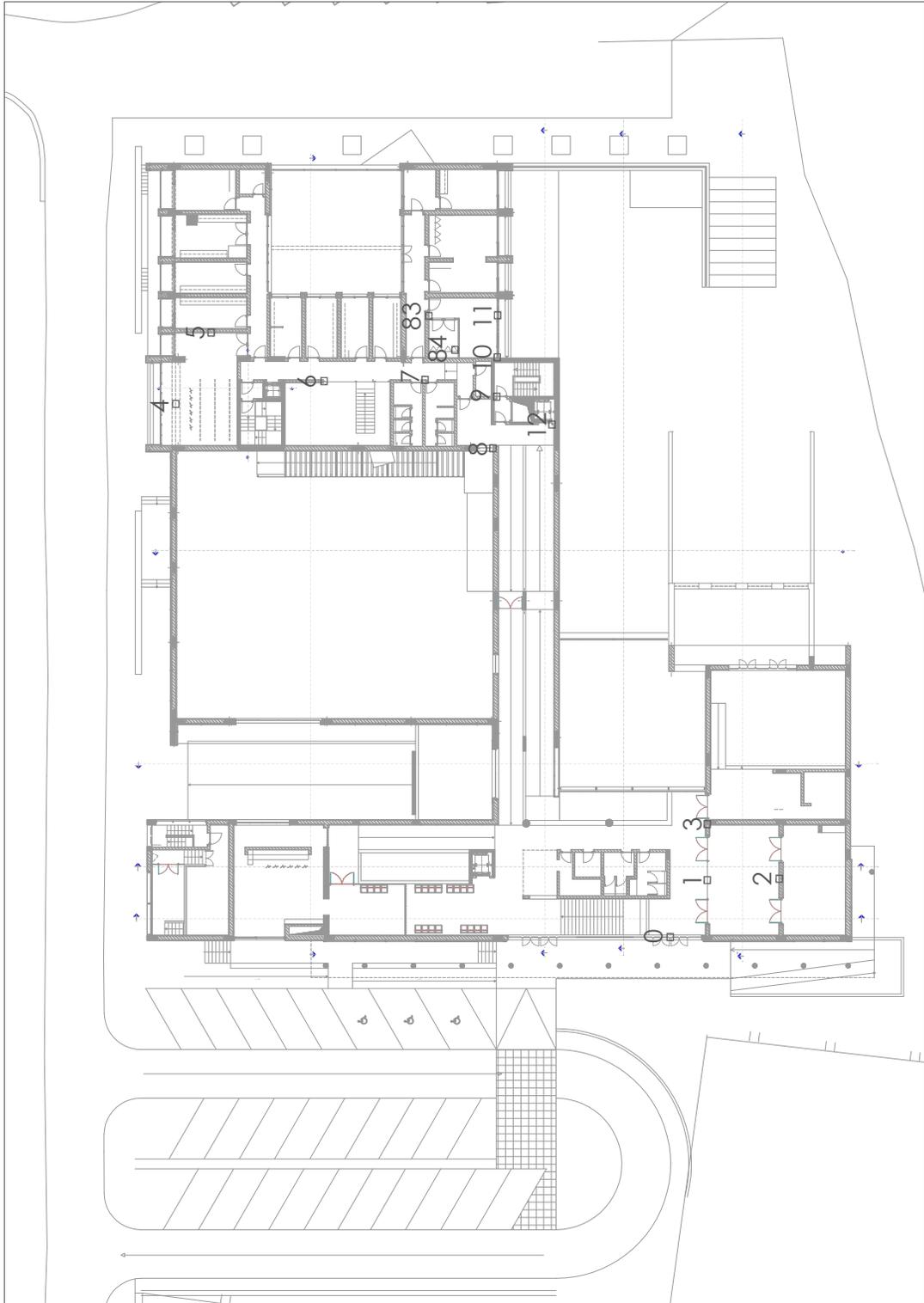


Figura 25: Planta do Piso 2

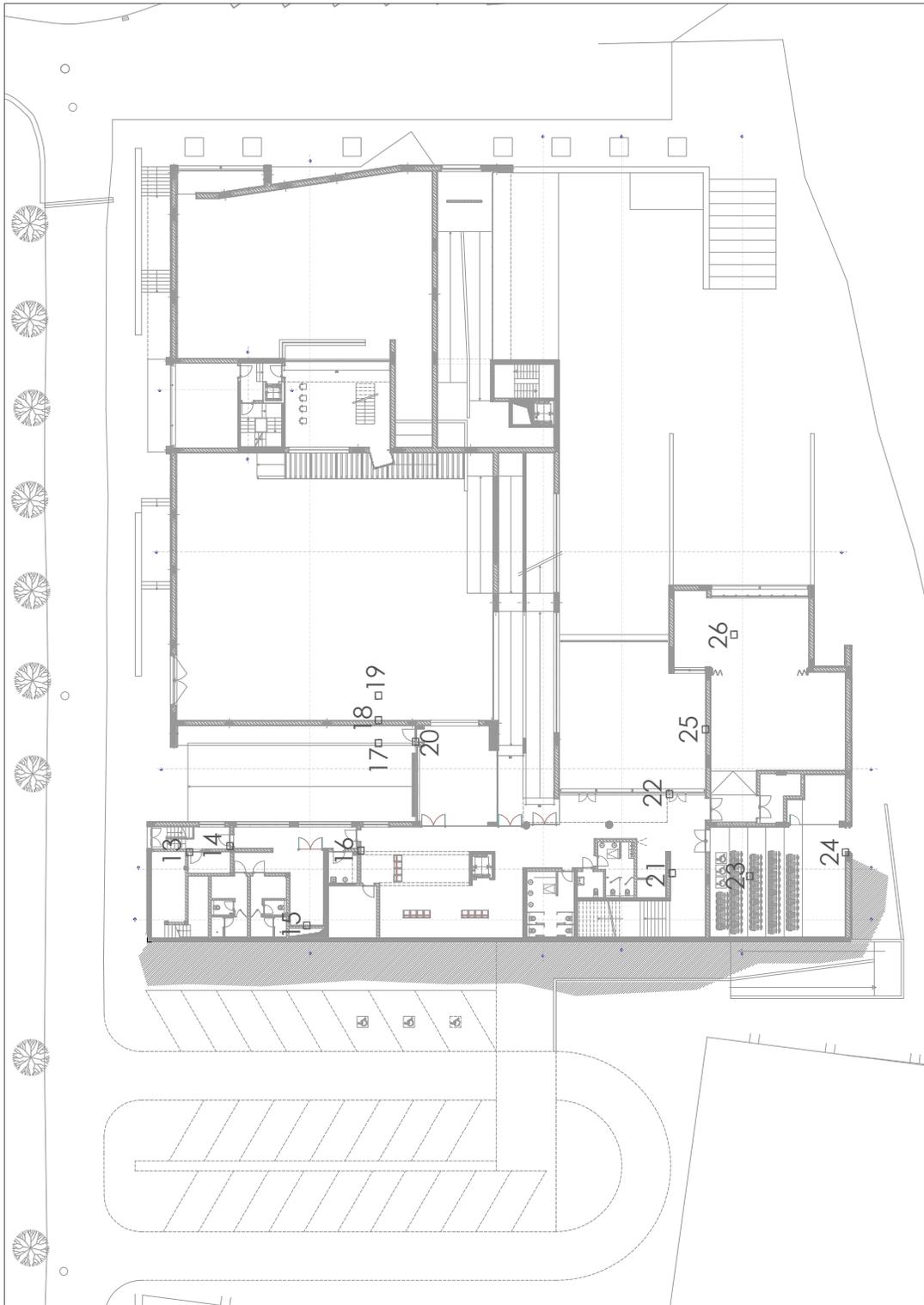


Figura 26: Planta do Piso 1

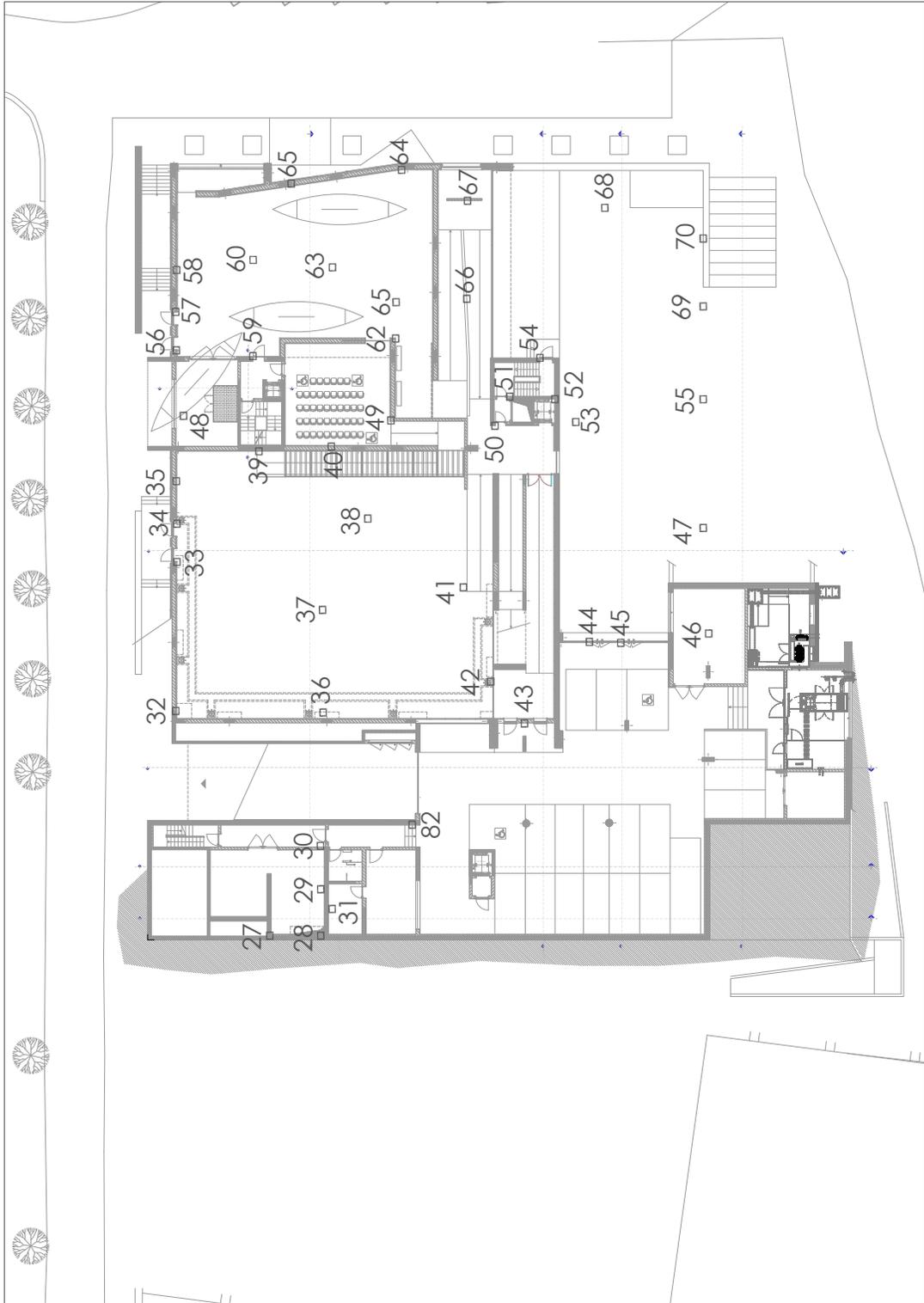


Figura 27: Planta do Piso 0

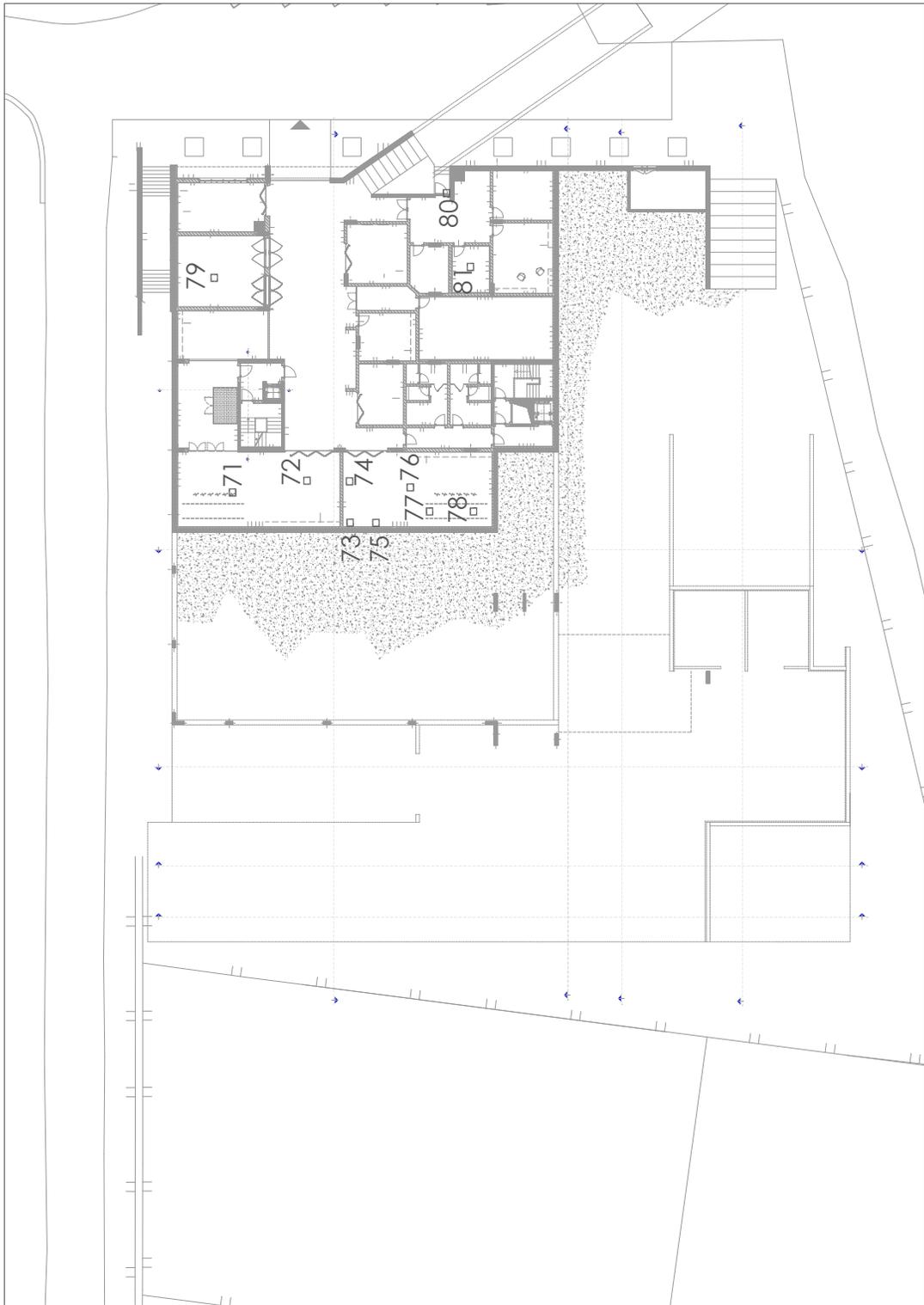


Figura 28: Planta do Piso -1

C Ficheiro de configuração

```
[General]
cmdenv-express-mode = true
record-eventlog = true
network = Wsn
sim-time-limit = 300 min

#####
#                               Simulation parameters
#####
tkenv-default-config = teste

**.**.coreDebug = false

**.playgroundSizeX = 65m
**.playgroundSizeY = 60m
**.playgroundSizeZ = 25m
**.numHosts = 85

**.drawMaxIntfDist = false

#####
#                               WorldUtility parameters
#####
**.world.useTorus = false

#####
#                               channel parameters
#####
**.connectionManager.sendDirect = false
**.connectionManager.pMax = 63mW
**.connectionManager.sat = -102dBm
**.connectionManager.alpha = 5.04
**.connectionManager.carrierFrequency = 2.412e+9Hz

#####
#           Parameters for the Host
#####
**.node[*].nicType = "Nic802154_TI_CC2420"

##### PhyLayer parameters #####
**.node[*].nic.phy.usePropagationDelay = true
**.node[*].nic.phy.thermalNoise = -100dBm
**.node[*].nic.phy.useThermalNoise = true
```

```
**node[*].nic.phy.analogueModels = xmldoc("LNS_config.xml")
***node[*].nic.phy.analogueModels = xmldoc("SPL_config.xml")
# Portas: -96dBm
# Restantes: -102dBm
**node[0].nic.phy.sensitivity = -96dBm
**node[8].nic.phy.sensitivity = -96dBm
**node[13].nic.phy.sensitivity = -96dBm
**node[14].nic.phy.sensitivity = -96dBm
**node[20].nic.phy.sensitivity = -96dBm
**node[22].nic.phy.sensitivity = -96dBm
**node[24].nic.phy.sensitivity = -96dBm
**node[33].nic.phy.sensitivity = -96dBm
**node[34].nic.phy.sensitivity = -96dBm
**node[43].nic.phy.sensitivity = -96dBm
**node[44].nic.phy.sensitivity = -96dBm
**node[45].nic.phy.sensitivity = -96dBm
**node[50].nic.phy.sensitivity = -96dBm
**node[54].nic.phy.sensitivity = -96dBm
**node[56].nic.phy.sensitivity = -96dBm
**node[57].nic.phy.sensitivity = -96dBm
**node[59].nic.phy.sensitivity = -96dBm
**node[80].nic.phy.sensitivity = -96dBm
**node[*].nic.phy.sensitivity = -102dBm
# Portas: 2.0mW
# Restantes: 63.0mW
**node[0].nic.phy.maxTXPower = 2.0mW
**node[8].nic.phy.maxTXPower = 2.0mW
**node[13].nic.phy.maxTXPower = 2.0mW
**node[14].nic.phy.maxTXPower = 2.0mW
**node[20].nic.phy.maxTXPower = 2.0mW
**node[22].nic.phy.maxTXPower = 2.0mW
**node[24].nic.phy.maxTXPower = 2.0mW
**node[33].nic.phy.maxTXPower = 2.0mW
**node[34].nic.phy.maxTXPower = 2.0mW
**node[43].nic.phy.maxTXPower = 2.0mW
**node[44].nic.phy.maxTXPower = 2.0mW
**node[45].nic.phy.maxTXPower = 2.0mW
**node[50].nic.phy.maxTXPower = 2.0mW
**node[54].nic.phy.maxTXPower = 2.0mW
**node[56].nic.phy.maxTXPower = 2.0mW
**node[57].nic.phy.maxTXPower = 2.0mW
**node[59].nic.phy.maxTXPower = 2.0mW
**node[80].nic.phy.maxTXPower = 2.0mW
**node[*].nic.phy.maxTXPower = 63.0mW
```

```

##### MAC layer parameters #####
**.node[*].nic.mac.queueLength = 5
**.node[*].nic.mac.headerLength = 24bit
# Portas: 2.0mW
# Restantes: 63.0mW
**.node[0].nic.mac.txPower = 2.0mW
**.node[8].nic.mac.txPower = 2.0mW
**.node[13].nic.mac.txPower = 2.0mW
**.node[14].nic.mac.txPower = 2.0mW
**.node[20].nic.mac.txPower = 2.0mW
**.node[22].nic.mac.txPower = 2.0mW
**.node[24].nic.mac.txPower = 2.0mW
**.node[33].nic.mac.txPower = 2.0mW
**.node[34].nic.mac.txPower = 2.0mW
**.node[43].nic.mac.txPower = 2.0mW
**.node[44].nic.mac.txPower = 2.0mW
**.node[45].nic.mac.txPower = 2.0mW
**.node[50].nic.mac.txPower = 2.0mW
**.node[54].nic.mac.txPower = 2.0mW
**.node[56].nic.mac.txPower = 2.0mW
**.node[57].nic.mac.txPower = 2.0mW
**.node[59].nic.mac.txPower = 2.0mW
**.node[80].nic.mac.txPower = 2.0mW
**.node[*].nic.mac.txPower = 63.0mW
**.node[*].nic.mac.stats = true
**.node[*].nic.mac.trace = true

**.node[*].nic.mac.ccaDetectionTime = 0.0005s
**.node[*].nic.mac.aTurnaroundTime = 0.000192s #no radio switch times
**.node[*].nic.mac.rxSetupTime = 0s #no radio switch times

**.node[*].nic.mac.backoffMethod = "linear"
**.node[*].nic.mac.macMaxCSMABackoffs = 14
**.node[*].nic.mac.contentionWindow = 20
**.node[*].nic.mac.aUnitBackoffPeriod = 0.04s

# MAC Ack settings (disabled)
**.node[*].nic.mac.useMACAcks = false
**.node[*].nic.mac.ackLength = 0bit
**.node[*].nic.mac.macMaxFrameRetries = 0
**.node[*].nic.mac.macAckWaitDuration = 0s
**.node[*].nic.mac.sifs = 0s

**.node[*].nic.mac.macMinBE = 0 #only used for exponential backoffs
**.node[*].nic.mac.macMaxBE = 0 #only used for exponential backoffs

```

```

##### Application layer parameters #####
**.node[0].appl.nbPackets = 0
**.node[*].appl.nbPackets = 0
**.node[*].appl.destAddr = 0
**.node[*].appl.initializationTime = 3 s
**.node[*].appl.trafficType = "periodic"

##### NETW layer parameters #####
**.node[*].networkType = "WiseRoute"
**.node[84].netwl.routeFloodsInterval = 1200 s
**.node[*].netwl.sinkAddress = 84
**.node[*].netwl.trace = true
**.node[*].netwl.stats = true
**.node[*].netwl.headerLength = 24 bit
# RSSI threshold for route selection
**.node[*].netwl.rssiThreshold = -50 dBm

##### Mobility #####
**.node[*].mobilityType = "StationaryMobility"
**.node[*].mobility.debug = false
**.node[*].mobility.initFromDisplayString = false

##### Nodes #####
##### Sink #####
# Node: 84
**.node[84].mobility.initialX = 49m
**.node[84].mobility.initialY = 24m
**.node[84].mobility.initialZ = 18m
##### Sensores Verdes #####
# Node: 1
**.node[1].mobility.initialX = 5m
**.node[1].mobility.initialY = 46m
**.node[1].mobility.initialZ = 21m
# Node: 2
**.node[2].mobility.initialX = 5m
**.node[2].mobility.initialY = 52m
**.node[2].mobility.initialZ = 21m
# Node: 4
**.node[4].mobility.initialX = 44m
**.node[4].mobility.initialY = 2m
**.node[4].mobility.initialZ = 18m
# Node: 5
**.node[5].mobility.initialX = 50m
**.node[5].mobility.initialY = 5m

```

```
**node[5].mobility.initialZ = 18m
# Node: 16
**node[16].mobility.initialX = 8m
**node[16].mobility.initialY = 18m
**node[16].mobility.initialZ = 16m
# Node: 21
**node[21].mobility.initialX = 6m
**node[21].mobility.initialY = 43m
**node[21].mobility.initialZ = 16m
# Node: 23
**node[23].mobility.initialX = 5m
**node[23].mobility.initialY = 49m
**node[23].mobility.initialZ = 21m
# Node: 25
**node[25].mobility.initialX = 17m
**node[25].mobility.initialY = 46m
**node[25].mobility.initialZ = 16m
# Node: 26
**node[26].mobility.initialX = 25m
**node[26].mobility.initialY = 48m
**node[26].mobility.initialZ = 16m
# Node: 31
**node[31].mobility.initialX = 3m
**node[31].mobility.initialY = 15m
**node[31].mobility.initialZ = 14m
# Node: 32
**node[32].mobility.initialX = 19m
**node[32].mobility.initialY = 2m
**node[32].mobility.initialZ = 14m
# Node: 35
**node[35].mobility.initialX = 38m
**node[35].mobility.initialY = 2m
**node[35].mobility.initialZ = 14m
# Node: 36
**node[36].mobility.initialX = 18m
**node[36].mobility.initialY = 14m
**node[36].mobility.initialZ = 14m
# Node: 37
**node[37].mobility.initialX = 27m
**node[37].mobility.initialY = 14m
**node[37].mobility.initialZ = 14m
# Node: 38
**node[38].mobility.initialX = 35m
**node[38].mobility.initialY = 18m
**node[38].mobility.initialZ = 14m
```

```
# Node: 39
**.node[39].mobility.initialX = 40m
**.node[39].mobility.initialY = 9m
**.node[39].mobility.initialZ = 14m
# Node: 40
**.node[40].mobility.initialX = 40m
**.node[40].mobility.initialY = 15m
**.node[40].mobility.initialZ = 14m
# Node: 41
**.node[41].mobility.initialX = 29m
**.node[41].mobility.initialY = 25m
**.node[41].mobility.initialZ = 14m
# Node: 42
**.node[42].mobility.initialX = 21m
**.node[42].mobility.initialY = 28m
**.node[42].mobility.initialZ = 14m
# Node: 46
**.node[46].mobility.initialX = 25m
**.node[46].mobility.initialY = 46m
**.node[46].mobility.initialZ = 14m
# Node: 47
**.node[47].mobility.initialX = 34m
**.node[47].mobility.initialY = 45m
**.node[47].mobility.initialZ = 14m
# Node: 48
**.node[48].mobility.initialX = 43m
**.node[48].mobility.initialY = 3m
**.node[48].mobility.initialZ = 14m
# Node: 49
**.node[49].mobility.initialX = 43m
**.node[49].mobility.initialY = 20m
**.node[49].mobility.initialZ = 14m
# Node: 53
**.node[53].mobility.initialX = 42m
**.node[53].mobility.initialY = 35m
**.node[53].mobility.initialZ = 14m
# Node: 55
**.node[55].mobility.initialX = 45m
**.node[55].mobility.initialY = 46m
**.node[55].mobility.initialZ = 14m
# Node: 58
**.node[58].mobility.initialX = 55m
**.node[58].mobility.initialY = 2m
**.node[58].mobility.initialZ = 14m
# Node: 60
```

```
**node[60].mobility.initialX = 56m
**node[60].mobility.initialY = 9m
**node[60].mobility.initialZ = 14m
# Node: 61
**node[61].mobility.initialX = 62m
**node[61].mobility.initialY = 12m
**node[61].mobility.initialZ = 14m
# Node: 62
**node[62].mobility.initialX = 49m
**node[62].mobility.initialY = 20m
**node[62].mobility.initialZ = 14m
# Node: 63
**node[63].mobility.initialX = 55m
**node[63].mobility.initialY = 15m
**node[63].mobility.initialZ = 14m
# Node: 64
**node[64].mobility.initialX = 63m
**node[64].mobility.initialY = 21m
**node[64].mobility.initialZ = 14m
# Node: 65
**node[65].mobility.initialX = 52m
**node[65].mobility.initialY = 20m
**node[65].mobility.initialZ = 14m
# Node: 66
**node[66].mobility.initialX = 53m
**node[66].mobility.initialY = 26m
**node[66].mobility.initialZ = 14m
# Node: 67
**node[67].mobility.initialX = 60m
**node[67].mobility.initialY = 26m
**node[67].mobility.initialZ = 14m
# Node: 68
**node[68].mobility.initialX = 60m
**node[68].mobility.initialY = 37m
**node[68].mobility.initialZ = 14m
# Node: 69
**node[69].mobility.initialX = 52m
**node[69].mobility.initialY = 46m
**node[69].mobility.initialZ = 14m
# Node: 70
**node[70].mobility.initialX = 58m
**node[70].mobility.initialY = 45m
**node[70].mobility.initialZ = 14m
# Node: 71
**node[71].mobility.initialX = 37m
```

```

**.node[71].mobility.initialY = 7m
**.node[71].mobility.initialZ = 8m
# Node: 72
**.node[72].mobility.initialX = 38m
**.node[72].mobility.initialY = 13m
**.node[72].mobility.initialZ = 8m
# Node: 74
**.node[74].mobility.initialX = 38m
**.node[74].mobility.initialY = 17m
**.node[74].mobility.initialZ = 8m
# Node: 76
**.node[76].mobility.initialX = 37m
**.node[76].mobility.initialY = 22m
**.node[76].mobility.initialZ = 8m
# Node: 83
**.node[83].mobility.initialX = 51m
**.node[83].mobility.initialY = 23m
**.node[83].mobility.initialZ = 18m

##### Sensores Azuis #####
# Node: 73
**.node[73].mobility.initialX = 34m
**.node[73].mobility.initialY = 17m
**.node[73].mobility.initialZ = 8m
# Node: 75
**.node[75].mobility.initialX = 34m
**.node[75].mobility.initialY = 19m
**.node[75].mobility.initialZ = 8m
# Node: 79
**.node[79].mobility.initialX = 54m
**.node[79].mobility.initialY = 5m
**.node[79].mobility.initialZ = 8m

##### Sensores Vermelhos #####
# Node: 0
**.node[0].mobility.initialX = 0m
**.node[0].mobility.initialY = 43m
**.node[0].mobility.initialZ = 21m
# Node: 8
**.node[8].mobility.initialX = 40m
**.node[8].mobility.initialY = 28m
**.node[8].mobility.initialZ = 18m
# Node: 13
**.node[13].mobility.initialX = 7m
**.node[13].mobility.initialY = 3m

```

```
**node[13].mobility.initialZ = 16m
# Node: 14
**node[14].mobility.initialX = 8m
**node[14].mobility.initialY = 7m
**node[14].mobility.initialZ = 16m
# Node: 20
**node[20].mobility.initialX = 16m
**node[20].mobility.initialY = 22m
**node[20].mobility.initialZ = 16m
# Node: 22
**node[22].mobility.initialX = 12m
**node[22].mobility.initialY = 43m
**node[22].mobility.initialZ = 16m
# Node: 24
**node[24].mobility.initialX = 7m
**node[24].mobility.initialY = 57m
**node[24].mobility.initialZ = 15m
# Node: 33
**node[33].mobility.initialX = 31m
**node[33].mobility.initialY = 2m
**node[33].mobility.initialZ = 14m
# Node: 34
**node[34].mobility.initialX = 34m
**node[34].mobility.initialY = 2m
**node[34].mobility.initialZ = 14m
# Node: 43
**node[43].mobility.initialX = 18m
**node[43].mobility.initialY = 31m
**node[43].mobility.initialZ = 14m
# Node: 44
**node[44].mobility.initialX = 24m
**node[44].mobility.initialY = 36m
**node[44].mobility.initialZ = 14m
# Node: 45
**node[45].mobility.initialX = 24m
**node[45].mobility.initialY = 39m
**node[45].mobility.initialZ = 14m
# Node: 50
**node[50].mobility.initialX = 42m
**node[50].mobility.initialY = 28m
**node[50].mobility.initialZ = 14m
# Node: 54
**node[54].mobility.initialX = 48m
**node[54].mobility.initialY = 32m
**node[54].mobility.initialZ = 14m
```

```
# Node: 56
**.node[56].mobility.initialX = 48m
**.node[56].mobility.initialY = 2m
**.node[56].mobility.initialZ = 14m
# Node: 57
**.node[57].mobility.initialX = 52m
**.node[57].mobility.initialY = 2m
**.node[57].mobility.initialZ = 14m
# Node: 59
**.node[59].mobility.initialX = 48m
**.node[59].mobility.initialY = 8m
**.node[59].mobility.initialZ = 14m
# Node: 80
**.node[80].mobility.initialX = 61m
**.node[80].mobility.initialY = 21m
**.node[80].mobility.initialZ = 8m

##### Sensores Pretos #####
# Node: 9
**.node[9].mobility.initialX = 44m
**.node[9].mobility.initialY = 29m
**.node[9].mobility.initialZ = 18m
# Node: 10
**.node[10].mobility.initialX = 48m
**.node[10].mobility.initialY = 29m
**.node[10].mobility.initialZ = 18m
# Node: 11
**.node[11].mobility.initialX = 51m
**.node[11].mobility.initialY = 29m
**.node[11].mobility.initialZ = 18m
# Node: 12
**.node[12].mobility.initialX = 42m
**.node[12].mobility.initialY = 33m
**.node[12].mobility.initialZ = 18m
# Node: 15
**.node[15].mobility.initialX = 1m
**.node[15].mobility.initialY = 13m
**.node[15].mobility.initialZ = 16m
# Node: 17
**.node[17].mobility.initialX = 16m
**.node[17].mobility.initialY = 19m
**.node[17].mobility.initialZ = 16m
# Node: 18
**.node[18].mobility.initialX = 18m
**.node[18].mobility.initialY = 19m
```

```

**.node[18].mobility.initialZ = 16m
# Node: 19
**.node[19].mobility.initialX = 20m
**.node[19].mobility.initialY = 19m
**.node[19].mobility.initialZ = 16m
# Node: 27
**.node[27].mobility.initialX = 0m
**.node[27].mobility.initialY = 10m
**.node[27].mobility.initialZ = 14m
# Node: 28
**.node[28].mobility.initialX = 0m
**.node[28].mobility.initialY = 14m
**.node[28].mobility.initialZ = 14m
# Node: 29
**.node[29].mobility.initialX = 4m
**.node[29].mobility.initialY = 14m
**.node[29].mobility.initialZ = 14m
# Node: 30
**.node[30].mobility.initialX = 8m
**.node[30].mobility.initialY = 14m
**.node[30].mobility.initialZ = 14m
# Node: 51
**.node[51].mobility.initialX = 44m
**.node[51].mobility.initialY = 30m
**.node[51].mobility.initialZ = 14m
# Node: 52
**.node[52].mobility.initialX = 44m
**.node[52].mobility.initialY = 33m
**.node[52].mobility.initialZ = 14m
# Node: 77
**.node[77].mobility.initialX = 35m
**.node[77].mobility.initialY = 23m
**.node[77].mobility.initialZ = 8m
# Node: 78
**.node[78].mobility.initialX = 35m
**.node[78].mobility.initialY = 27m
**.node[78].mobility.initialZ = 8m
# Node: 81
**.node[81].mobility.initialX = 55m
**.node[81].mobility.initialY = 26m
**.node[81].mobility.initialZ = 8m

##### Outros Sensores #####
# Node: 3
**.node[3].mobility.initialX = 10m

```

```
**node[3].mobility.initialY = 46m
**node[3].mobility.initialZ = 21m
# Node: 6
**node[6].mobility.initialX = 46m
**node[6].mobility.initialY = 15m
**node[6].mobility.initialZ = 18m
# Node: 7
**node[7].mobility.initialX = 46m
**node[7].mobility.initialY = 23m
**node[7].mobility.initialZ = 18m
# Node: 82
**node[82].mobility.initialX = 9m
**node[82].mobility.initialY = 22m
**node[82].mobility.initialZ = 14m
```

Listing 10: Template base para a WSN do Museu da Baleia

D MiXiM

Introdução

De uma forma sucinta e simplificada, o processo de transmissão e receção de comunicações é tratado do seguinte modo pelo simulador:

Na camada MAC do nó que emite o pacote, é adicionado a este a potência de transmissão, passando-o posteriormente para a camada física. Nesta, no módulo *connection manager* são criadas e enviadas cópias do pacote para todos os nós dentro da distância de interferência. Para cada um destes nós recetores, na respetiva camada física, são aplicados os modelos analógicos implementados (configurados pelo ficheiro .xml). Cada um destes gera o valor da atenuação causado por esse modelo e adiciona-o ao pacote. Finalmente, é passado para o módulo *decider* onde a potência do sinal na receção é calculada, com o recurso à potência de transmissão e às atenuações previamente adicionadas, decidindo-se depois se o pacote é ou não recebido.

De uma forma simplificada o modelo analógico é o responsável por mudar o valor da atenuação do sinal para simular propriedades como *shadowing*, *fading*, *pathloss* ou obstáculos. O *decider*¹⁹ tem como principal função a de decidir quais os pacotes recebidos que deverão ser passados para a camada MAC, tendo também a função de determinar se o meio de comunicação está livre ou ocupado num determinado momento ou intervalo de tempo. Por sua vez, o *Connection Manager* controla os parâmetros usados para definir a distância máxima sobre a qual a transmissão de um nó pode causar interferência noutros nós.

Connection Manager:

```
**.connectionManager.sendDirect = false
**.connectionManager.pMax = 100mW
**.connectionManager.sat = -84dBm
**.connectionManager.alpha = 5.0
**.connectionManager.carrierFrequency = 2.412e+9Hz
```

Listing 11: Connection Manager

¹⁹<http://mixim.sourceforge.net/doc-2.1/MiXiM/doc/doxy/a00070.html>

E Análise estatística à escalabilidade do *Wiseroute*

Para tentar analisar esta limitação, foi feito um pequeno estudo para testar a escalabilidade utilizando o modelo *Log Normal Shadowing*. Este consistiu na utilização do protocolo Wiseroute, progressivamente para 8, 16, 32, 64 e 128 nós, dispostos, num padrão circular à volta da estação. Todos eles estavam a uma distância de 75 metros desta, e foram inseridos entre os nós da iteração anterior, como disposto na Figura 29.

Para este estudo em particular, foram utilizados os parâmetros dispostos na listagem 12, o que, de acordo com a fórmula ??, proporciona uma distância de comunicação aproximada a 135 m.

```
**.connectionManager.alpha = 3.0
**.connectionManager.carrierFrequency = 2.412e+9Hz
...
**.node[*].nic.phy.sensitivity = -84dBm
...
**.node[*].nic.mac.txPower = 100mW
```

Listing 12: Parâmetros referentes à distância de comunicação

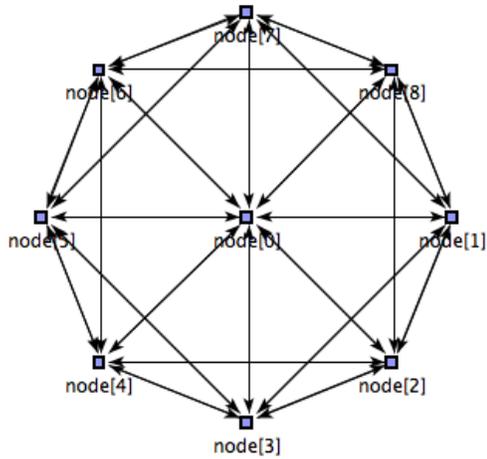
Cada simulação foi então executada num período de 1 hora (tempo simulado), que, com o broadcast do Wiseroute definido para 1200 segundos, equivalia ao envio de 3 broadcasts por parte do nó 0. Este valor foi escolhido pois considerou-se que, na realidade, esperar mais do que uma hora para todos os nós obterem esta rota pode ser excessivo.

Para a iteração de 128 nós, a mais penalizadora em termos de interferências devido aos nós vizinhos, a mesma experiência foi repetida, com diferentes seeds, 100 vezes, o que permite também eliminar qualquer efeito particular desta simulação.

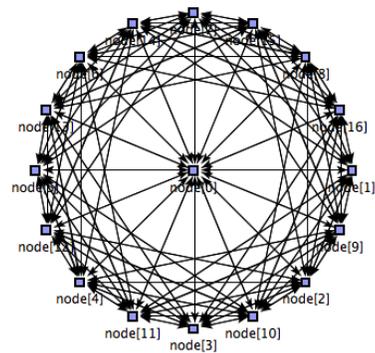
Os resultados foram os esperados. Mesmo para os 128 nós dispostos de forma equidistante da estação, não houve qualquer problema com a criação de uma rota para esta, conforme disposto na figura 30.

O processo foi então repetido, agora com os nós dispostos em dois círculos, de modo a promover saltos em direção à estação. A figura 31 mostra as distâncias relativas entre a estação (nó 0), e dois outros nós, um pertencente ao círculo interior (nó 1) e o outro, ao exterior (nó 2). Na mesma figura está também representada a distância máxima de comunicação calculada anteriormente.

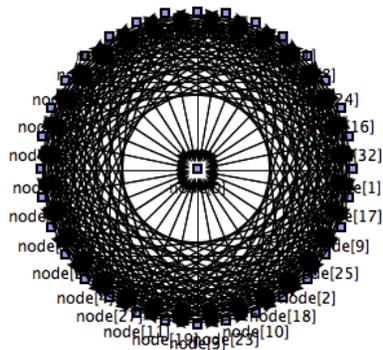
Ao realizar a simulação para 16 nós, seguindo a disposição da figura 32b, obtiveram-se resultados inesperados, mas em concordância com simulações anteriores. Repetiu-se então a experiência 100 vezes e obtiveram-se os resultados expostos na figura 33. Desta, observa-se que em apenas 23 das repetições a totalidade dos nós conseguiu criar uma rota para estação. De realçar que os nós em falha, são os pertencentes ao círculo exterior. Realizando a mesma experiência para 32 nós, observam-se resultados mais extremos, sendo que nenhum dos nós pertencentes ao círculo exterior é capaz de criar a devida rota.



(a) 8 nós



(b) 16 nós



(c) 32 nós

Figura 29: Disposição dos nós em torno da estação para as três primeiras iterações

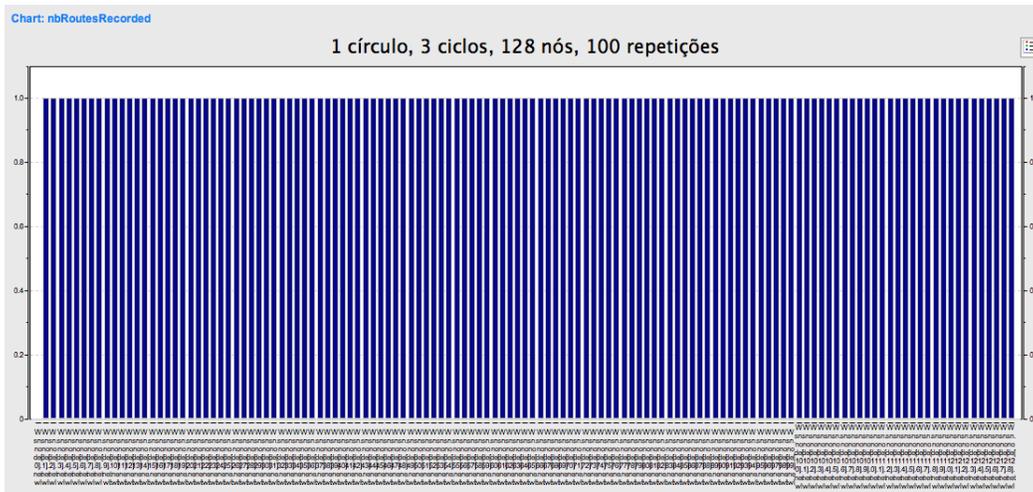


Figura 30: Teste com nós dispostos em círculo, durante 3 ciclos.

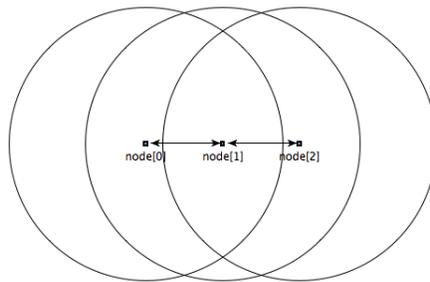
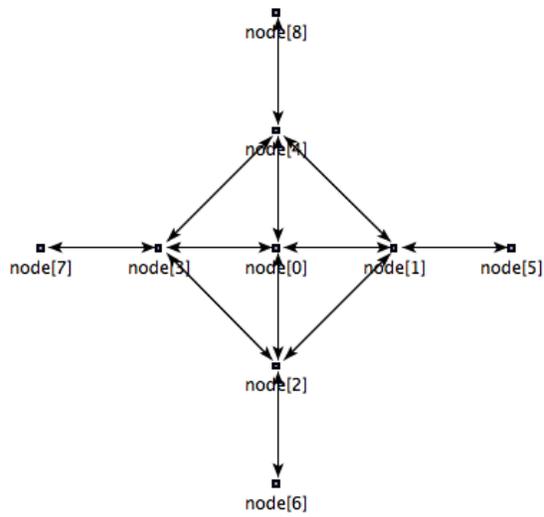
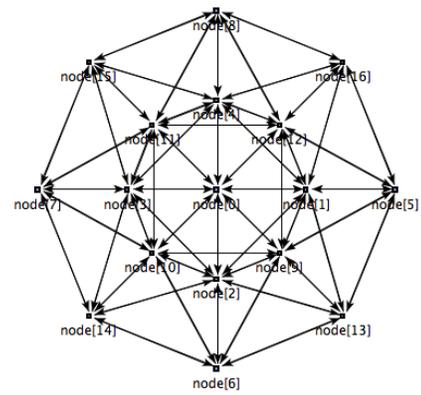


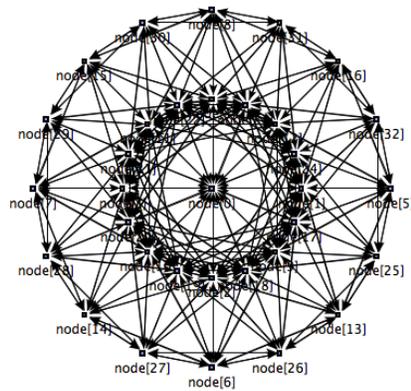
Figura 31: Recepção de pacotes para a criação de rotas



(a) 8 nós



(b) 16 nós



(c) 32 nós

Figura 32: Disposição dos nós em torno da estação para as duas primeiras iterações

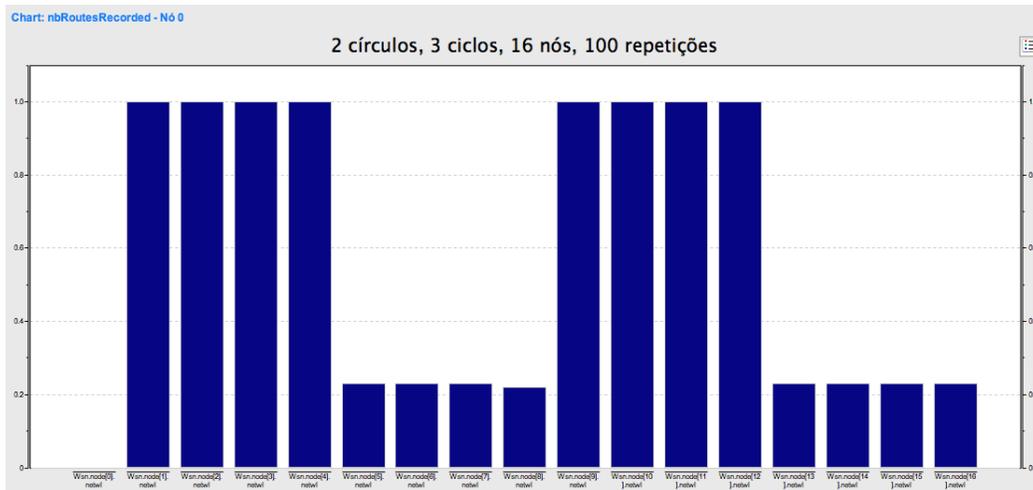


Figura 33: Teste com nós dispostos em dois círculos, durante 3 círculos.

F Resultados: simulações com 5 pacotes

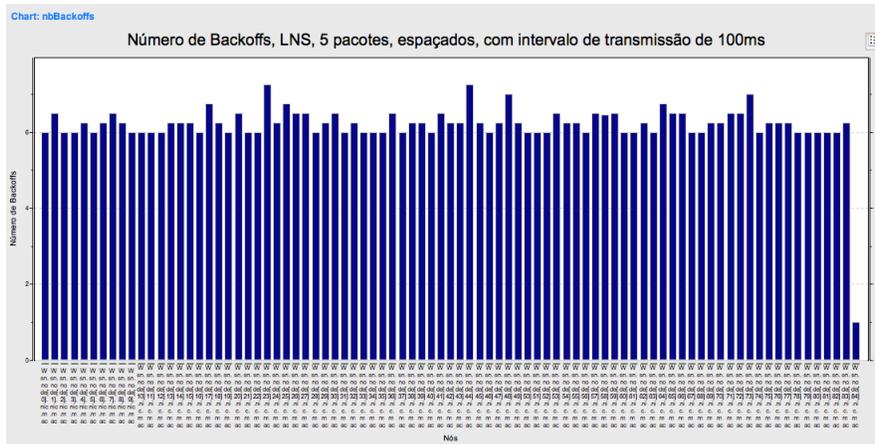


Figura 34: Uma repetição

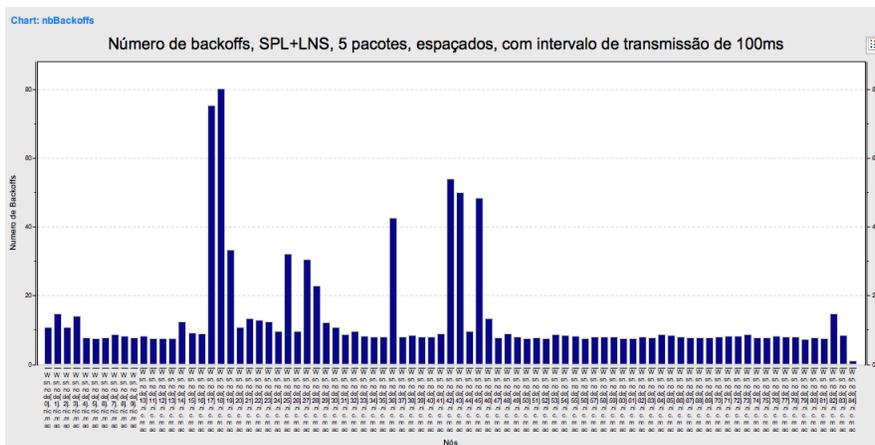


Figura 35: Vinte e cinco repetições

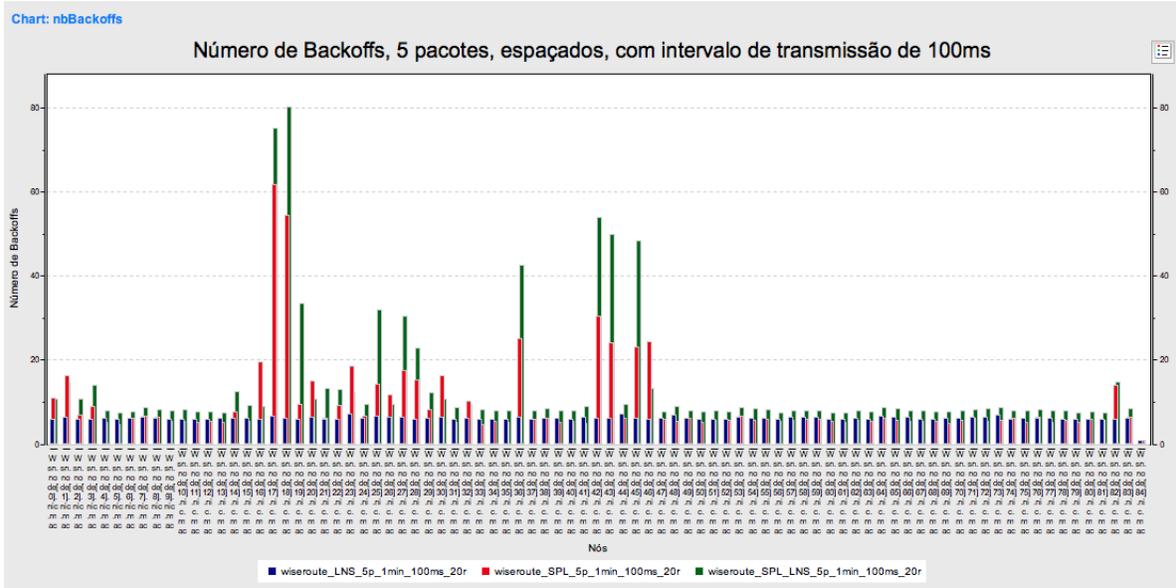


Figura 36: 5 pacotes espaçados em um minuto.

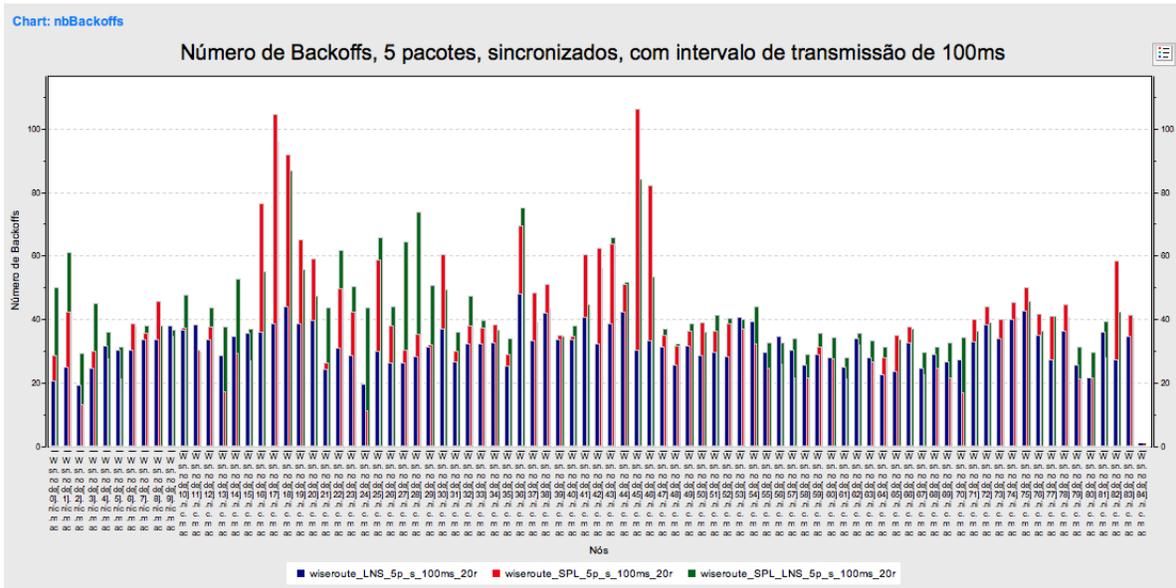


Figura 37: 5 pacotes sincronizados