Departamento de Engenharia Eletrotécnica

Faculdade de Ciências e Tecnologia

Universidade de Coimbra

Comunicação Ad Hoc em Equipas de Robôs Móveis Utilizando a Tecnologia ZigBee

Amadeu Socorro Lopes Fernandes

Dissertação para obtenção do grau de Mestre em

Engenharia Eletrotécnica e de Computadores

Setembro de 2012

FCTUC

Departamento de Engenharia Eletrotécnica

Faculdade de Ciências e Tecnologia

Universidade de Coimbra

Comunicação Ad Hoc em Equipas de Robôs Móveis Utilizando a Tecnologia ZigBee

Orientador:

Prof. Doutor Rui Rocha

Co-Orientadores:

Eng.º David Portugal

Eng.º Micael Couceiro

Júri:

Prof. Doutor Paulo Jorge Carvalho Menezes

Prof. Doutor Luís Alberto da Silva Cruz

Dissertação para obtenção do grau de Mestre em

Engenharia Eletrotécnica e de Computadores

Setembro de 2012

Agradecimentos

Embora me subscrevo como autor desta dissertação muitas pessoas deram o seu contributo para que a sua realização fosse possível. A essas pessoas quero exprimir os meus sinceros agradecimentos.

Em primeiro lugar um agredecimento especial ao Professor Doutor Rui Rocha na minha orientação, pela atenção, rigor, organização e confiança depositada em mim, durante a realização deste trabalho.

Agradeço também aos meus co-orientadores David Portugal e Micael Couceiro pela estimulação constante, disponibilidade e a boa receção e integração no grupo de trabalho.

Quero também agradecer ao Instituto de Sistemas e Robótica, pelas ótimas condições disponibilizadas, bem como a simpatia dos profissionais desta instituição.

Aos meus pais e meus irmãos, agradeço profundamente pela dedicação, pelo amor, pelo sacrifício e pela confiança que sempre depositaram em mim. Quero agradecer a toda a minha família, tios, avós, primos e à minha namorada Ângela Cruz, não podendo esquecer de todos os meus amigos pelo apoio nos momentos mais difíceis da vida. À minha tia Margarida Veiga Lopes e Tadeu Veiga um muito obrigado pela contribuição fundamental que tiveram, para que a minha formação fosse possível.

Um obrigado muito especial, pelo tempo disponibilizado e atenção por parte das pessoas que contribuíram para a realização deste trabalho, André Araújo (MRL, ISR), Arlindo Veiga (IT), Sónia Semedo, Tatiana Chantre, Fredilson Fortes, Yvan Ramos e todas as outras que indiretamente deram a sua contribuição.

Por todos os momentos passados, pela experiência de uma nova cultura, e principalmente pelo conhecimento adquirido que me acompanhará ao longo da minha vida, obrigado Coimbra, obrigado Portugal.

Resumo

Os robôs móveis TraxBot desenvolvidos no ISR Coimbra possuem módulos XBee que utilizam a tecnologia ZigBee como protocolo de rede, possibilitando a comunicação inter-robô. Estes módulos acoplam-se diretamente à placa de processamento de baixo nível dos robôs, o Arduino Uno. O objetivo deste trabalho consiste em implementar a comunicação *ad hoc* sem fios entre membros de uma equipa de robôs móveis, utilizando a tecnologia ZigBee, fazendo a integração e o desenvolvimento de funcionalidades do módulo XBee OEM RF através das funções série *standard* do Arduino. Este trabalho proporciona uma ferramenta útil de investigação, possibilitando a interação e cooperação de uma equipa de robôs móveis, nomeadamente em *swarm robotics*, patrulhamento, busca e salvamento, entre outros.

Adicionalmente, as funções de comunicação são disponibilizadas através de um driver de integração das plataformas em ROS (*Robotic Operating System*), desenvolvido no mesmo laboratório, visto ser uma ferramenta com provas dadas e muito utilizada pela comunidade de investigação na área de robótica móvel. De modo a validar os requisitos funcionais, foram realizados diversos testes ao nível de comunicação ponto a ponto, fazendo a leitura do RSSI (*Received Signal Strength Indication*) das mensagens recebidas pelos companheiros de forma a estimar a distância aos mesmos. Para além disso, foi analisado o tempo de ida e volta de uma mensagem, isto é o RTT (*Round-Trip Time*), tanto em comunicações ponto a ponto como multi-hop.

Palavras-Chave: Comunicação *ad hoc*, robótica cooperativa, ZigBee, XBee, Arduino.

Abstract

"TraxBot mobile robots developed at ISR Coimbra are endowed with XBee modules that use the ZigBee technology as the network protocol, allowing inter-robot communication. These modules fit directly on top of the processor board of the robot, Arduino Uno, which is located inside the platform. The aim of this work is to implement *ad hoc* wireless communication between different members of the team of mobile robots, using the ZigBee technology, integrating and developing the features of the XBee OEM RF module via the standard Arduino serial commands. This work provides a useful tool for research, enabling the interaction and cooperation of a team of mobile robots in areas such as swarm robotics, patrol, search and rescue, among others.

Additionally, the communication functions are provided by a driver of the platform in ROS (Robotic Operating System), developed in the same laboratory, as this is a proven tool, which is widely used in the research community in the area of mobile robotics. In order to validate the functional requirements, several tests were performed at the level of peer communication, namely reading the RSSI (Received Signal Strength Indication) of received messages by peers and the estimation of the distance to them. In addition, the Round Trip Time(RTT) of a message, both in point to point communications and multi-hop was analyzed."

Key Words: *Ad hoc* communication, cooperative robotic, ZigBee, XBee, Arduino.

Declaração

O trabalho nesta dissertação é baseado em pesquisas realizadas no Laboratório de Robótica Móvel (LRM) do Instituto de Sistemas e Robótica (ISR), em Coimbra, Portugal. Nenhuma parte desta dissertação foi apresentada em outro lugar para qualquer outro grau ou qualificação, e é todo o meu trabalho próprio, a menos que referenciado em contrário no texto.

"*God grant me the serenity to accept the things I cannot change, the courage to change the things I can, and the wisdom to know the difference*".

Dr. Reinhold Niebuhar

Conteúdo

Lista de Figuras

Lista de Tabelas

Acrónimos

ACK *Acknowledge.*

APS *Application Sublayer.*

AP *Access Point.*

BPSK + ASK *Binary Phase Shift Keying + Amplitude-Shift Keying.*

BPSK *Binary Phase Shift Keying.*

BSS *Base Station Subsystem.*

CBC *Cipher Block Chaining.*

CCK *Complementary Code Keying .*

COFDM *Coded Orthogonal Frequency Division Multiplexing.*

CSMA/CA *Carrier Sense Multiple Access with Collision Avoidance.*

CSMA/CD *Carrier Sense Multiple Access with Collision Detection.*

DSSS *Direct Sequence Spread Spectrum.*

FHSS *Frequency Hopping Spread Spectrum .*

FH–CDMA *Frequency Hopping - Code Division Multiple Access.*

FSSS/CDMA *Frequency Hopping Spread Spectrum/Code Division Multiple Access.*

GFSK *Gaussian Frequency-Shift Keying.*

IEEE *Institute of Electrical and Electronics Engineers.*

ISM *Industrial,Scientific and Medical.*

LQI *Link Quality Indication.*

LR–WPAN *Low data Rate - Wireless Personal Area Network.*

MAC *Medium Access Control.*

M–QAM *Multi-Level Quadrature Amplitude Modulation.*

OEM *Original Equipment Manufacturer.*

OFDM *Orthogonal Frequency - Division Multiplexing.*

OSI *Open Systems Interconnection.*

O–QPSK *Offset Quadrature Phase-Shift Keying.*

PAN *Personal Area Network.*

PHY *Physical Layer.*

POS *Personal Operating Space.*

PPDU *Physical Protocol Data Unit.*

QoS *Quality of Service.*

RFD *Reduced Function Device.*

RF *Radio Frequency.*

RX *Receive or Receiver.*

SAP *Service Access Point.*

SFD *Start-of-Frame Delimiter.*

TDMA/TDD *Time Division Multiple Access/Time Division Duplexin.*

TRX *Transceiver.*

TTL *Transistor-Transistor Logic.*

UART *Universal Assynchronous Receiver/Transmitter.*

USB *Universal Serial Bus.*

WLAN *Wireless Local Area Network.*

WMAN *Wireless Metropolitan Area Network.*

WPAN *Wireless Personal Area Network.*

WWAN *Wireless Wide Area Network.*

Capítulo 1

Introdução

Esta dissertação relata a implementação de comunicação *ad hoc* em equipas de robôs móveis. Esta foi desenvolvida no Laboratório de Robótica Móvel (LMR) do Instituto de Sistemas e Robótica (ISR) em Coimbra. O principal objetivo deste trabalho de dissertação consiste em implementar comunicação *ad hoc* utilizando a tecnologia ZigBee em equipas de robôs móveis. Este trabalho teve início numa análise comparativa das principais tecnologias de comunicação sem fios utilizadas atualmente na robótica: Wi-Fi, Bluetooth e ZigBee [FCP+12]. Em seguida, e com base nos resultados comparativos das diferentes tecnologias sem fios integrou-se a tecnologia ZigBee nos robôs móveis TraxBot [APC+12], expandindo as funcionalidades dos módulos XBee OEM RF para Arduino, cujo desenvolvimento foi feito em C/C++. Foram realizados vários testes para avaliar a comunicação entre os membros das equipas. Por fim, essas funcionalidades foram disponibilizadas através de um driver de integração dos robôs em ROS (*Robotic Operating System*) [A12], desenvolvido no mesmo laboratório, acompanhado de vários testes a fim de avaliar o bom funcionamento do sistema.

1.1 Contexto e motivação

As tecnologias de comunicação sem fios têm vindo a evoluir rapidamente nos últimos anos, estando amplamente disseminadas na nossa sociedade [CSI07]. Avanços em sistemas microeletromecânicos permitem a integração de sensores, processamento de sinal

e capacidade de Rádio Frequência em dispositivos muito pequenos [CMR11]. Todos os tipos de aplicações portáteis tendem a ser capazes de comunicar sem uso de uma ligação cablada [CMR11]. O objetivo da comunicação sem fios consiste em reunir informações para realizar uma tarefa num determinado ambiente. Para isso, um nó inteligente típico (*por ex.*, sensor, robô, entre outros) é constituído por uma unidade de aquisição, processamento e transmissão de dados. A unidade de aquisição é constituída por transdutores. A unidade de processamento é constituída por microcontroladores e memória. Finalmente, a unidade de comunicação é constituída por *transceivers* para transmitir e receber dados. Devido a estes avanços tecnológicos, têm surgido tecnologias de comunicação sem fios de baixo custo, baixo consumo de energia e baixas taxas de transmissão de dados, que têm motivado a investigação no campo da robótica móvel, principalmente em áreas que exigem interação e cooperação entre robôs móveis. Tecnologias sem fios, tais como Wi-Fi, Bluetooth e ZigBee, permitem aos robôs móveis comunicarem de forma *ad hoc*, possibilitando a cobertura de uma vasta área geográfica, sem a necessidade de utilização de estações centralizadas de comunicação, em ambientes de difícil implementação ou em situações de emergências (*por ex.*, cenários de busca e salvamento). Investigação nas áreas de patrulhamento multi-robô, *swarm robotics* e busca e salvamento (*search and rescue*) tem sido levada a cabo no Laboratório de Robótica Móvel (LRM) do Instituto de Sistemas e Robótica (ISR) em Coimbra [PR11], [CRF+12], [Chopin12]. Para esse fim, foram desenvolvidas neste laboratório plataformas móveis de baixo custo denominadas de TraxBot [APC+12]. Este trabalho incide na implementação da comunicação *ad hoc* nestas plataformas móveis utilizando a tecnologia ZigBee, através da integração do módulo OEM RF XBee série 2, acoplado à placa de processamento da plataforma, Arduino Uno, e a extensão do driver ROS dos robôs.

1.2 Objetivos

O primeiro objetivo deste trabalho é implementar a comunicação *ad hoc* entre diferentes membros da equipa de robôs móveis de baixo custo, fazendo a integração e o desenvolvimento de funcionalidades do módulo XBee OEM RF através de comandos série *standard*

do Arduino. O segundo é a integração das funcionalidades no driver de integração desses robôs em ROS.

Finalmente, esta dissertação também tem o objetivo de avaliar a precisão da estimação de posições dos robôs recorrendo à leitura do RSSI das mensagens recebidas. Para verificar todos estes objetivos, vários testes foram realizados e reportados neste documento.

1.3 Organização da dissertação

Esta dissertação encontra-se organizada em 6 capítulos. O primeiro capítulo apresenta o contexto e motivação, bem como os objetivos específicos deste trabalho. O segundo capítulo aborda a comunicação sem fios e as principais tecnologias utilizadas atualmente, incluindo uma comparação das mesmas. O capítulo 3 descreve a importância da tecnologia ZigBee nas equipas de robôs móveis e como utilizar os módulos XBee série 2 para realizar comunicação *ad hoc multi hop* entre os membros das equipas. O capítulo 4 relata a implementação do sistema, a integração dos módulos XBee série 2 no TraxBot e o desenvolvimento das funcionalidades pretendidas. As experiências e a discussão dos resultados são apresentadas no capítulo 5. Finalmente, no capítulo 6, faz-se o resumo do trabalho e apresenta-se as conclusões finais e sugestões para investigações futuras.

Capítulo 2

Comunicação sem fios

A comunicação sem fios é a transferência de informação entre dois ou mais pontos, que não estão ligados fisicamente. Tudo começou com Michael Faraday (e quase ao mesmo tempo com Joseph Henry) que demonstrou a indução Magnética em 1831, seguido de James C. Maxwell (1831- 1879) que estabeleceu as bases teóricas para os campos eletromagnéticos com as suas famosas equações (1864) [L98]. Finalmente, Heinrich Hertz (1857- 1894) foi o primeiro a demonstrar a transmissão da energia elétrica como ondas eletromagnéticas, através do espaço (1886), provando assim as equações de Maxwell. Anos mais tarde, Nicolas Tesla (1856-1943) aumentou a distância de transmissão das ondas eletromagnéticas.
O nome que tem certamente uma ligação mais estreita com a comunicação sem fios é Guglielmo Marconi (1874-1937). Ele fez a primeira demonstração de telegrafia sem fio em 1895, utilizando a transmissão de ondas longas, com potência de transmissão elevada ($>$ 200 kW). Em 1901 realizou-se a primeira transmissão transatlântica [S03].
A comunicação sem fios (ou *wireless*) já está inserida na sociedade em redes como WPANs, WLANs, WMANs e WWANs [FCP+12], todas voltadas para utilizadores finais de pequenas, médias e grandes empresas, onde o objetivo é a transferência de grandes volumes de dados e voz a altas velocidades. As áreas de aplicação e o alcance do sinal são os critérios para essa divisão [SMK+07] (ver Figura 2.1).
A *Wireless Wide Area Network* (WWAN) é uma rede sem fios destinada a abranger uma grande área geográfica, como uma cidade estado ou um país, através da utilização de vários locais com antenas ou sistemas de satélite [FCP+12]. As *Wireless Metropolitan Area Network* (WMAN) permitem aos utilizadores estabelecer ligações sem fios entre diferentes

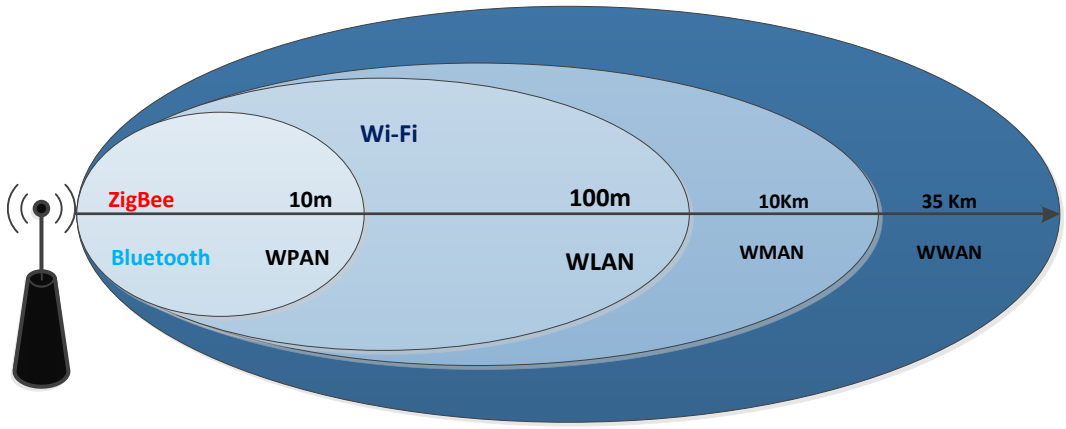

Figura 2.1: Divisão das redes sem fios.

localizações numa área metropolitana (*por ex.*, entre vários edifícios de escritórios numa cidade ou num espaço universitário), sem os custos elevados de instalar fibras óticas ou cabos de cobre e linhas dedicadas [FCP+12]. A *Wireless Local Area Network* (WLAN) é uma rede local sem fios que atinge uma área geográfica limitada, cobre uma área equivalente a um quarto, um edifício ou mesmo um *campus*. Tem como objetivo a substituição de cabos possibilitando assim flexibilidade nas comunicações. Podem ser necessárias infra-estruturas de rede para o funcionamento ou podem operar de modo *ad hoc*. As redes cujo alcance podem atingir os 10m são classificadas como *Wireless Personal Area Network* (WPAN) [SMK+07]. Um caso particular da WPAN é o *Low data Rate-Wireless Personal Network* (LR-WPAN) que é uma rede de comunicação simples, de baixo custo, que permite ligações sem fios em aplicações com potência limitada e requisitos de rendimento mínimo [IIL09]. Essas redes são caracterizadas por terem uma baixa taxa de transferência de dados e baixa potência. A possibilidade de ligação no modo *ad hoc*, modo que não utiliza nenhuma infra-estrutura, é uma das suas vantagens.

2.1 Redes móveis *ad hoc*

As redes móveis *ad hoc*, ou *Mobile Ad hoc NETworks* – MANETs, são paradigmas para a comunicação móvel em que os nós são móveis, sendo assim dinamicamente e arbitrariamente localizados, de tal forma que a comunicação entre os nós não depende de qualquer infra-estrutura estática subjacente [H09]. Os nós de uma MANET são geralmente dispositivos portáteis móveis com recursos limitados: energia, capacidade de processamento e capacidade de armazenamento. Uma vez que nenhuma estrutura fixa ou administração centralizada está disponível, essas redes são auto-organizadas [OHK11]. Devido à falta de infra-estrutura e ao alcance de transmissão limitado dos nós numa MANET, estes dependem dos vizinhos para fazer chegar a informação ao nó com que pretendem comunicar (*i.e.*, comunicação multi-hop). Tais redes são formadas por um número de dispositivos, possivelmente heterogéneos, com capacidades de comunicação sem fios que se associam e dissociam livremente. As MANETs podem ser aplicadas a redes em malha sem fios, aplicações militares, acesso móvel à Internet, redes de respostas a emergência, entre outros. O projeto e a implementação dessas redes apresentam uma série de desafios, tais como [MM08]:

- Auto-organização – cada nó numa MANET deve ser capaz de associar e dissociar da rede livremente e sem qualquer infra-estrutura fixa. São necessários protocolos [ZKS08] que possam apoiar as tarefas de construção de topologia, reconfiguração e manutenção, bem como o reencaminhamento e monitorização de tráfego e controlo de admissão;

- Escalabilidade – refere-se à sua capacidade de reter determinados parâmetros de desempenho, independentemente do número de nós na rede. Isso é altamente dependente da *overhead* da pilha protocolar;

- Tempo de atraso das mensagens (Delay) – o atraso é um parâmetro crítico em determinadas aplicações, como por exemplo em aplicações militares, como as comunicações no campo de batalha ou deteção e monitoramento de tropas, ou em aplicações de saúde, entre outros;

- Taxa de transferência de dados – é um dos parâmetros mais importantes para avaliar

o desempenho das redes de comunicação. A nível da camada física PHY, esta pode ser afetada por erros de pacotes de dados causados pelo ruido e pelas interferências [SGK09];

- Perda de dados – devem existir mecanismos ativos tanto na camada MAC como nas camadas superiores para transferências de dados de forma fiável.
- Gestão de energia – a gestão de energia é importante principalmente quando os nós são alimentados por baterias;
- Manutenção – todas as tarefas em MANETs devem ser automatizadas ou bastante simples para serem realizadas por operadores não humanos;
- Auto-reparação – a rede tem de funcionar mesmo quando o nó controlador sai da rede ou tiver uma falha;
- Heterogeneidade – os nós das redes podem ser dispositivos com diferentes características;

Além disso estas redes são confrontadas com os tradicionais problemas inerentes às comunicações sem fios, tais como segurança, interferências, baixa largura de banda, etc. Apesar de muitas restrições de projeto, as redes móveis *ad hoc* oferecem inúmeras vantagens. Primeiro de tudo, este tipo de rede é altamente adequado para uso em situações em que haja a necessidade de instalar rapidamente uma rede de comunicação (por ex. situações de emergência). Por causa das capacidades de auto-criação, auto-organização e auto-administração as redes *ad hoc* podem ser implementadas rapidamente com mínima intervenção do utilizador. Não há necessidade de um planeamento detalhado de instalação e das ligações de uma estação base.

Ao contrário de uma rede WLAN, que precisa de infra-estrutura de comunicação, numa rede *ad hoc* os próprios nós são responsáveis por fazer o reencaminhamento dos pacotes [ML08]. Nós móveis comunicam-se com outros nós fora do seu alcance, utilizando ligações sem fios *multi-hop*. Além disso, a topologia das redes *ad hoc* é altamente dinâmica, não podem ser usados protocolos de reencaminhamento tradicionais. O protocolo de reencaminhamento deve ser capaz de manter-se com a mobilidade dos nós, que muitas vezes mudam drasticamente a topologia da rede de forma imprevisível [WZA03]. As redes

de infra-estruturas utilizam controlo centralizado, que representa desvantagens para nós móveis, devido a:

• Informação centralizada, se o nó falhar a rede estará comprometida, em outras palavras, baixa robustez.

• Custos elevados, para garantir certo nível de qualidade de serviço (QoS) ao nível de transmissão, são necessárias estações de base caras para cobrir a área de serviço.

• Maior necessidade de comunicação com estação central.

2.2 Tecnologias de comunicação sem fios

Figura 2.2: Logótipo das três tecnologias sem fios: Wi-Fi, Bluetooth e ZigBee.

Uma das razões do rápido crescimento das tecnologias de comunicação sem fios é a capacidade em fornecer flexibilidade e mobilidade na rede. Outro benifício é a formação das redes de forma dinâmica, com baixo custo e fácil implementação [LAS07]. A Figura 2.1 mostra a classificação das redes de comunicação sem fios conforme o alcance. Geralmente, as redes de comunicação sem fios são asseguradas por três tecnologias: Wi-Fi, Bluetooth e ZigBee, que respeitam as normas IEEE 802.11a/b/g, 802.15.1 e 802.15.4, respetivamente .

2.2.1 Wi-Fi

A Wi-Fi (*Wireless Fidelity*) é uma marca registada originalmente pela Wi-Fi Alliance [FCP+12], para descrever a tecnologia subjacente de redes locais sem fios (WLAN) com base nas especificações da IEEE 802.11 [SMK+07]. A Wi-Fi Alliance foi fundada em 1999 por cinco companhias: 3Com, Aironet, Lucent Tecnologies, Nokia e Symbol Tecnolo-

gies. Nesse mesmo ano, a marca foi adotada para a tecnologia baseada na norma IEEE 802.11 [FCP+12]. A tecnologia Wi-Fi permite que o utilizador tenha acesso à internet em qualquer lugar, onde haja uma rede Wi-Fi, quando ligamos um computador ou outro dispositivo a um *access point* (AP) ou de forma *ad hoc*. A transmissão é feita através de ondas de rádio, minimizando o custo em relação às redes Ethernet [ER12]. A arquitetura IEEE 802.11 consiste em várias camadas que interagem e oferecem uma WLAN, que suporta a mobilidade de forma transparente para as camadas superiores. Uma rede baseada na norma IEEE 802.11 é constituida pelos seguintes componentes [S03]: BSS (*Basic Service Set*), STA (*Stations*), AP (*Access Points*) e ESS (*Extend Service Set*). Existem dois componentes principais descritos pela norma 802.11:, a estação móvel (STA) e os pontos de acesso (AP). O STA é o dispositivo que liga à rádio frequência (RF) ou meio sem fio e é normalmente referido como um adaptador de rede ou cartão de interface de rede (*Network Interface Card* - NIC). Cada STA apoiará um conjunto bem definido de serviços, incluindo autenticação privacidade e entrega de dados. Todas as redes 802.11 são construídas em torno de um conjunto de serviços básicos (BSS), que é simplesmente um grupo de STAs tentando se ligar com o outro, e um identificador de serviços (*Service Set Identifier* - SSID) que é utilizado para identificar um ESS. *Extend Service Set*, como o nome indica é um nível acima do BSS na arquitetura 802.11. Quando existem múltiplas APs e STAs (múltiplos BSSs) para formar uma rede WLAN maior, é chamado de conjunto de serviços estendido (ESS). As principais características do Wi-Fi são [SK10]:

- Mobilidade;
- Altas transferências de dados, quando comparado com as tecnologias Bluetooth e ZigBee;
- Facilidade de uso;
- Baixo custo em relação às redes Ethernet.

Essa tecnologia é amplamente utilizada no acesso à Internet, devido a altas taxas de dados quando comparadas com as tecnologias Bluetooth e ZigBee. Devido a uma das características fundamentais dessa tecnologia, que é a elevada mobilidade, hoje em dia é possível construir redes em hotéis, bibliotecas, faculdades, *campus*, lojas, cafés, hospitais,

entre outros. Para além da sua utilização no campo de redes de computadores, também pode ser aplicado na localização e navegação de robôs móveis [BV10].

2.2.2 Bluetooth

O Bluetooth é um padrão global para comunicação sem fios, de curto alcance que permite a ligação de uma vasta gama de dispositivos electrónicos [B12]. A banda de operação do Bluetooth é a *Industrial, Scientific and Medical* (ISM) 2.4 GHz, não necessitando de nenhuma autorização para ser utilizada e está disponível na maioria dos países. O nome Bluetooth vem do nome de um Viking: Harald Blätand (Harald the Bluetooth) [FCP+12], que conseguiu a façanha de unificar dentro do mesmo reino, Dinamarca e Noruega, no momento em que a Europa estava dividida pelas diferenças religiosas e territoriais [LAS07]. Da mesma forma, o protocolo procura unir diferentes tecnologias, como telefones móveis e computadores. As principais características desta tecnologia são a robustez, baixa potência e baixo custo [LYC06]. Em Fevereiro de 1998, a IBM, INTEL, Nokia e Toshiba, juntaram-se à companhia Sueca e em Maio criaram o *Special Interest Group* (SIG) [LAS07]. O conceito Bluetooth e PAN já foram englobados pelo IEEE (o qual tem a marca registada WPAN) no grupo de trabalho 802.15. No entanto, o IEEE limita-se a desenvolver normas, apenas para as duas camadas de protocolo inferiores definidas pelo modelo de referência *Open Systems Interconnection* (OSI) [Osi12]. Por outro lado o SIG desenvolve especificações técnicas de apoio de forma a permitir interoperabilidade dos dispositivos Bluetooth e certifica a conformidade dos produtos [B12].

Para estabelecer a comunicação nos sistemas Bluetooth, primeiro o dispositivo tem que identificar dispositivos na vizinhança (discovery) e, de seguida, deve haver um circuito pré estabelecido. A comunicação é baseada no princípio de mestre-escravo [LAS07], que pode ser feita através de dois tipos de topologias: a *piconet* e a *scatternet*. Uma *piconet* é uma WPAN que consiste num nó mestre e até sete nós escravos ativos, situados dentro de um raio de cobertura de 10 metros [MS10], que utilizam a mesma *hopping sequence* (frequência para o salto (hop) num momento especifico). A *hopping sequence* do canal é definida com base no endereço do mestre que define a *piconet*. Todos os dispositivos

participantes na comunicação numa dada *piconet* são sincronizados utilizando o relógio do mestre [LAS07]. O mestre é responsável por alocar/bloquear nova ligação. Qualquer dispositivo Bluetooth pode ser mestre ou escravo, dependendo do cenário de aplicação. Os escravos comunicam diretamente com o mestre e sob o controlo do mesmo. A comunicação do mestre para os escravos pode ser ponto a ponto ou ponto multi-ponto. A norma suporta *broadcast*, simplesmente removendo o endereço de destino. Para além do modo ativo, a fim de reduzir o consumo de energia, os dispositivos escravos podem estar no modo *parked* ou *standby*. Como os dispositivos ativos têm um endereço de 3 bits, a *piconet* só suporta até sete escravos, sendo que os que estão em modo *standby* utilizam endereços de 8 bits e os *parked* não precisam de endereços. Dois dispositivos escravos não podem comunicar directamente, exceto na fase de *discovery* [LAS07].

Várias piconets podem sobrepôr-se e formar uma *scatternet* [LAS07]. Nas *scatternets* cada *piconet* utiliza diferentes *hopping sequence*, que são determinadas pelo seu mestre. Um escravo pode fazer parte de várias *piconets*. Cada *piconet* tem um único *hopping sequence* que é determinado pelo identificador do dispositivo, esse identificador é constituído por 48 bits e é único no mundo. Antes de sair de uma *piconet*, o escravo informa o mestre atual que não estará disponível para um determinado período de tempo e os restantes na *piconet* continuam a comunicar normalmente. Um mestre também pode deixar a sua *piconet* e agir como escravo numa outra *piconet*. Assim que o mestre deixar uma *piconet*, todo o tráfego dentro dessa *piconet* fica suspenso, até o retorno do mesmo. Uma *piconet* pode comunicar com outras partilhando um nó com diferentes *piconets*. O nó partilhado recebe o nome de *bridge*.

A tecnologia Bluetooth foi desenvolvida com o intuito de estabelecer a comunicação sem fios entre dispositivos, que exigem baixo consumo de energia e baixa taxa de dados. Hoje em dia esta tecnologia está presente nos automóveis, na eletrónica de consumo, nos computadores, na saúde e *fitness,* nas casas inteligentes, na robótica móvel [MS10], etc.

2.2.3 ZigBee

O ZigBee é um padrão global para comunicação sem fios, com foco na padronização e permitindo interoperabilidade de produtos. O nome ZigBee surgiu em analogia à forma como as abelhas deambulam entre flores e trocam informação com outras abelhas sobre onde encontrar recursos [SM06]. Foi criado pela IEEE [K03] com a norma 802.15.4 [K03], e a ZigBee Alliance, para fornecer a primeira norma geral para aplicações de rede que usam a norma IEEE 802.15.4. A ZigBee Alliance é um consórcio industrial que visa promover e desenvolver redes sem fio para fins de controlo e monitoramento industrial, mas também para redes domésticas, aplicações em sensores médicos, jogos e outras áreas de aplicação onde são necessárias redes de baixo custo, baixa potência e interoperabilidade. Para o efeito, a ZigBee Alliance desenvolveu (ZigBee Alliance 2004) e reviu a especificação ZigBee (ZigBee Alliance 2006) [FCP+12].

A rede ZigBee permite comunicações robustas e opera na frequência ISM (*Industrial, Scientific and Medical*), não necessitando de licença para funcionamento na maioria dos paises. As redes ZigBee oferecem uma excelente imunidade contra interferências, e a capacidade de hospedar milhares de dispositivos numa rede (teoricamente 65.536), com taxas de transferências de dados a 250Kbps. As principais razões para o uso da tecnologia ZigBee [CMR11] são:

- Fiabilidade e auto-reparação(*self-healing*);
- Suporta grande número de nós;
- Facilidade de implementação;
- Baixo consumo de energia;
- Segurança;
- Pode ser usado a nível global;
- Engloba uma comunidade com 30 ou mais fornecedores de produtos e serviços;
- Tem protocolos de padrões abertos com taxas de financiamento insignificante ou praticamente inexistente;

- O *firmware* pode ser atualizado remotamente;
- Baixa potência;
- Não necessita de muita manutenção.

O ZigBee está disponível em dois conjuntos de recursos [FCP+12]: ZigBee PRO e ZigBee. Ambos os conjuntos definem como operam as redes em malha. O ZigBee PRO, a especificação mais usada, é otimizada para baixo consumo de energia e apoiar grandes redes com milhares de dispositivos enquanto o ZigBee é destinado às redes com menor número de dispositivos.

2.2.3.1 Dispositivos ZigBee

Nas redes ZigBee participam dois tipos de dispositivos físicos: os **Full Function Devices (FFD)** e os **Reduced Function Devices (RFD).** Os FFDs são dispositivos de função completa que funcionam em qualquer topologia, capazes de desempenhar as funções de Coordenadores de rede, *Routers* ou *End devices*, comunicam com qualquer outro dispositivo e são complexos de implementar. Por outro lado os RFDs são dispositivos de função reduzida, são limitados à topologia em estrela, não podem ser Coordenadores de rede, só comunicam com o Coordenador de rede e têm uma implementação muito simples.
Um FFD comunica com RFDs ou outros FFDs, enquanto um RFD apenas comunica com um FFD. O RFD é destinado a aplicações simples, como um interruptor de luz ou um sensor infravermelho passivo, pois estas aplicações não têm necessidades de enviar grandes quantidades de dados e só se podem associar com um único FFD de cada vez. Consequentemente, o RFD pode ser implementado usando um mínimo de recurso e capacidade de memória [K03]. A norma ZigBee atual requer que um FFD esteja sempre ativo, o que na prática significa que FFDs devem ser constantemente alimentados. Para FFDs alimentados por baterias, estas terão uma vida útil de poucos dias [CMR11]. No padrão ZigBee existem três classes de dispositivos lógicos (Coordenador, Router e End devices) que definem uma rede:

- **Coordenador**, forma a raiz da rede em árvore e pode ser a ponte para outras redes, no intuito de expandi-la, visto que cada nó Coordenador suporta mais de 65000 dispositivos ZigBee. Há exatamente um Coordenador em cada rede, este é responsável por iniciar a rede e selecionar os parâmetros de rede, tais como o canal de rádio frequência, identificador da rede e outros parâmetros operacionais [FCP+12]. O Coordenador também pode armazenar informações sobre a rede como chaves de segurança [CMR11].

- **Router**, funciona como nó intermédio da rede, retransmitindo dados para outros dispositivos. O router pode ser ligado a uma rede já existente, sendo capaz de aceitar ligações de outros dispositivos e ser uma espécie de retransmissor para a rede. A rede pode ser aumentada usando routers ZigBee [CMR11].

- **End devices**, poderão ser de baixa potência ou dispositivos alimentados por baterias. Estes obtêm informações a partir dos diversos sensores e *switches* e têm funcionalidades suficientes para comunicar com os seus "pais" (o coordenador ou um router). Esta funcionalidade reduzida potencia a redução de custos. Estes dispositivos não precisam de estar continuamente ativos, enquanto os outros pertencentes às duas outras categorias precisam.

2.2.3.2 Pilha protocolar

A arquitetura *Low data Rate-Wireless Personal Network* (LR-WPAN) é definida em termos de blocos, a fim de simplificar o padrão. Esses blocos são chamados de camadas. Cada camada é responsável por uma parte do padrão e oferece serviços às camadas mais altas. A organização da pilha protocolar (ver Figura A.1) é baseada no modelo OSI (*Open Systems Interconnection*) de sete camadas [K03]. O ZigBee baseia-se no padrão IEEE 802.25.4, que define as camadas Física (PHY) e MAC (Media Acess Control). A ZigBee Alliance define as camadas de rede e de aplicação.

A camada física (PHY) segue o padrão IEEE 802.15.4, sendo a camada mais próxima da camada do hardware, o qual controla e comunica com um transceiver de rádio diretamente [CMR11]. Esta camada permite a transmissão e receção das PPDUs (Physical

Protocol Data Units), através das ondas de rádio [K03]. As ondas de rádio, deverão funcionar nas seguintes frequências de licença livre: 868 MHz para aplicações Europeias, 902- 928 MHz para aplicações da América do Norte e 2.4 GHz para aplicações globais. Existe um único canal entre 868 e 868.6 MHz, 10 canais entre 962.0 e 928.0 MHz e 16 canais entre 2.4 e 2.4835 GHz [SM06]. Todas as três bandas usam DSSS como modo de acesso. Esta camada é responsável pelas seguintes tarefas [SM06]: ativação e desativação do transmissor de rádio, deteção da potência dentro do canal atual (ED - *Energy Detection*), indicação da qualidade de ligação dos pacotes recebidos (LQI - *Link Quality Indication*), seleção da frequência do canal e transmissão e receção de dados.

A camada MAC (*Medium Acess Control*) fornece a interface entre as camadas física e de rede, para além de gerar e reconhecer os endereços da rede esta camada desempenha as seguintes funções: gestão de *beacons*, acesso aos canais, validação de tramas, relatório de entrega das tramas enviadas e associar e dissociar nós.

A camada de rede é a camada de interface entre as camadas de aplicação e a MAC. Sendo também responsável pela descoberta de novos dispositivos na vizinhança, formação de redes e reencaminhamento (é o processo de seleção do caminho para retransmitir as mensagens para o nó de destino). Esta camada fornece ampla segurança de rede, e permite baixa potência para os dispositivos de forma a maximizar a vida útil da bateria.

A camada de Aplicação é a camada mais alta da pilha protocolar ZigBee e abriga objetos de aplicação. Os objetos de aplicação é o software num *endpoint* que controla um dispositivo ZigBee. Um único nó ZigBee suporta até 240 objetos de aplicação. Cada um suporta *endpoint* numerados entre 1 e 240 (com o *endpoint* 0 reservado para o ZDO).

A subcamada API oferece ferramentas, e descrição como criar um perfil para a pilha ZigBee. Os serviços fornecidos no suporte à aplicação são *Discovery* e *Binding*. O primeiro identifica outros pontos ativos na região de alcance daquele dispositivo. O segundo une dois ou mais dispositivos considerando as suas necessidades e serviços. No ZigBee Device Object é onde está definido o papel do dispositivo na rede (coordenador, router ou end device).

O ZigBee oferece mecanismos de segurança para a camada de rede e camada de suporte à aplicação, cada qual é responsável pela segurança das suas tramas (*frames)*. Serviços de segurança incluem métodos de estabelecimento e transporte das chaves, proteção das

tramas e gestão dos dispositivos [CMR11].

2.2.3.3 Modo de operação das redes ZigBee

Os dispositivos da rede ZigBee operam em dois modos diferentes:

- *Modo Beacon*, no modo *beacon,* todos os dispositivos com funções de *router,* sinaliza a sua presença aos outros *routers* da mesma rede. Já os outros nós não precisam de sinalizar a sua presença, mas têm de ser configurados para perceberem o período em que ocorrerá a sinalização. Pois nesse modo, a maioria dos dispositivos (End devices) estarão no modo *sleep,* o que permite a poupança de energia [FCP+12].

- *Non-Beacon*, neste modo a maioria dos dispositivos da rede permanece sempre com os seus recetores ativos, consumindo mais energia. A camada MAC é responsável pela programação (escalonamento) das transmissões de tramas no modo *non-beacon* usando o método CSMA/CA (*Carrie Sense Multiple Acess With Collision Avoidance*). Neste método, cada nó verifica o canal antes de iniciar a sua transmissão (avaliação de canal livre) para diminuir a possibilidade de transmissões simultâneas. Se o canal estiver livre, o nó inicia a sua transmissão. Quando o nó suspeitar da possibilidade de colisão, aborta a transmissão e aguarda um período aleatório para tentar novamente [SBF+06].

2.2.3.4 Topologia

A Figura 2.3 mostra os três tipos de topologias consideradas pela norma IEEE 802.15.4, a topologia em estrela (star), malha (mesh) e árvore (cluster tree).

Topologia em Estrela (Star Topology), é constituída por um único coordenador e qualquer número de dispositivos finais. Nesta topologia é adotado o modelo de rede mestre - escravo, onde o mestre é o Coordenador ZigBee (FFD) e o escravo pode ser FFD ou RFD. Os dispositivos finais (ED) são física e eletricamente separados uns dos outros e passam informações através do Coordenador. Estes apenas podem comunicar com o coordenador, isto é, não suportam redes multi-hop e redes em malha [CMR11].

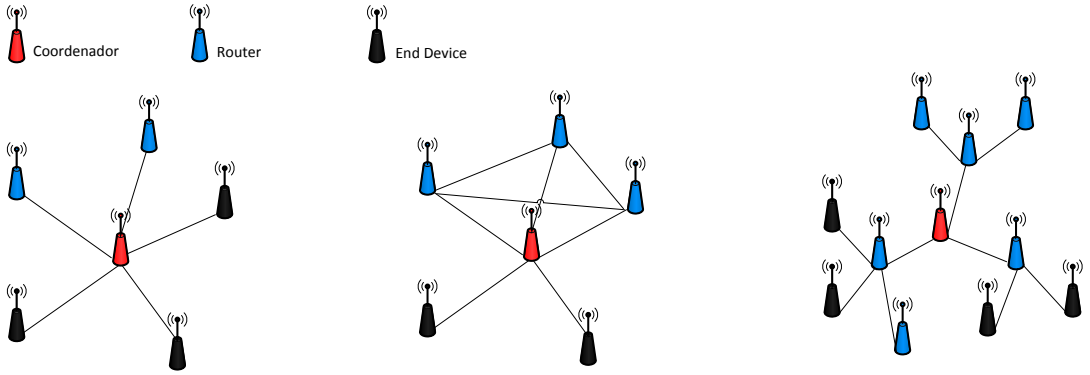

Figura 2.3: Topologia em estrela, malha e árvore.

Topologia em Malha (Mesh Topology), nesta topologia cada nó pode comunicar com qualquer outro dentro do seu alcance. O modo Beacon não é permitido, a sua manutenção é complexa, mas é mais robusta e tolerante a falhas [CMR11].

Topologia em Árvore (Cluster Tree Topology), é similar à topologia em estrela, com a diferença que o nó coordenador pode ter ligações a FFDs, que por sua vez poderão estar ligados a mais nós FFDs e RFDs, de forma a expandir a rede [CMR11].

2.2.4 Análise comparativa

O IEEE define apenas normas para as camadas Medium Access Control (MAC) e Physical (PHY). Diferentes companhias e alianças (ver Tabela 2.1) trabalham para desenvolver especificações que abranjam as camadas de rede, perfis de segurança e aplicações de forma a oferecer interoperabilidade e o potencial comercial destas tecnologias.

A IEEE 802 LAN/MAN Standards desenvolve padrões de rede local e de redes da área Metropolitana, das quais as normas IEEE 802.15.1, IEEE 802.15.3 e a IEEE 802.11/a/b/g fazem parte [IEE12]. A Figura 2.4, mostra a comparação desses padrões em termos de alcance e taxa de transmissão de dados. O ZigBee é o que tem menor taxa de transmissão de dados mas tem maior alcance em relação ao Bluetooth. Por outro lado a Wi-Fi apresenta maior taxa de transferência de dados.

Tecnologia	Norma IEEE	Companhia/Aliança	Tipo de Rede
ZigBee	IEEE 802.15.4	ZigBee Alliance	LR-WPAN
Bluetooth	IEEE 802.15.1	SIG (Special Interest Group)	WPAN
Wi-Fi	IEEE 802.11a/b/g	Wi-Fi Alliance	WLAN

Tabela 2.1: Normas e companhias responsáveis pelas tecnologias.

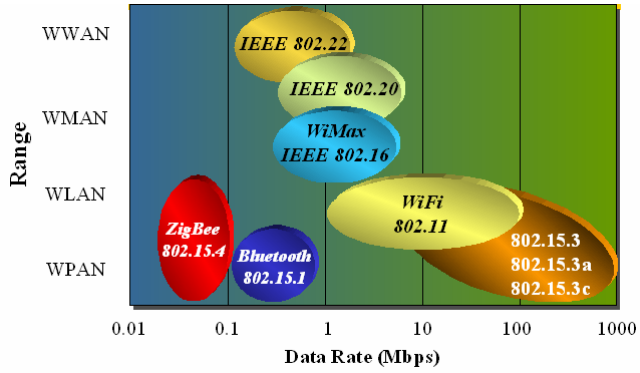

Figura 2.4: Especificação da norma IEEE 802 [FCP+12].

A Tabela B.1 faz um resumo das principais diferenças entre ZigBee, Bluetooth e Wi-Fi. Todas as tecnologias usam a técnica *Spread Spectrum* na banda 2.4 GHz, cuja licença é gratuita na maioria dos países conhecida como *Industrial, Scientific and Medical* (ISM). Esta técnica oferece eficiência de energia na transmissão [PK09]. O ZigBee utiliza o *Direct Sequence Spread Spetrum* (DSSS) com 16 canais e largura de banda de 2 MHz, enquanto o Bluetooth utiliza o *Frequency Hopping Spread Spectrum* (FHSS) com 79 canais e largura de banda de 1MHz. Ambas as técnicas também implementam modulações, que permitem reduzir a complexidade do projeto eletrónico e reduzir o consumo de energia. Os MACs do Wi-Fi e do ZigBee são ambos CSMA/CD, que é um protocolo MAC adequado para aplicações de dados. Cada *piconet* do Bluetooth utiliza técnica TDMA/TDD, que é mais apropriado para aplicações de telefone. *Piconets* diferentes são separadas utilizando a técnica FSSS CDMA. A taxa de dados de ZigBee está na faixa das taxas de dados do Bluetooth que são adequados para aplicações em redes de sensores *ad hoc*. WLANs (Wi-Fi) proporcionam maiores taxas de dados (54Mb/s para o IEEE 802.11g), que são mais adequados para redes de computadores domésticos, pequenos escritórios e redes de acesso

difundidos pelos edifícios públicos. O ZigBee especifica dois tipos de dispositivos (FFD e RFD), que permitem o projeto dos nós finais de forma simples, que na maior parte do tempo se encontram "adormecidos", o que aumenta o tempo de vida da bateria. Apesar da semelhança do número de canais para Wi-Fi e ZigBee a 2.4 GHz são semelhantes, os canais ZigBee são mais estreitos (2MHz), permitindo a implementação de 16 canais não sobrepostos. O IEEE 802.11 DSSS tem apenas três canais não sobrepostos [PK09]. Bluetooth permite até 79 *piconets*, o que está muito além das outras duas tecnologias, mas o número de nós por terminal tem um papel importante na prática uma vez que, normalmente temos poucas *piconets* e um grande número de nós. Nas redes de sensores *ad hoc*, um máximo de 7 nós ligados simultaneamente constitui uma limitação do Bluetooth. Deste modo o ZigBee ultrapassa esta limitação ao permitir maior número de nós por redes. Todos os recursos do ZigBee são projetados para minimizar o consumo de energia, o que tem como resultado o aumento da vida útil da bateria.

A cobertura do Wi-Fi é maior, sendo muito importante para redes de computadores. No entanto o Bluetooth e o ZigBee são projetados para redes de sensores *ad hoc*, cuja cobertura pode ser estendida pela topologia selecionada. A capacidade do ZigBee para formar redes, usando a topologia em árvore, aumenta a cobertura da rede *ad hoc* de forma significativa, tornando-a uma escolha mais adequada para aplicações onde existe um número de sensores espalhados por uma grande área geográfica [PK09]. Como resultado dessas diferenças, Wi-Fi e Bluetooth descobriram suas aplicações específicas. Wi-Fi domina o mercado de WLAN para redes domésticas, pequenos escritórios e acesso a edifícios públicos *ad hoc*. Bluetooth tem substituído os fios das redes *ad hoc* para telefones, ligações dentro dos automóveis e para dispositivos de áudio. Espera-se que o ZigBee desenvolva um mercado similar para a robótica móvel, redes de sensores, para aplicações de dados médicos e outros [PK09].

2.3 Sumário

Neste capítulo foi abordada a comunicação sem fios e a forma como está presente na sociedade, para além das diferentes tecnologias sem fios:Wi-Fi, Bluetooth e ZigBee. Adicionalmente, foram discutidas as vantagens e desvantagens da comunicação *ad hoc* em relação a comunicação utilizando infra-estruturas. No próximo capítulo será abordada a comunicação entre robôs móveis utilizando a tecnologia ZigBee e o módulo XBee série 2 que será utilizado na comunicação.

Capítulo 3

Redes móveis *ad hoc* baseadas em ZigBee

3.1 Comunicação entre robôs móveis

Hoje em dia a comunicação sem fios tornou-se um modo de comunicação essencial para robôs móveis [FLW+05]. Tal como para os humanos, a comunicação é a base para a interação e cooperação entre agentes de uma equipa. Com o avanço da tecnologia *Very Large Scale Integration* (VLSI) e com o crescimento do poder de computação nos últimos anos, a utilização de robôs tornou-se mais comum, com o uso de plataformas mais inteligentes e robustas. Para além disso, os sistemas de múltiplos robôs têm vindo a ser cada vez mais usados devido às suas vantagens, como a divisão de tarefas, o paralelismo ou a robustez a falhas [R06]. Isso significa que as plataformas devem possuir a capacidade de interação e executar trabalhos de cooperação. A força motriz no desenvolvimento de um sistema móvel de cooperação é o seu potencial para reduzir a necessidade de presença humana em aplicações perigosas.

Aplicações tais como a eliminação de resíduos tóxicos, atividades em centrais nucleares, combates de incêndio, pesquisa militar ou civil em missões de resgate, exploração planetária, segurança, vigilância e tarefas de reconhecimento são exemplos de aplicações flagrantes onde a robótica móvel poderá ter alguma palavra a dizer [WZA03], [Chopin12]. Nestes casos a comunicação sem fios fornece a baixo custo soluções para redes de robôs móveis que cooperam de forma eficiente [WZA03].

Figura 3.1: Evolução da comunicação sem fio entre robôs móveis.

A Figura 3.1 mostra a evolução da comunicação sem fios entre robôs móveis. No início das comunicações sem fios entre robôs, a tecnologia infravermelho ou *Infra-Red*(IR) foi utilizada em grande escala, por causa do seu baixo custo [WZA03]. Para além disso, não é necessária licença e os aparelhos elétricos não interferem com a transmissão infravermelha [S03]. Mas esta tecnologia acarreta algumas desvantagens tais como, baixa largura de banda em comparação com outras tecnologias WLAN [S03], elevada suscetibilidade a obstáculos e má qualidade de comunicação (efeito de chuva) [WZA03].

3.1.1 Vantagens da tecnologia ZigBee

A tecnologia ZigBee foi concebida com o intuito de criar uma norma simples para a comunicação sem fios, de baixo custo e com baixa potência de operação, sendo ideal para aplicações de radiofrequência com baixa taxa de transmissão de dados, necessidade de segurança e longa duração de execução [ZA12]. Assim sendo, esta tecnologia é bastante atrativa do ponto de vista da robótica móvel para habilitar a interação e coordenação de uma equipa de pequenos robôs, como por exemplo sistemas *swarm* [CFL+12]. Os requisitos de um robô móvel [K07] para ser usado como parte de um sistema de *swarm robotics* difere dos de um robô móvel para ser usado de forma autónomo. De entre os requisitos necessários, pode-se destacar a capacidade de comunicação, o consumo de energia, o tamanho e o custo [K07], [CFL+12]. O ZigBee através dos módulos OEM RF XBee oferece estes requisitos que são fundamentais nesta área de investigação.

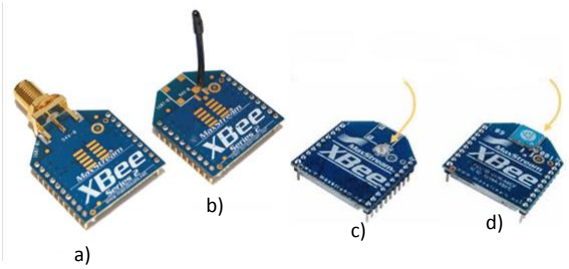

Figura 3.2: Módulo XBee Série 2 com diferentes antenas, a) RPSMA, b) Whip, c) U.FL connector, d) Chip integrado.

3.2 Módulos OEM RF XBee

Os módulos *Original Equipment Manufactures RF* (OEM RF) XBee, são soluções embebidas fabricadas pela Digi International [DI12], oferecendo ligações sem fios por rádio frequência (RF) entre os mesmos, utilizando como protocolo de rede a norma IEEE 802.504/ZigBee. Estes módulos foram construídos com a finalidade de apoiar as necessidades únicas de baixo custo, baixo consumo de energia das redes de sensores sem fio [I07], operando dentro da frequência 2.4GHz e estando disponíveis em dois grandes grupos:

- XBee/XBee-Pro Série 1 (XBs1);
- XBee/XBee-Pro Série 2 (XBs2);

Os módulos XBs1 utilizam a norma IEEE 802.504 como protocolo de rede, para uma rápida ligação ponto a ponto e ponto a multiponto (topologia em estrela), enquanto os módulos XBs2 também conhecidos como XBee/XBee-Pro ZNet 2.5, utilizam como protocolo de rede o ZigBee, permitindo para além das ligações ponto a ponto, ponto a multiponto a implementação de redes em malha. A seleção de um desses módulos deve ser baseada nas necessidades específicas da aplicação. Estes exigem potência mínima e fornecem entrega de dados fiável entre os dispositivos remotos. São constituídos por um microcontrolador e um emissor-receptor, cujas principais características estão apresentadas na Tabela C.1 . Os dois grupos de módulos possuem a mesma forma física e muitos pinos são compatíveis, mas não são interoperáveis. Como as especificações de *hardware* não são muito diferentes, as principais diferenças entre os XBs1 e os XBs2, são mais evidentes nas características

Características	XBee/XBee-Pro (S1)	XBee/XBee-Pro (S2)
Norma/protocolo	IEEE 802.504	Pilha protocolar ZigBee/Rede Mesh (malha)
Tipo de dispositivo lógico	Coordenador/ End-devices	Coordenador, Router ou End-devices
Topologia de rede	Topologia em estrela	Topologia em estrela, mesh e árvore
Chipset	Freescale	Ember
Multi-hop	Não	Sim

Tabela 3.1: Principais diferenças entre XBee/XBee-Pro (S1) e XBee/XBee-Pro (S2).

de *firmware*, conforme apresentado na Tabela 3.1.

3.2.1 Modos de operação

Os módulos XBee utilizam comunicação assíncrona através da porta série, como interface entre um módulo e um hospedeiro. Através da porta série o módulo pode comunicar com qualquer outro dispositivo com lógica e tensão UART (*Universal Assynchronous Receiver/Transmitter*) compatível ou através de conversores de nível para dispositivos série [I07]. Estes funcionam em modo transparente ou no modo de comando (por omissão operam no modo transparente). Neste modo, os módulos atuam como um substituto da linha série, isto é, toda a informação recebida no pino de entrada do módulo é colocada em espera no *buffer* de transmissão para ser transmitida pela antena. Da mesma forma, toda a informação recebida pela antena é colocada no *buffer* de receção para ser enviada para o pino de saída. Os módulos permitem o controlo de fluxo [I07].

Em modo de comando, os carateres recebidos são interpretados como comandos. Existem dois tipos de modo de comando, i) o modo de comando API (*Aplication Programming Interface*) e o ii) modo de comando AT. O modo de comando AT possibilita a modificação e leitura dos parâmetros do módulo através do envio de carateres específicos [FCP+12]. O modo de operação API é uma alternativa ao modo AT, modo definido por omissão. No modo API, todos os dados que entram e saem do módulo XBee estão contidos em

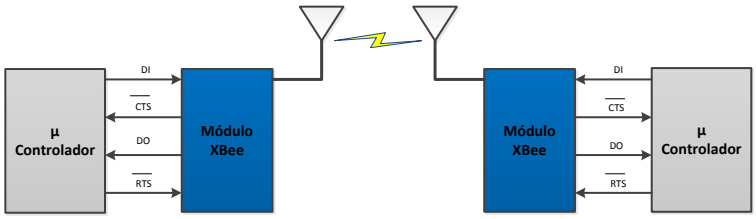

Figura 3.3: Fluxo de dados.

Figura 3.4: Estrutura geral das tramas em modo API.

tramas que definem as operações ou eventos dentro do mesmo. O modo de comando API possibilita a configuração do módulo ao nível da camada de aplicação, que tem de criar os respetivos pacotes, com dados, endereços e identificadores, estabelecendo a comunicação com outros dispositivos. A Figura 3.4 mostra a estrutura geral das tramas API, em que todos os dados recebidos antes do delimitador são descartados silenciosamente. A soma de verificação serve para testar a integridade dos dados [FCP+12], [I07].

A opção API facilita diversas operações, tais como os seguintes exemplos:

• Transmissão de dados para múltiplos destinos sem necessidade de mudar para o modo de comando;

• Aviso de receção de cada pacote RF transmitido;

• Identificação do endereço de origem de cada pacote recebido;

Inerentes ao funcionamento transparente estão os seguintes comportamentos:

• Se os parâmetros dos registos dos módulos estão a ser definidos ou consultados, uma operação especial é necessária para a transição do módulo para o modo de comando.

• Nos sistemas ponto-multiponto, a aplicação deve enviar informação extra de modo a que o(s) módulo(s) de receção pode(m) distinguir dados provenientes de unidades remotas diferentes.

3.2.2 Configuração

A configuração dos módulos faz-se, através de um terminal série ou utilizando a aplicação disponibilizada online pela Digi International [DI12] apresentada na Figura D.1. Utilizando a aplicação é possível atualizar o *firmware* do módulo, configurando-o como Coordenador, *Router* ou *End device*, modo de operação (AT ou API), interface série e a configuração dos parâmetros para a formação de uma rede utilizando os módulos XBee, tais como o PAN ID e o *Scan Channel*.

3.2.3 Formação de redes e endereçamento

Uma rede PAN ZigBee consiste num Coordenador e um ou mais *Routers* e *End devices*. A criação da rede ZigBee, dá-se de forma automática pelos módulos XBee durante o arranque. O Coordenador inicia a rede ZigBee, fazendo um varrimento na área onde se encontra para ver quais são os endereços de PAN e canais que estão disponíveis. Uma vez iniciado, *Routers* e *End-devices* podem ligar-se à rede e recebem um endereço de 16 bits. O Coordenador e os *Routers* podem ou não aceitar a associação de outros dispositivos e permitem no máximo, a associação de 8 dispositívos como "filhos" .

A Formação de redes é regida pelos comandos: SC *(Scan Channels)*, ID (PAN ID), SD *(Scan Duration)* e NJ *(Node Join Time)*. Que podem ser configurados utilizando um terminal série ou o software X-CTU.

A fim de formar uma rede, um Coordenador deve selecionar um canal de operação sem uso e um identificador (ID PAN) para a sua rede, procedendo da seguinte forma: Primeiro procede a um SC *(Scan Channels)*, cuja duração é definida pelo parâmetro SD *(Scan Duration)*. Assim que esta é terminada, o coordenador envia um *beacon request* e mantém-se em escuta . A informação do SC e do *beacon scan (active scan)* é utilizada para escolher o canal e o ID da PAN. Se o parâmetro PAN ID for 0xFFFF, o coordenador irá escolher um ID aleatório. Caso contrário o Coordenador iniciará o PAN com o seu parâmetro ID especificado. Após o arranque do Coordenador, este permite que nós se associem a ele por um tempo especificado no parâmetro NJ (Node Juntion Time). Neste ponto o

canal de operação do coordenador e o PAN ID, podem ser lidos utilizando os comandos CH (Operating Chanel) e ID (PAN ID) [FCP+12]. Antes de um *Router* participar numa rede ZigBee, deve localizar um Coordenador ou um outro router que já pertencem a uma PAN e associar-se a ele. Para isso envia *beacons* em cada um dos canais e fica à escuta. Se o ID for 0xFFFF, o *Router* vai tentar associar a qualquer router ou coordenador. Se a associação for bem sucedida então o *Router* foi iniciado com êxito. Nesse momento é possível consultar o endereço de rede de 16 bits utilizando o comando MY.
Associar um *End-device* a uma PAN é semelhante à associação de um router, sendo a única diferença não permitirem que outros dispositivos se associem a estes. Ao ser associado é possível com o comando MY obter o endereço de 16 bits do *End-device*. Quando a rede está formada, todos os dispositivos XBee, são identificados pelos seus endereços únicos (número de série) ou um identificador configurável pelo utilizador (string ASCII de 20 bytes) que pode ser lido utilizando os comandos SH (Serial Hight) e SL (Serial Low) [FCP+12]. O identificador da String ASCII é configurado utilizando o comando NI (Node Identifier). Para transmitir uma mensagem para outro dispositivo é apenas necessário configurar o endereço de destino. Este pode ser o número de série (64 bits), o endereço de 16 bits ou o NI-String. Os módulos XBee suportam endereçamento para o Coordenador (cujo endereço de rede é "0x0000", conhecido *a priori*) e *broadcast* (cuja mensagem é sempre endereçada para "0xFFFF"). O endereço de destino pode ser configurado no modo AT, utilizando os comandos DL, DH ou DN. Em modo API, a trama *ZigBee Transmit Request API* (0x10) [FCP+12] é utilizada para especificar o endereço de destino.

3.3 Módulos OEM XBee Shields

Os XBee Shields [Ard10] (Figura 3.5) servem de interface entre o módulo XBee e o microcontrolador Arduino Uno [Ard10]. O shield da direita tem dois *jumpers* (pequenas mangas de plástico removíveis, que encaixam em cada dois dos três pinos rotulados XBee/USB). Estes determinam como é feita a comunicação série do módulo XBee para o microcontrolador (Atmel AVR ATMega 328p) e o chip FTDI-USB na placa Arduino. Com os *jumpers* na posição XBee, o pino DOUT do módulo XBee está ligado ao pino RX do microcontrolador; e o DIN está ligado ao TX. Os pinos RX e TX do microcontrolador

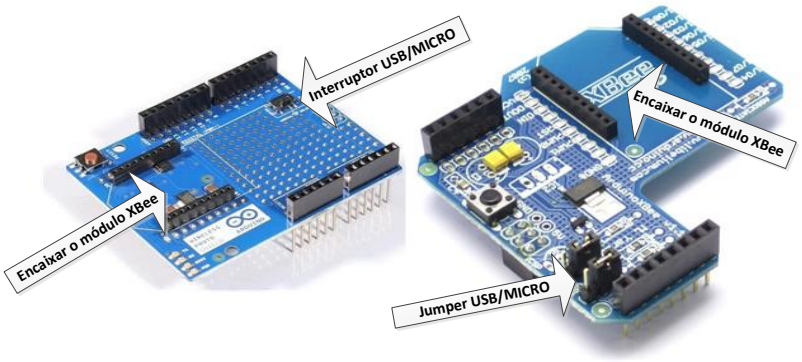

Figura 3.5: XBee Shield.

estão ainda ligados aos pinos TX e RX do chip FTDI-USB respetivamente. Todos os dados enviados apartir do microcontrolador serão transmitidos para o computador via USB ou para o módulo XBee para serem transmitidos pela antena. No entanto, o microcontrolador só será capaz de receber dados, apartir do módulo XBee, não através da USB do computador. Quando os *jumpers* estão na posição USB, o pino DOUT do módulo XBee está ligado ao pino RX do chip FTDI e o pino DIN ao TX. Isto significa que o módulo comunica diretamente com o computador, no entanto isso só funciona se o microcontrolador for removido da placa ou se programar o arduino com código vazio (ver Algoritmo E.1). Se o microcontrolador for deixado na placa ou se o mesmo não for programado com o código do Algoritmo E.1, o módulo conseguirá comunicar com o computador, mas nem o microcontrolador, nem o computador comunicará com o módulo. Para o Shield da esquerda o funcionamento é idêntico, a única diferença, que tem incorporado um interruptor com as posições MICRO/USB, em que o MICRO funciona da mesma forma que a posição XBee no outro shield. Para além das caracteristicas já mencionadas, mais detalhes sobre estes Shields estão descritos em [FCP+12].

3.4 Sumário

Neste capítulo, foi discutida a comunicação entre robôs móveis utilizando a tecnologia ZigBee. Introduziu-se também os módulos XBee série 2, que são utilizados nesta dissertação na comunicação entre os robôs. No próximo capítulo será abordada a comunicação *ad hoc multi-hop* e a implementação do sistema de comunicação entre os robôs móveis.

Capítulo 4

Implementação de redes móveis *ad hoc* ZigBee

4.1 Hardware

Os robôs móveis TraxBot [APC+12] desenvolvidos no ISR Coimbra(ver Figura 4.1) possuem módulos XBee série 2 que se acoplam à placa de controlo Arduino Uno no interior das plataformas. Esses robôs foram projetados para atender às seguintes necessidades [APC+12]:

- Robustez;
- Baixo custo;
- Dimensão reduzida;
- Elevada autonomia energética;
- Possibilidade de expansão;
- Processamento híbrido, num microcontrolador dedicado ou num notebook;
- Comunicação sem fios ZigBee, ou WiFi 802.11 b/g se for acoplado um notebook;
- Integração em ROS.

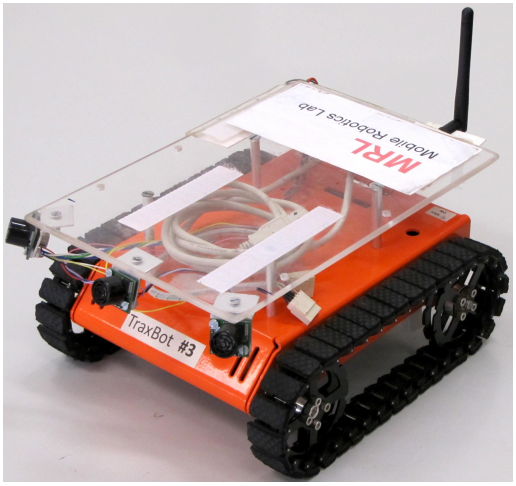

Figura 4.1: Imagem do robô TraxBot.

A Figura 4.2 mostra um esquemático dos principais componentes destes robôs, em que a placa de processamento, o Arduino Uno, encontra-se no meio da Figura. As informações dos restantes componentes do circuito principal e da montagem dos robôs podem ser encontradas em [APC+12].

O Arduino Uno é uma placa de desenvolvimento *open source,* que tem embebido um microcontrolador de 8 bits Atmel ATmega 328p da Atmel [Atmel09]. Atendendo às necessidades do projeto, esta placa é ideal para robôs compactos, permitindo a otmização de espaço e consumo de energia. Possui um circuito integrado composto por uma unidade central de processamento (CPU - *Central Processing Unit*), memória e entradas e saídas. O CPU possui uma frequência de relógio de 16MHz e oferece 14 portos de entrada/saída (6 dos quais podem ser utilizados como saídas analógicas em largura de pulso modulado PWM (*Pulse Width Modulated*), 32kB de memória de programa, 2kB de SRAM e duas interrupções externas que são mapeadas nos pinos 2 e 3 da placa. Esta placa tem uma série de facilidades para comunicar com o computador, outro Arduino ou outros microcontroladores. O ATmega8U2 [Atmel09] embebido na placa Arduino Uno permite a comunicação série através de USB para virtualizar a porta COM, que pode ser utilizada para a comunicação com um computador externo. O microcontrolador também oferece a comunicação série UART TTL (5v). Adicionalmente, o Arduino possui um ambiente de desenvolvimento *open source* e multi-plataforma (*i.e.*, executa em Windows, Macintosh OSX e Linux) com características extensíveis de *software* e *hardware*, ou seja pode ser

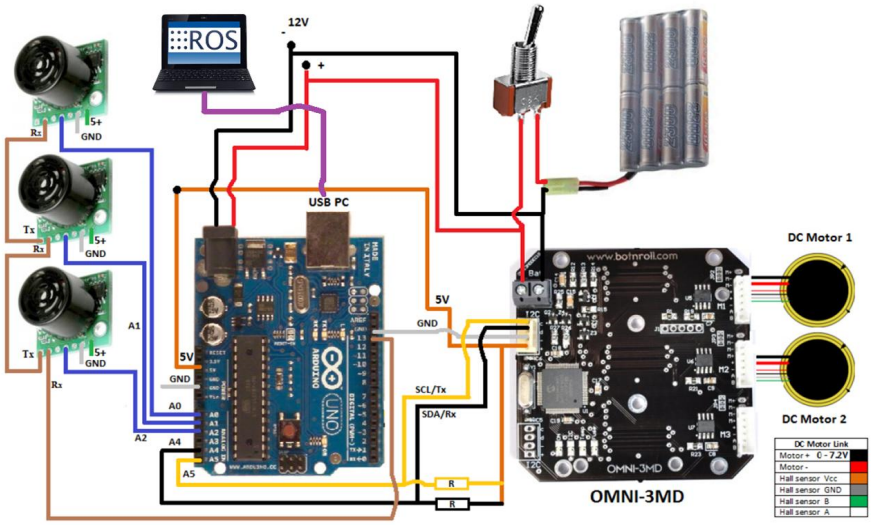

Figura 4.2: Circuito principal do TraxBot [APC+12].

estendida através do uso de bibliotecas C/C++. Podem ainda ser acoplados *shields* no topo da placa, alargando assim as suas capacidades.

4.2 Integração dos módulos XBee em robôs móveis

Aproveitando as características extensíveis de *hardware* do Arduino Uno, os módulos XBee série 2 são integrados nos TraxBot utilizando os *Shield* XBee, que os acopla à placa através de ligações próprias para o efeito, como mostra a Figura 4.3.

Os módulos XBee comunicam com o Arduino Uno utilizando a comunicação série UART TTL (5V), disponível nos pinos 0 (RX) e 1 (TX) da placa e estendida utilizando o XBee Shield.

4.3 Implementação de software

O objetivo principal deste trabalho é implementar a comunicação entre robôs móveis de uma mesma equipa. O diagrama de casos de uso (Figura 4.4) descreve os requisitos funcionais pretendidos. Os robôs, atores deste sistema, são habilitados a identificar novos

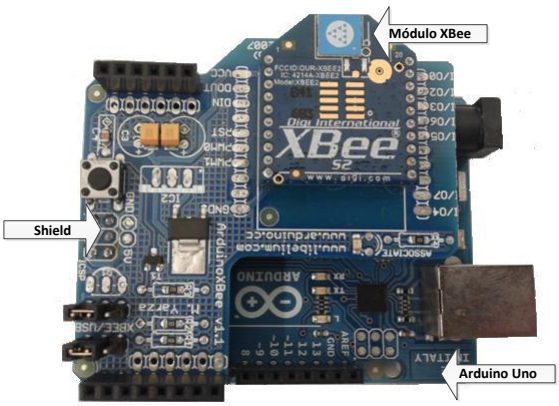

Figura 4.3: Arduino Uno, Shield e o módulo XBee.

nós na rede (sejam robôs ou outros agentes), tal como nós que abandonam a rede. Para além disso, deverão enviar e receber comunicações simples de outros nós; fazer reencaminhamento de dados e *broadcast*; e ler a intensidade do nível de sinal(RSSI) das mensagens enviadas pelos nós vizinhos.

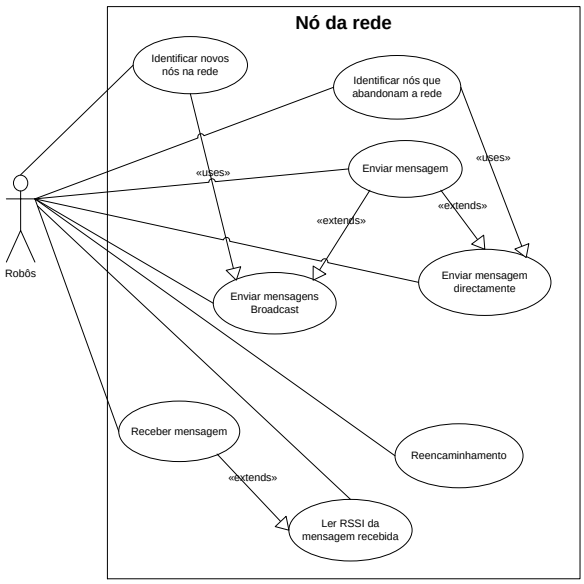

Figura 4.4: Diagrama de casos de uso do sistema.

A Figura 4.5 mostra as principais dependências do *software* desenvolvido, *XBeeNode*

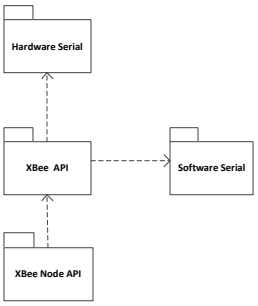

Figura 4.5: Diagrama de pacotes.

API, que depende diretamente do pacote *XBee API* (implementa as funcionalidas dos módulos XBee) e indiretamente dos pacotes *HardwareSerial* e *SoftwareSerial.* O diagrama de classes do *software* desenvolvido encontra-se na Figura F.1. Para os casos de uso *enviar mensagem* e *receber mensagem* são utilizadas as funções do pacote de software XBee API, que são bibliotecas *open source* do Arduino. Essas funções estão disponíveis com os nomes *send()* e *readPacket()* na classe XBee (ver Figura F.1), que o que fazem é meramente escrever e ler na porta série utilizando as funções da biblioteca *HardwareSerial* também disponível no ambiente de desenvolvimento do Arduino. A biblioteca *SoftwareSerial* foi utilizada para a virtualização de uma porta série através dos pinos digitais do Arduino, como será explicada na subsecção 4.3.2.

O caso de uso *Identificar novos nós na rede* é implementado na classe *XBeeNode* através da função *nodeDiscovery().* A função *nodeDiscovery()*, identifica todos os nós que já aderiram à PAN ZigBee. A emissão deste comando faz com que o módulo envie uma mensagem *broadcast* através da PAN e os dispositívos que recebem o comando irão enviar uma mensagem de resposta que inclui os endereços de 64 e 16 bits do dispositivo e o identificador do nó, NI [FCP+12].

O fluxograma da Figura G.1 descreve o funcionamento da função *nodeDiscovery().* Utilizando a função *send()* da classe *XBee*, é enviada uma mensagem contendo o comando ND. De seguida espera-se 6 segundos até todos os nós responderem. As respostas chegam em tramas específicas [FCP+12]. As respostas são depois processadas utilizando a função *processResponse()* implementada na classe *XBeeNode*, que separa os campos da mensagem onde estão os endereços de 16 bits (endereço de rede) e o endereço de 64 bits (número de série do dispositivo), entre outros [FCP+12]. Quando a mensagem é processada, estes

endereços são guardados numa tabela (*XBeeNodeList*) que possui informação de cada nó da rede, utilizando a função *updateAddr().*

O caso de uso *Identificar nós que abandonam a rede* é implementado através da função *checkTheNetwokOutput().* A função *updateTabAbandon()* é utilizada para atualizar a tabela de endereços, consoante os nós que abandonam a rede. O caso de uso *Ler RSSI da mensagem recebida* foi implementado através da função *getRSSI(),* que utiliza a função *sendAtCommand()* para consultar o RSSI da mensagem recebida.

4.3.1 Comunicação entre os robôs móveis

Como já foi referido na secção 3.2, os módulos XBee integram a pilha protocolar ZigBee, permitindo a transmissão dos pacotes de dados por *broadcast*, *multicast* e *unicast*, endereçando os dispositivos através dos endereços de 16 ou 64 bits.

Transmissões por *broadcast* destinam-se a ser propagadas na rede de modo a que todos os nós recebam a mensagem. Para isso, estes nós devem retransmitir a mensagem 3 vezes. Cada nó que transmite por *broadcast* cria uma entrada na tabela de *broadcast* local durante 8 segundos, que é usada para controlo das mensagens, evitando que estas sejam retransmitidas indifinidamente. Esta tabela suporta 8 entradas [I07].

A transmissão por *multicast* é semelhante, a diferença é que somente os nós que fazem parte do grupo *multicast* é que receberão as mensagens. As transmissões *unicast* ZigBee utilizam endereços de 16 bits para transmitir os dados, mas somente os endereços de 64 bits são conhecidos *a priori*. Neste caso, antes de estebelecer a rota de comunicação, os dispositivos têm de obter os endereços de 16 bits dos outros que fazem parte da rede. Assim sendo, o nó envia uma mensagem por *broadcast* para obter o endereço de rede dos outros nós. O dispositivo que tiver o endereço de 64 bits igual ao da mensagem de *broadcast* envia uma mensagem com o seu endereço de rede para o nó de origem. Quando o nó receber a mensagem, poderá transmitir os dados.

Com módulos XBee Série 2, utilizando a topologia em malha (ver Figura 2.3) pode -se realizar a comunicação multi-hop. Para descobrir a rota da mensagem os dispositivos configurados como *Routers* ou Coordenador podem participar no estabelecimento da rota

entre a origem e o destino utilizando um processo chamado *route discovery* [I07]. Este processo baseia-se no algoritimo de encaminhamento *Ad-hoc On-demand Distance Vector routing* (AODV) [I07].

A pilha protocolar do ZigBee também faz a confirmação das mensagens enviando mensagens de ACK provenientes das camadas MAC ou *Aplication Support* (APS) [I07].

Dado que estas operações estão implementadas na pilha protocolar ZigBee, neste trabalho cada nó móvel terá uma tabela onde serão guardados todos os endereços dos nós na rede. Quando um necessite comunicar com outro, deverá apenas percorrer a tabela, escolher o endereço desejado e enviar a mensagem. As restantes operações, tais como a escolha da rota e o reencaminhamento da mensagem, são realizadas automaticamente pelos módulos XBee série 2.

4.3.2 Extensão do driver ROS dos robôs

Com o crescimento da robótica nas últimas décadas [GJS+07] e, principalmente, com a integração de diversos sensores em plataformas robóticas, o desenvolvimento de *software* para robôs tornou-se uma tarefa árdua. Diferentes robôs podem ter *hardware* distintos, tornando a reutilização de código não trivial. Além disso, exige ao programador um conhecimento avançado do *hardware* específico. Para contornar esses desafios, muitos investigadores da área da robótica desenvolveram uma grande variedade de *frameworks* para gerir a complexidade e facilitar a rápida criação de protótipos de *software* para experiências, resultando em muitos *softwares* para sistemas robóticos utilizados no meio académico e na indústria. Cada um desses *frameworks* foi projetado para uma finalidade específica, em respostas às limitações encontradas noutros *frameworks* disponíveis ou na tentativa de melhorar os aspetos mais importantes [QGC+09]. Um dos *frameworks* mais utilizados atualmente para o efeito é o ROS (*Robot Operating System*).

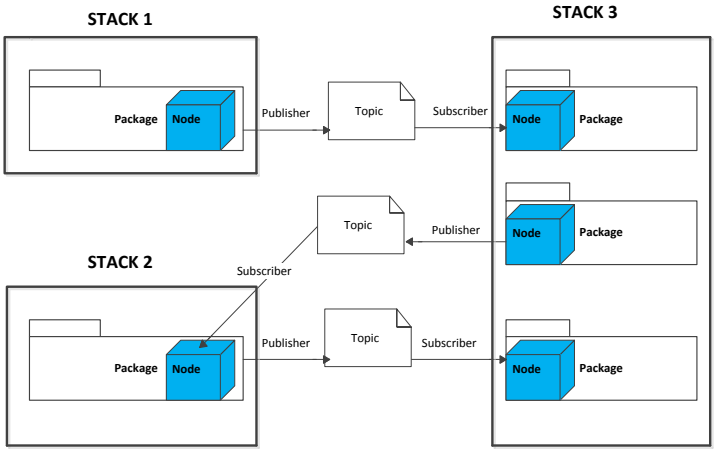

Figura 4.6: Organização do sistema de ficheiros ROS.

4.3.2.1 ROS: Robot operating system

O ROS é um sistema operativo, *open source* e muito utilizado atualmente, que fornece bibliotecas e ferramentas para facilitar o desenvolvimento de *software* para robótica de forma rápida e simples com o qual o programador não tem de se preocupar com as dificuldades inerentes à programação do *hardware* [RC12]. No entanto, não é um sistema operativo comum do ponto de vista de escalonamento de processos, mas oferece serviços de sistemas operativos *standard,* tais como abstração de *hardware,* controlo a baixo nível, comunicação entre processos utilizando mensagens, e um sistemas de ficheiros próprio. O sistema de ficheiros ROS é constituído por *packages,* que estão organizados em *stacks.* As *packages* são constituídos por processos, chamados de *nodes* que comunicam entre si através de mensagens, publicando ou subscrevendo estas num tópico, como mostra a Figura 4.6.

Uma característica essencial no ROS é que estes *nodes* podem correr na mesma máquina ou em máquina diferentes, sendo totalmente transparente para o utilizador. O objetivo de disponibilizar as funcionalidades do módulo XBee no driver ROS decorre precisamente da necessidade de abstração do *hardware,* possibilitando o utilizador usá-las de forma fácil e transparente satisfazendo as suas necessidades. Para além disso, estas funcionalidades representam um contributo para a comunidade ROS, que não suporta comunicação ZigBee.

Como mostra a Figura 4.7, a comunicação entre o computador que executa o ROS e o

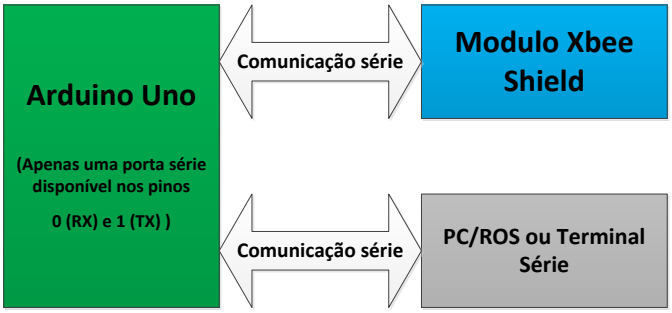

Figura 4.7: Comunicação entre o Arduino, Módulo XBee Shield e PC ROS ou Terminal série.

Arduino é feita utilizando um cabo USB, e esta mesma porta é utilizada para a comunicação com o XBee série 2.

Devido à limitação do Arduino Uno possuir apenas uma porta série para comunicação, houve a necessidade de encontrar uma solução para implementar a comunicação entre o Arduino Uno com o módulo XBee e ao mesmo tempo, comunicar com um computador executando o ROS ou um terminal série.
Uma das possibilidades seria a substituição da placa Arduino Uno por outra placa de desenvolvimento sem esta limitação, *por ex.* Arduino Mega [Ard10]. No entanto esta foi descartada pois violava uma das restrições do projeto: o custo. A solução adotada foi a virtualização de uma porta série através dos pinos digitais do Arduino (ver Figura 4.8) utilizando a biblioteca *SoftwareSerial* para a comunicação com o módulo XBee série 2 e utilizar a porta série para a comunicação com o computador. Para além das necessidades de adaptação do *hardware*, foi necessário adaptar a biblioteca *Xbee API*, de modo a funcionar com a biblioteca *SoftwareSerial* que emula as funcionalidades do *HardwareSerial*.

Encontrada a solução para a comunicação entre o XBee série 2, o Arduino Uno e o computador, procedeu-se à extensão do driver ROS. A Figura 4.9 representa o diagrama de blocos da comunicação entre o driver ROS e o Arduino Uno. É de notar que o *firmware* residente no Arduino Uno pode comunicar com qualquer outro *framework* , ou um terminal série desde estes respeitem o mesmo protocolo de comunicação, que será descrito mais à frente neste capítulo. O driver ROS foi assim denominado inicialmente porque foi desenvolvido no intuito de integrar os robôs TraxBot na arquitetura ROS. Para a comunicação série entre o PC/ROS foi utilizada a *package cereal_port* [CPP11] que

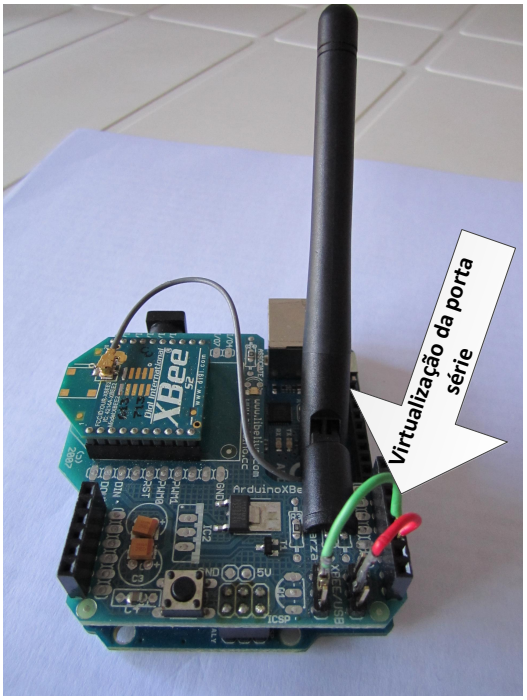

Figura 4.8: Virtualização da porta série para a comunicação entre o Arduino e o XBee.

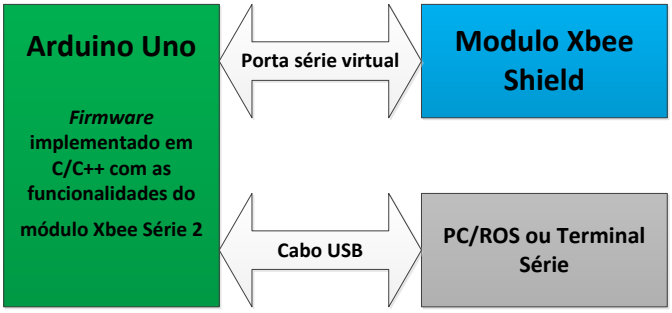

Figura 4.9: Diagrama de blocos.

está incluida na *stack serial_ communication* [SCS11], desenvolvidos no Laboratório de Sistemas Embebidos (LSE) do ISR. Esta *package* disponibiliza funções para comunicação série através do ROS, a fim de trocar dados entre o Arduino Uno e o PC.

A Figura 4.10, mostra a estrutura da trama do protocolo utilizado na comunicação entre o PC/ROS e o Arduino Uno. O carater '@' é utilizado para indicar o início da trama e são utilizadas vírgulas ',' para separar os parâmetros. O carater '*e*' é utilizado para identificar a terminação da trama. Os dados que não estão contidos nas tramas específicas serão descartados. Como se pode verificar na trama de dados, o comando identifica a ação a ser desencadeada pelo *firmware* residente no Arduino Uno. Por exemplo, se

Char inicial	comando	sep	1º parâmetro	sep	2º parâmetro	sep	...	parâmetro n	Char final
@	1 -24	,	Int num	,	Int num	,	...	Int num	e

Figura 4.10: Estrutura da trama do protocolo utilizado [A12].

um utilizador no ROS deseja enviar uma mensagem *broadcast* para os robôs na rede, a trama deverá ser constituída por "@23,p_1,p_2,...,p_ne", ou se desejar enviar uma mensagem para um determinado robô, a trama seria constituída por "@22,idRobot,p_1,p_2,...,p_ne", onde p_1...p_n são os parâmetros (ver Figura 4.10). No ROS driver estão incluídos todos os requisitos funcionais descritos no caso de uso (ver Figura 4.4) e outras funcionalidades que permitem obter informações tais como: característica do módulo XBee Série 2 que está integrado no robô, configurações da rede e endereços do próprio módulo. O algoritmo do driver encontra-se em anexo (ver H.1) e o código em C++ pode ser consultado no manual de referência.

4.4 Sumário

Neste capítulo foi abordada a implementação da comunicação *ad hoc multi-hop*, utilizando a tecnologia Zigbee. Para isso, foram utilizados os módulos XBee série 2 que permitem a implementação da topologia em malha. Também foi abordada a implementação do software de modo a permitir a comunicação entre os membros da equipa dos robôs móveis e a extensão dessas funcionalidades para o driver ROS dos robôs. No capítulo 5 será apresentado os resultados e análise dos mesmos.

Capítulo 5

Apresentação e análise de resultados

5.1 Medição do RSSI

A propagação de ondas eletromagnéticas através de qualquer meio diferente do vácuo é sempre acompanhada de perdas causadas pela absorção de potência pelas partículas do meio, o que causa a diminuição da potência da onda com a distância, ou seja a atenuação. O *Received Signal Strength Indication* (RSSI), em Português, Indicação da Intensidade do Sinal Recebido, é a medida de potência de um sinal recebido. O conhecimento do RSSI pelo nó é de primordial importância numa rede de sensores sem fios (RSSF), visto que este é o ponto de partida para importantes serviços próprios dessas redes. O RSSI é utilizado para diversas funções, especialmente para a localização dos nós dentro da rede e para a estimação da qualidade de suas ligações. Porém, as incertezas envolvidas na medição da intensidade do sinal recebido conduzem a imprecisões nos resultados obtidos em tais serviços executados pela rede [GV06]. Os módulos XBee oferecem duas formas de ler o nível de sinal (em -dBm) do último pacote recebido com sucesso:

1. Codificado na largura de pulso do sinal modulado (PWM - *Pulse-Width Modulated*) e disponível no pino 6 do módulo XBee. Quando o módulo receber uma mensagem, o PWM é definido com base no valor do RSSI recebido. Esta informação diz respeito apenas à qualidade do último hop.

2. Utilizando comandos API (comando DB). O comando DB também indica apenas o RSSI do último *hop*. Se a transmissão for multi-hop este comando não indica a

Figura 5.1: Setup da experiência "*indoor*" realizada.

qualidade de todos os *hops*.

Os valores do RSSI dos módulos XBee série 2 variam entre -26 a -98 dBm, utilizando as antenas integradas *Whip* [I07]. No entanto, o manual especifica que os valores são entre -40 dBm e a sensibilidade do módulo XBee recetor [I07]. Uma da vantagens principais do 2º método para a leitura do RSSI é não utilizar interrupções externas, visto que o Arduino Uno só suporta duas interrupções externas, e foi estabelecida como uma restrição do projeto, sendo estas reservadas para aplicações futuras para o TraxBot. Assim sendo, nestas experiências foi utilizado o método 2 para a leitura do RSSI.

5.1.1 Experiências *"indoor"*

Nas experiências de leitura do RSSI, foram colocados dois robôs TraxBot contendo os módulos XBee série 2 (*i.é.*, um como Coordenador e outro como Router/End device), em linha de vista, variando a distância entre eles com incrementos de 10cm até 20m, a fim de verificar a relação dos valores de RSSI e a distância. Os dados foram adquiridos pelo *notebook* incorporado no TraxBot, para posterior tratamento, como se verifica na Figura 5.1. Por cada incremento foram adquiridos 6 medidas. O resultado é apresentado na Figura 5.2, que mostra a relação entre o RSSI e a distância da mediana dos dados adquiridos. Verifica-se que à medida que a distância aumenta, o valor do RSSI tende a diminuir. Como pode ser verificado na mesma figura, há muita variação dos sinais recebidos, sobretudo para distâncias elevadas devido às múltiplas reflexões das ondas electromagnéticas nas paredes.

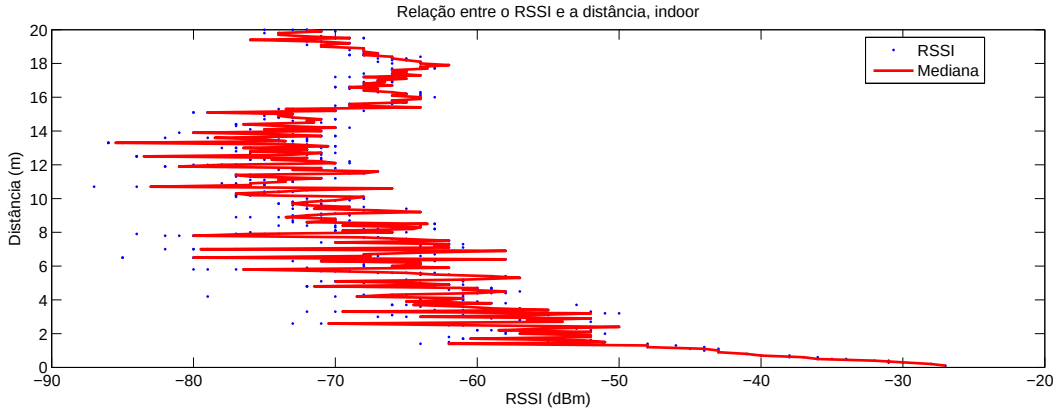

Figura 5.2: Relação entre RSSI e a distância.

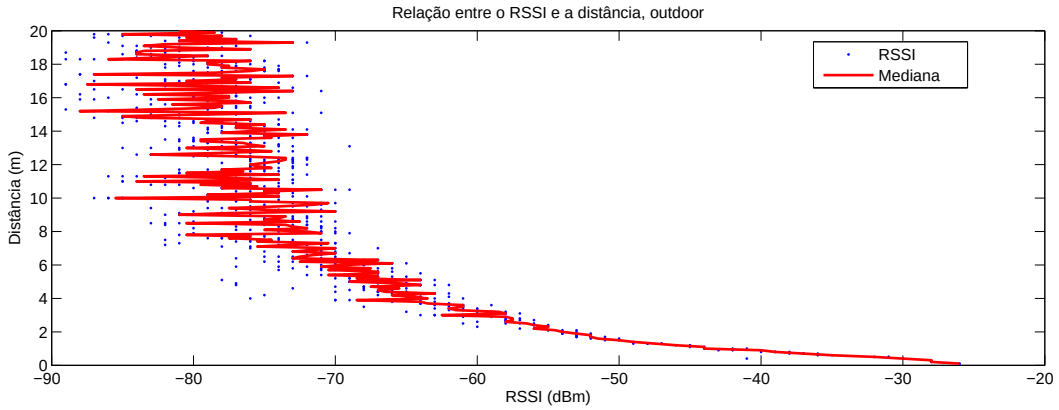

Figura 5.3: Relação entre RSSI e a distância.

5.1.2 Experiências *"outdoor"*

A Figura 5.3, indica também a relação entre o RSSI e a distância num ambiente aberto (ver Figura 5.8). Verifica-se é que sinal tem menor variação em comparação com a experiência realizada no ambiente "indoor", mas mesmo assim apresenta significativas variações. Com o intuito de obter uma relação entre RSSI e distância com menor ruído, foram realizadas novas experiências "outdoor" aumentando o número de amostras das leituras para 30 por cada iteração e substituindo as antenas dos módulos por antenas conetores U.FL [DBA12](ver Figura 4.8).

Como se pode verificar na Figura 5.4, que representa a curva da mediana dos dados adquiridos a variação nos dados de RSSI, utilizando as antenas integrados *Whip,* continuam com variação semelhante , mas com a alteração das antenas para conetores

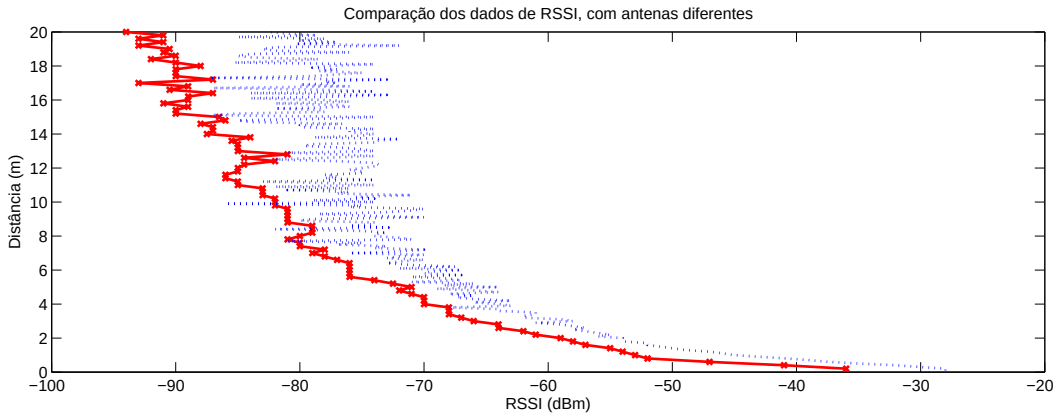

Figura 5.4: Comparação dos dados de RSSI entre antenas integradas Whip e conetores U.FL.

U.FL, as variações diminuiram significativamente. Foram estas as antenas utilizadas nas experiências da estimação de distância entre nós utilizando o RSSI, apresentadas na próxima secção.

5.2 Estimação da distância entre nós utilizando o RSSI

Existem vários métodos para estimar a distância entre dois robôs. Para isso é necessário avaliar grandezas físicas que estabelecem uma relação conhecida com a distância. Esta grandeza pode ser o tempo que um sinal demora a percorrer a distância entre o recetor e o emissor, isto é, o tempo de chegada (TOA) [SCM+11], a potência de um sinal acústico [LVH+05], o RSSI, entre outros. É importante considerar que todos estes métodos estão sujeitos a erros, a utilização de cada um depende dos requisitos do projeto.

Neste trabalho o RSSI foi utilizado para estimar a distância entre dois robôs. Utilizando os dados obtidos nas experiências *outdoor* foi calculada uma equação que relaciona a distância em função do RSSI. A equação (5.1), foi obtida utilizando a ferramenta do matlab *Curve Fitting Tool* [fitTool] para aproximar a curva que relaciona a distância (d) em função dos dados de RSSI ($rssi$), através de uma função gaussiana com os seguintes coeficientes $a_1 =$ *4.283*, $b_1 =$ *-90.84*, $c_1 =$ *6.973*, $a_2 =$ *31.69*, $b_2 =$ *-128.6*, $c_2 =$ *41.24*. A

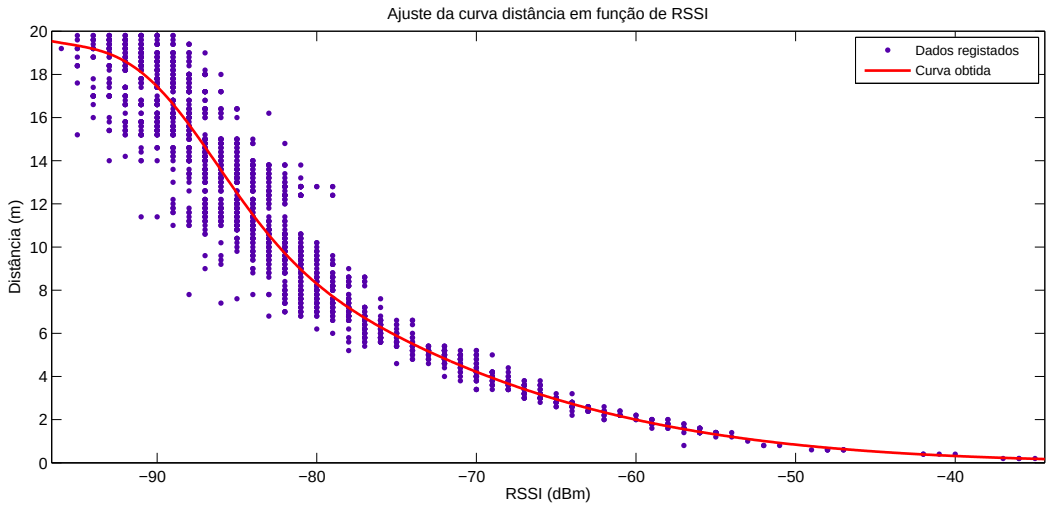

Figura 5.5: Curva de estimação de distância, obtida em ambiente *outdoor* e sem obstáculos.

qualidade da curva obtida é avaliada pelo parâmetro $R^2 = 0.9668$, em que R^2 (R-*square*), denominado por coeficiente de determinação, varia entre *0.0 e 1.0* .Sendo que, quanto mais próximo de *1.0* , melhor será a aproximação.

$$d(rssi) = a_1 exp(-((rssi - b_1)/c_1)^2) + a_2 exp(-((rssi - b_2)/c_2)^2) \tag{5.1}$$

Após a obtenção da equação (5.1), que relaciona a distância com o RSSI, foi realizada uma experiência "outdoor" em espaço aberto onde são obtidas 30 amostras do RSSI, com a distância dos robôs a incrementar de 20cm até um máximo de 20 metros. A Figura 5.6 mostra o resultado obtido na experiência de estimação da distância, nesta figura está a ser comparada a distância real (linha azul continua) e a distância estimada (linha preta tracejada) e a vermelho a evolução do erro absoluto (calculado através da mediana do RSSI) em função da distância. O que se verifica é que apartir dos 6m a estimação da distância, começa a ser menos precisa devido a flutuações nos valores do RSSI . A variável da equação (5.1), *rssi,* é obtida calculando a mediana das 30 amostras de *rssi* para cada iteração.

De forma a quantificar o erro, para cada distância foi calculado o erro médio e o erro máximo (*i.é.*, pior caso) numa amostra de 30 medidas de RSSI obtidos, o resultado é apresentado na Figura 5.7.

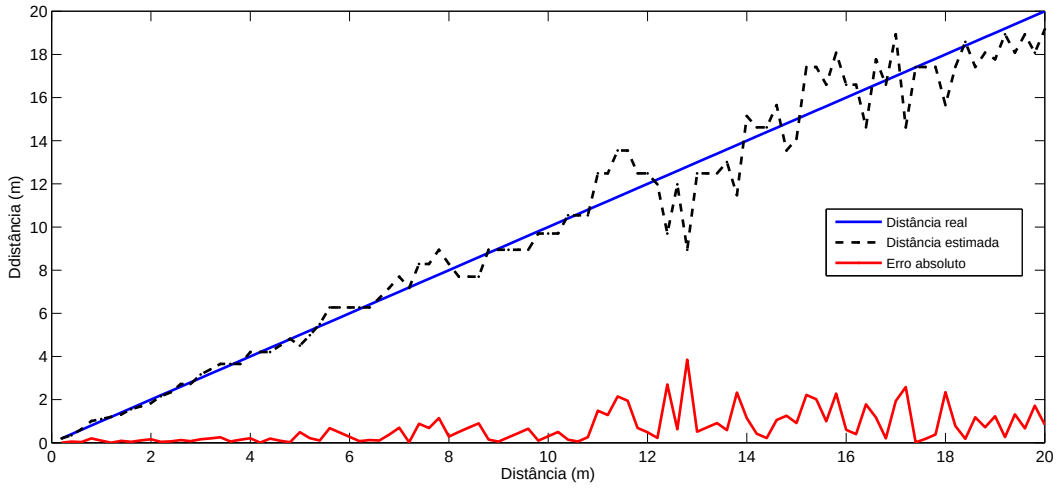

Figura 5.6: Análise do erro na estimação da distância.

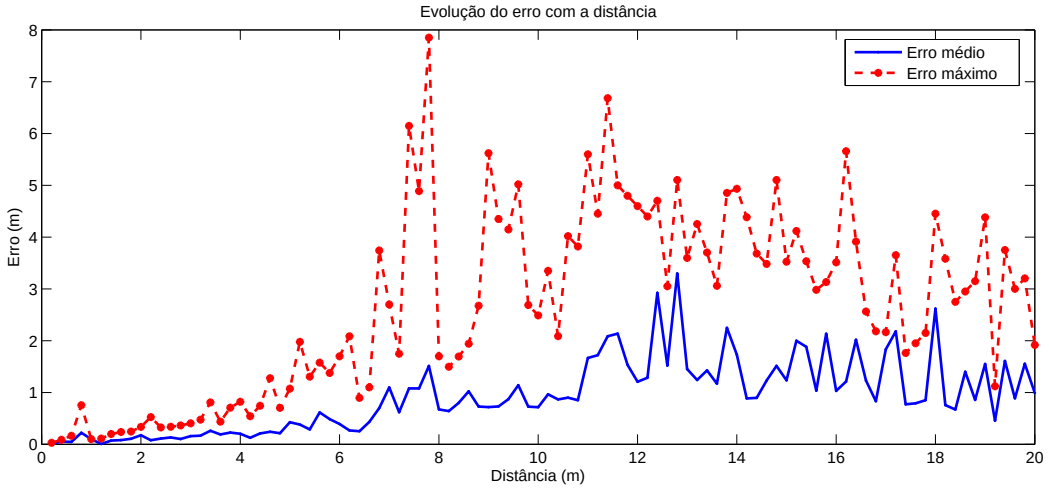

Figura 5.7: Evolução do erro com a distância.

5.3 Estimação da localização de um robô por trilateração

A localização de um robô consiste em identificar a sua posição num sistema de coordenadas. Em robótica móvel, a necessidade de um robô saber a sua localização é um requisito importante, mas de natureza complexa. A complexidade consiste em equipar robôs móveis com poder de processamento elevados, implementando filtros de localização de localização de frequências adequadas, ou equipamentos sensoriais de alta precisão, que podem ter custos insuportáveis principalmente quando se trata de investigação em *swarm*

robotics. Para combater eventuais faltas de precisão no *hardware,* existe a necessidade de desenvolver *software* eficiente, escolhendo métodos adequados tendo em conta os sensores disponíveis no robô.

A trilateração é um dos métodos de localização de robôs móveis, entre vários outros tais como: a odometria [MTS06] que baseia na integração incremental da posição ao longo do tempo; a triangulação que consiste em encontrar a posição do robô usando a medição de ângulos e as possíveis relações entre este e possíveis referências disponíveis; GPS e bússulas, sistemas de posicionamento por visão, etc. Ao contrário dos métodos de localização referidos, a trilateração utiliza a estimação da distância entre a posição desconhecida do robô aos outros, que servem de referência para a estimação da posição. Note que durante o texto será referida a posição do robô ou a posição do nó, mas este método é utilizado para a localização de qualquer objeto, usando a medição de distância e as possíveis relações entre este e as referência disponíveis, conhecidas *a priori*.

Quando a localização é feita no plano 2D são necessários pelo menos três pontos de referência para o robô estimar de forma precisa e unívoca a sua localização. Este método pode ser generalizado utilizando n pontos como referência, sendo neste caso designado por multilateração. A derivação matemática do problema (ver anexo I) parte do princípio que, se o robô conhecer a distância entre a sua localização (desconhecida) e um outro cuja posição é conhecida, a sua posição está situada em qualquer ponto sobre uma circunferência de centro na posição do robô conhecido e raio igual à distância conhecida. Como se pode notar, é uma informação paradoxal. É útil porque já é alguma informação, mas por outro lado não é uma informação precisa. No entanto, se este robô souber a distância a um outro, a sua localização será restrita a um ponto ou uma área entre dois pontos da intercessão entre as duas circunferências. O robô ainda não tem a sua localização precisa mas esta informação é mais restrita do que a anterior. A posição do robô ficará perfeitamente definida se este robô conhecer a distância a um terceiro, aí teoricamente a sua localização ficará restrita ao ponto de interseção entre as três circunferências, como indica a Figura 5.8. É importante referir que o método tem erros associados que serão detalhados mais à frente.

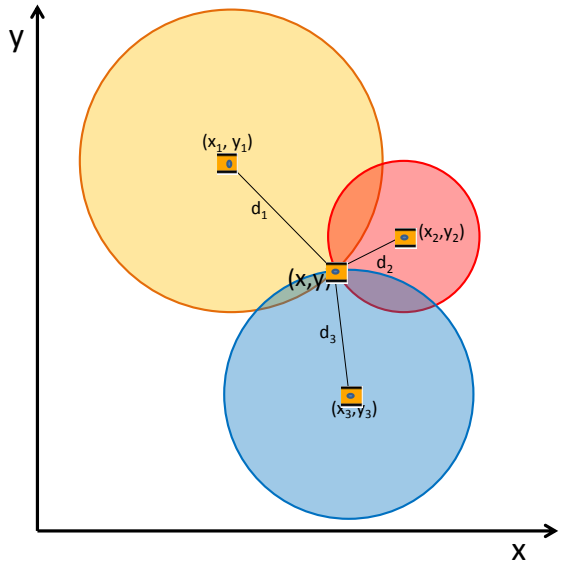

Figura 5.8: Diagrama de trilateração.

5.3.1 Experiências de trilateração

A experiência de trilateração foi realizada em ambiente *outdoor,* sem nenhuma obstrução, com os robôs distribuídos em forma de triângulo e um no centro, como indica a Figura 5.9. O robô do centro envia mensagens para os robôs que se encontram nos vértices do triângulo, e na mensagem de resposta é obtido o RSSI de cada um dos robôs dos vértices, a fim de estimar as distâncias entre o robô central e os dos vértices de forma a estimar a posição central. Este procedimento foi repetido, deslocando os robôs dos vértices até 10m com incrementos de 1m e registando 30 amostras por cada incremento.

A equação (5.2), onde (d_i) e $(rssi_i)$ representam a distância e o RSSI em relação ao robô i, respetivamente, foi obtida aproximando a curva da média dos dados de RSSI dos três robôs através de uma equação polinomial cúbica com $R^2 = 0.9915$ e os coeficientes $p_1 = -9.854e^{-005}$, $p_2 = -0.01332$, $p_3 = -0.6753$, $p_4 = -12.11$, usando os mesmos procedimentos para a obtenção da equação (5.1).

$$d_i(rssi) = p_1 rssi_i^3 + p_2 rssi_i^2 + p_3 rssi_i + p_4 \tag{5.2}$$

Note-se que a equação (5.1) obtida anteriormente para relacionar a distância com o RSSI

Figura 5.9: Imagem da experiência *"outdoor"* de trilateração.

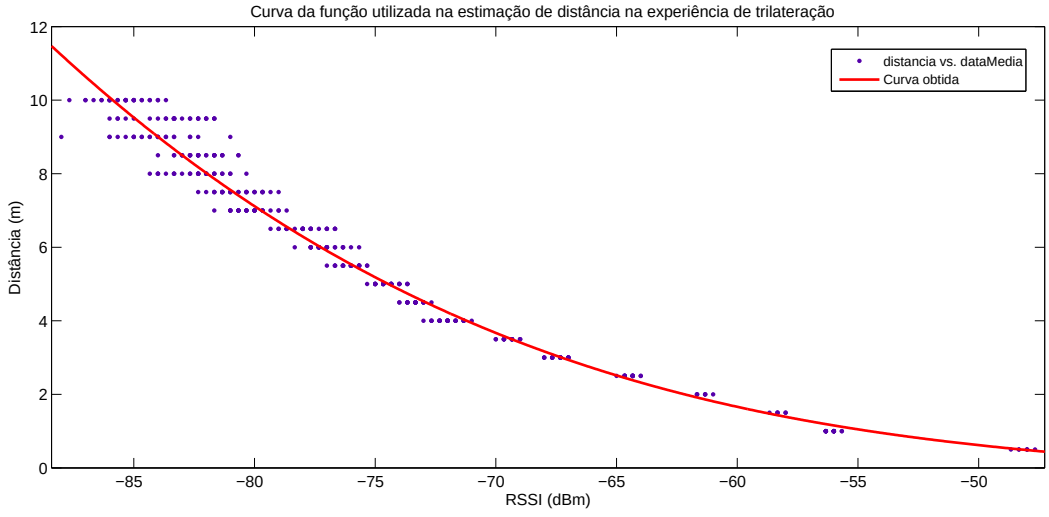

Figura 5.10: Curva de estimação de distância, obtida em ambiente *"outdoor"* e sem obstáculos

não é reutilizada, visto que os módulos XBee apresentam diferenças de fabrico. Assim sendo a equação (5.2) apresenta um compromisso nesta relação tendo em conta os módulos utilizados. Como foi referido na secção 5.3, a solução da trilateração seria teoricamente num ponto da interseção das três circunferências, mas como pode ser verificado na Tabela K.1, isso não acontece. A razão de não acontecer deve-se às medições de distâncias estarem associados a vários erros. Mais concretamente, utilizando a estimação de distância através do RSSI, o processo de localização é acompanhado por erros quando implementado em situações reais.

Para permitir a análise dos resultados experimentais como um todo, *i.e.*, os erros de posição envolvidos na trilateração para cada experiência, foi utilizada a metodologia das elipses dos mínimos quadrados para pontos 2D [FPF99], [CFP+12]. Isto permite analisar a precisão com base na área da elipse, isto é, a dispersão em relação à posição real. Para além disso, a exatidão dos dados também pode ser facilmente obtida com base na localização do centro da elipse em relação à posição real.

Como mostra a Figura 5.11, a 1 metro de distância a estimação é muito precisa e exata. No entanto, a partir dos 5 metros de distância, a precisão e a exatidão começam a diminuir, como pode ser verificado com o aumento da área da elipse e o distanciamento do centro da mesma face à origem. As elipses de erro para todos os casos de 1 a 10 metros são apresentadas em anexo (ver J.1).

Os erros obtidos podem ser causados por várias fontes, como ambiente da experiência, e o facto de que os sinais são confrontados com interferências, em situações reais. A interferência, influenciada pela temperatura, humidade ou obstruções podem ter impacto direto no sinal [ZFL+09], [GKL05]. Infelizmente estas interferências estão fora do controlo humano e podem contribuir para resultados inesperados. Outra fonte de erro é o modelo adotado para estimar a distância com base no RSSI. Neste trabalho foi utilizado o modelo de ajuste de curva (*curve fitting tool*). Para além disso, como foi referido atrás os próprios módulos apresentam diferentes valores de RSSI mesmo quando se encontram em situações semelhantes. Todos esses fatores contribuem para o erro de trilateração obtido. É importante notar que este método de localização assume que os robôs operam em espaço aberto, caso contrário a estimação da distância com base no RSSI é inválida.

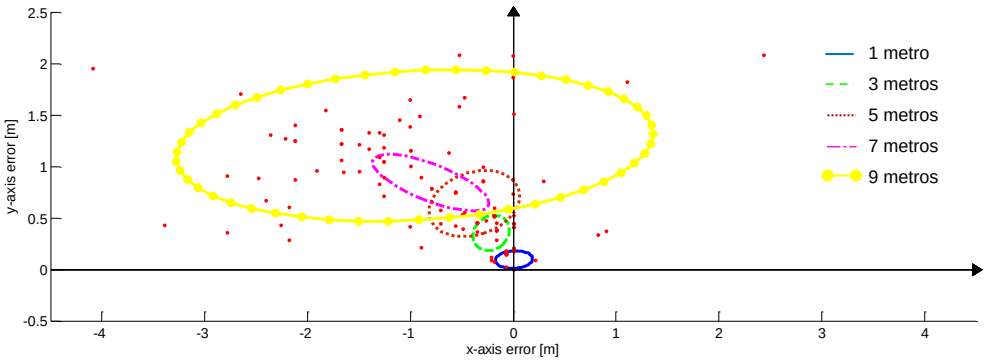

Figura 5.11: Evolução do erro de trilateração com a distância.

5.4 Análise do atraso na rede

Para medir o atraso na rede, o microcontrolador Arduino Uno, placa de processamento dos robôs TraxBot foi programado, para enviar mensagens para outro TraxBot. O módulo XBee série 2 do TraxBot que envia as mensagens foi configurado como Coordenador e o outro como Router. Nestas experiências o atraso do pacote é definido como a duração entre o envio de um pacote até a confirmação do mesmo por parte do recetor (ACK ao nível da camada MAC). São registadas 20 medidas para cada mensagem enviada, com diferentes tamanhos de dados. Em outras palavras o atraso é definido como o *Round-Trip Time* (RTT). Adicionalmente, para aferir o tempo de processamento das mensagens noutros nós, também se realizaram testes semelhantes, onde em vez de se aguardar pelo ACK o nó que envia aguarda por uma resposta semelhante ao seu "pedido", enviada pelo recetor original. Estas análises foram feitas para os casos *"single-hop"* e "*multi-hop*".

5.4.1 Caso *"single-hop"*

Neste caso foram enviadas mensagens do Coordenador para o Router, com diferentes tamanhos do campo de dados da trama de mensagem API do XBee série 2, medindo a duração entre o envio da mensagem até à confirmação da mesma por parte do recetor (ACK) ao nível da camada MAC, como mostra a Figura 5.12. As mensagens são constituídas por 12, 24, 36, 48, 60 e 72 bytes e o atraso no envio destas mensagens encontra-se na Tabela 5.1.

Tendo em conta que a taxa de transmissão do ZigBee é de 250Kbps, isto quer dizer que

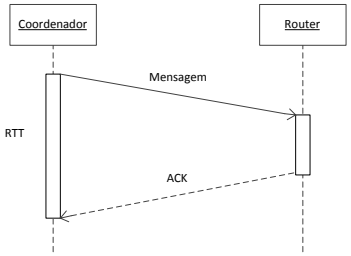

Figura 5.12: Diagrama de sequência.

Mensagens (bytes)	12	24	36	48	60	72
Média (ms)	63,9	74	88,2	102,2	116,9	130,3
Desvio Padrão (ms)	3,1	0,95	1,69	1,69	1,5	2,2

Tabela 5.1: Atraso das mensagens, com ACK ao nível da camada MAC.

se for enviada uma mensagem com 12 bytes, ignorando a *overhead* nas camadas de rede e MAC e o tempo de propagação o tempo de transmissão será de 12*8/250 = 0,384 msec. O tempo de transmissão é muito pequeno em relação ao RTT, isto mostra que a maior fatia do tempo é consumida no processamento das mensagens pelo microcontrolador do emissor e na comunicação entre este e o módulo XBee, que neste caso utiliza o *baud rate* definido por omissão nos módulos, 9600 bps. Para confirmar, que a maior fatia do RTT é no tempo de processamento do microcontrolador, realizou-se teste "pedido/resposta " discutido anteriormente.

Como era de esperar, houve um aumento significativo nos RTT das mensagens para aproximadamente o triplo em relação à experiência anterior. Isto verifica-se porque a mensagem é processada nos dois arduinos, processo que consome a maior fatia do tempo, enquanto que na outra experiência o tempo é medido com base na informação da confir-

Mensagens (bytes)	12	24	36	48	60	72
Média (ms)	187,2	238,8	289,25	339,1	390,65	441,4
Desvio Padrão (ms)	1,4	1,8	1,1	1,7	1,7	1,8

Tabela 5.2: Atraso das mensagens "pedido/resposta".

mação da mensagem pela camada MAC do módulo recetor.

5.4.2 Caso *"multi-hop"*

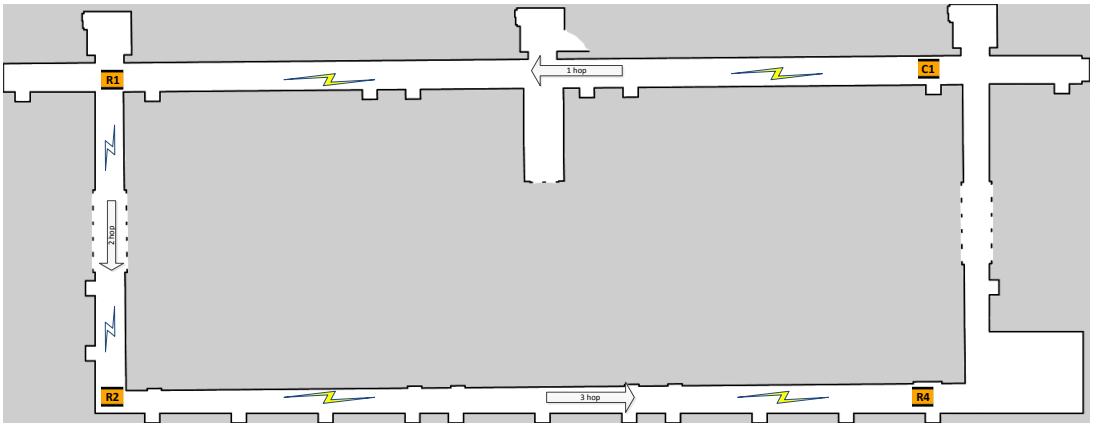

Figura 5.13: Mapa do piso 0 do ISR, com o posicionamento dos robôs.

No caso "multi-hop" as mensagens serão transmitidas do nó de origem para o nó de destino, passando por nós intermédios até chegar ao destino . Foram realizadas experiências "multi-hop" pedido/resposta. Neste caso, para além do tempo de processamento e transmissão no nó de origem, a mensagem sofrerá atrasos nos nós retransmissores. Esta experiência foi realizada nos corredores do piso 0 do ISR, colocando os robôs distribuídos da forma como indica a Figura 5.13. Foi fixado o Coordenador (C1) e o Router (R4) foi adicionado à rede. Em seguida foram enviadas mensagens do C1 para o R4, medindo o tempo entre o envio e a receção destas mesmas mensagens. O R4 foi afastado do C1 medindo o RSSI até perder a ligação. Neste ponto o RSSI apresentava um valor de -94dBm. Para retomar a comunicação entre C1 e R4 foi colocado um nó intermédio R1 e a comunicação foi estabelecida novamente passando pelo nó intermédio, isto é, comunicação multi-hop. Para aumentar o número de saltos *(hops)* para 3, o nó R4 foi afastado outra vez até perder a ligação com o C1, ligação essa que passava por R1. Foi colocado outro router R2 de modo a ser possível a comunicação entre o C1 e o R4, medindo o tempo entre o envio e receção das mensagens com diferentes tamanhos passando por nós intermédios, como indica a L.1.

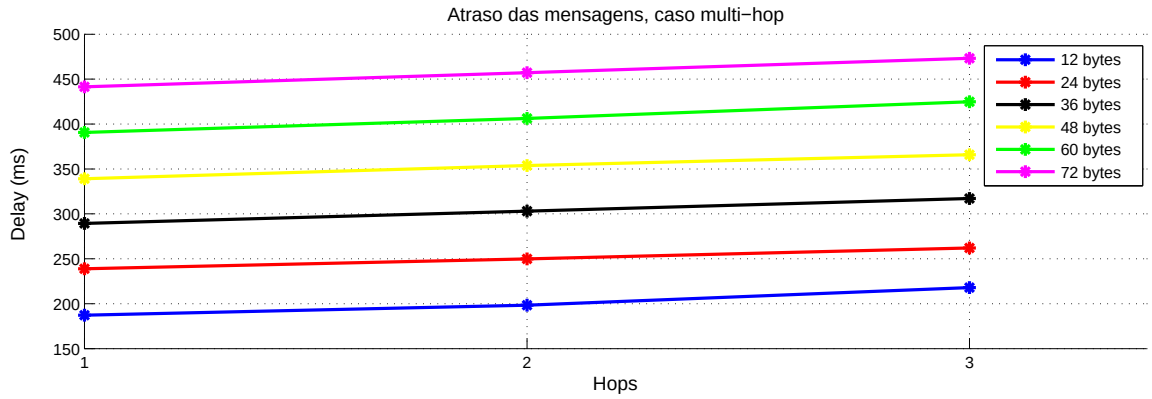

Figura 5.14: Atraso das mensagens.

Pela análise da Figura 5.14, pode-se verificar, que o tempo de retransmissão, é muito reduzido, uma vez que os nós intermédios não processam mensagens, apenas as reencaminham sem passá-las ao microcontrolador do Arduino.

5.5 Sumário

Neste capítulo foram realizadas algumas experiências para testar a comunicação *ad hoc* entre os robôs, medindo o atraso das mensagens para os casos *"multi-hop"* e *"single-hop"*. Nas mensagens recebidas foi medido o RSSI, estimando as distâncias aos outros nós estimando da posição de um robô tendo três nós como referência, utilizando o método de trilateração. No capítulo 6 faz-se o resumo do trabalho, apresenta-se as conclusões finais e sugestões para investigações futuras.

Capítulo 6

Conclusão

Esta dissertação incide sobre a comunicação *ad hoc* entre robôs móveis utilizando a tecnologia ZigBee, através da integração dos módulos XBee OEM RF série 2 nos TraxBot, expandindo as funcionalidades destes módulos para Arduino (baseada em C/C++) e a integração destas funcionalidades no driver ROS dos robôs. Como testes de desempenho da comunicação foram medidos o RSSI das mensagens recebidos pelos companheiros, estimando a distância aos mesmos. Utilizando a estimação da distância, procedeu-se à estimação de posição de um robô utilizando o método de trilateração, que apesar de revelar erros de diversas origens, é um método útil para assistir a localização de robôs dotados com estimação de posição por odometria . Para além disso foi analisado o atraso das mensagens para o caso *"single-hop"* e *"multi-hop"*, variando o campo de dados, e verificou-se que o que mais contribui no atraso das mensagens é o tempo de processamento das mensagens no microcontrolador e na comunicação entre este e o módulo XBee.

6.1 Contribuições

Este trabalho tem duas contribuições importantes. A primeira contribuição é a integração da comunicação *ad hoc* numa plataforma móvel de baixo custo, o TraxBot. Esta permite a realização de experiências, que exigem a necessidade de cooperação e interação entre multiplos robôs, como é o caso das investigações nas áreas de patrulhamento, *swarm robotics* e busca e salvamento, que têm sido levados a cabo no Laboratório de Robótica

Móvel (LRM) do Instituto de Sistemas e Robótica (ISR) em Coimbra, sem necessidade de infra-estruturas de rede, em locais de difícil implementação, ou em situações de emergência. A segunda contribuição é a expansão do driver ROS dos mesmos robôs, visto que o ROS é *framework*, *open source* muito utilizado atualmente e fornece bibliotecas e ferramentas para ajudar no desenvolvimento de *software* para a robótica de forma rápida e fácil sem a preocupação com questões inerentes à programação do *hardware.*

6.2 Trabalho futuro

Como trabalho futuro prevê-se uma série de desenvolvimentos nesta área. Os robôs TraxBot poderão beneficiar da estimação de posição por trilateração através do RSSI, fundindo esta informação com a sua própria estimação de odometria. Para além disto, está previsto o desenvolvimento de um driver ROS para comunicação *ad hoc* suprimindo a camada Arduino, de modo a reduzir o tempo de comunicação. Adicionalmente, devido ao trabalho desenvolvido nesta dissertação, os robôs móveis estão agora aptos para experiências de patrulhamento utilizando comunicação *ad hoc*. Finalmente, para experiências em *swarm robotics*, em ambiente laboratorial controlado, é possível adicionar nós estáticos de referência (*i.e.*, *landmarks*), para assistir na localização de pequenos robôs simplistas (que não são dotados de muitos sensores).

Apêndice A

Pilha Protocolar do ZigBee.

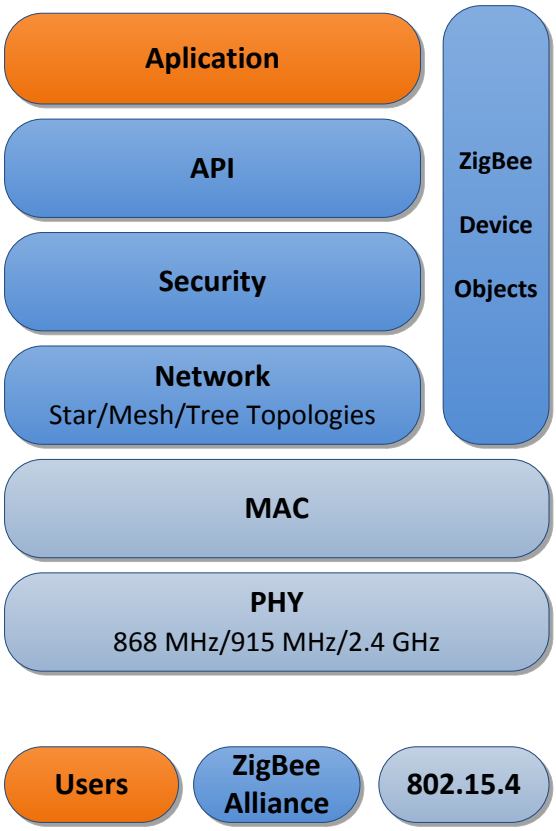

Figura A.1: Pilha Protocolar do ZigBee

Apêndice B

Tabela Comparativa entre ZigBee, Bluetooth e Wi-Fi.

	ZigBee	Bluetooth	Wi-Fi
Especificação IEEE	802.15.4	802.15.1	802.11/a/b/g
Aplicações	Controlo e monitorização	Alternativa aos cabos	Web, vídeo e e-mail
Banda de Frequência	868 MHz – Europa 915 MHz – América 2.4 GHz – Global	2.4 GHz	2.4GHz; 5GHz
Máxima taxa de dados	250 Kbps	1Mbps	54 Mbps
Alcance Nominal	10-100 Metros	10 Metros	100 Metros
Número de canais de Rádio Frequência (RF)	1/868 MHz 10/915 MHz 16/2.4 GHz	79	14(2.4 GHz)
Largura de banda do canal	0.3 MHz 0.6 MHz 2 MHz	1 MHz	22 MHz
Tipo de Modulação	BPSK (+ASK), O-QPSK	GFSK	BPSK, QPSK COFDM, CCK, M-QAM
Espalhamento de frequências	DSSS	FHSS	DSSS, CCK, OFDM
Topologia básica	Estrela	Piconet	BSS
Extensão da topologia básica	Topologia em árvore	Scatternet	ESS
Nº máximo de nós	65536	8	2007
Encriptação	AES (CRT, modo contador)	E0 stream cipher	RC4 stream cipher (WEP)
Autenticação	CBC – MAC	Shared secret	WPA2 (8022.11i)
Protecção de dados	16-bits CRC	16-bits CRC	32-bits CRC
Tempo de vida da bateria	Anos	Dias	Horas
Vantagens	Baixo consumo de energia Muitos dispositivos Baixo overhead Baixo custo	Amplamente aplicada à Electrónica de consumo	Domina as redes WLAN Amplamente disponíveis Alta taxa de transferência de dados
Desvantagens	Baixa taxa de transferência de dados	Consumo de energia, médio Poucos dispositivos na piconet	Alto consumo de energia

Tabela B.1: Tabela comparativa entre ZigBee, Bluetooth e Wi-Fi [FCP+12].

Apêndice C

Principais Características Xbee/Xbee-Pro Série2.

Especificação	XBee Série 2	XBee-Pro (Série 2)
Desempenho		
Alcance em áreas internas ou urbanas	40m	100m
Alcance em linha de visão (campo aberto)	120m	1.6km
Taxa de transferência	250kbps	250kbps
Potência de transmissão	2mW(+3dBm),boost mode enable 1.25mW(+1dBm),boost mode disable	60mW (+18 dBm) 10mW (+10 dBm) para variants internacionais
Requisitos energéticos		
Tensão de alimentação	2.1 - 3.6 V	3.0 - 3.4 V
Corrente de transmissão	40mA(@ 3.3 V, boost mode enable) 35mA(@ 3.3 V, boost mode disable)	295mA(@ 3.3 V)
Corrente de recepção	40mA(@ 3.3 V, boost mode enable) 38mA(@ 3.3 V, boost mode disable)	45mA(@3.3 V)
Geral		
Frequência	ISM 2.4 GHz	ISM 2.4GHz
Dimensões	2.438cm x 2.761cm	2.438cm x 3.294cm
Temperatura de operação	-40 a 85ºC(Industrial)	-40 a 85ºC(Industrial)
Antena	Chip integrado, Whip, RPSMA ou U.FL connector	Chip integrado, Whip, RPSMA U.FL connector
Redes e Segurança		
Topologia de rede	Point-to-point, point-to-multipoint, peer-to-peer & Mesh	Point-to-point, point-to-multipoint, peer-to-peer & Mesh
Número de canais	16 Direct Sequence Channels	13 Direct Sequence Channels
Opções de endereçamento	PAN ID e endereços, Cluster IDs e Endpoints (opcionais)	PAN ID e endereços, Cluster IDs e Endpoints (opcionais)

Tabela C.1: Principais características XBee/XBee-Pro Série 2 [I07].

Apêndice D

Software X-CTU.

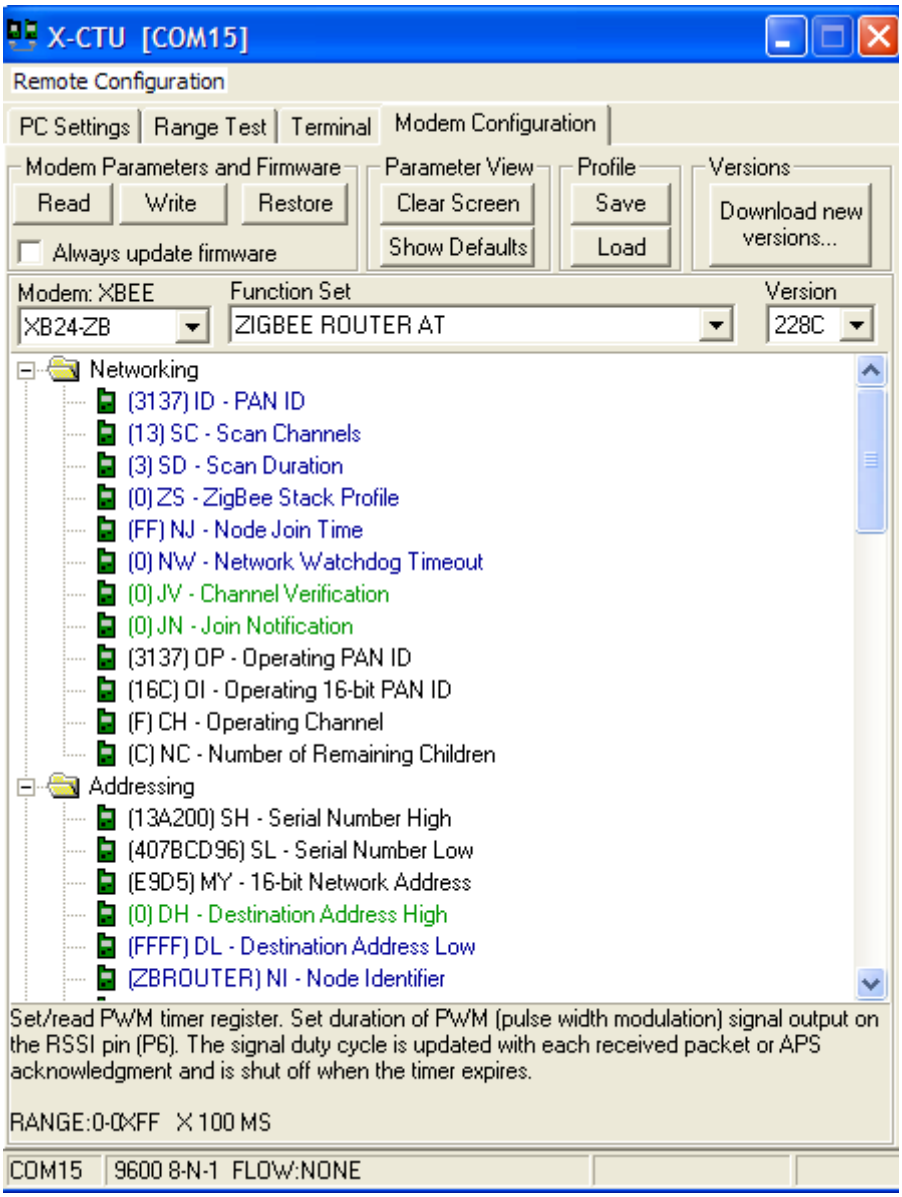

Figura D.1: Software XCTU.

Apêndice E

Código de configuração dos módulos sem retirar o microcontrolador do Arduino Uno.

Algoritmo E.1 Código de configuração dos módulos sem retirar o microcontrolador do Arduino Uno.

```
void setup{}
void loop{}
```

Apêndice F

Classe *XbeeNode* e as suas dependências.

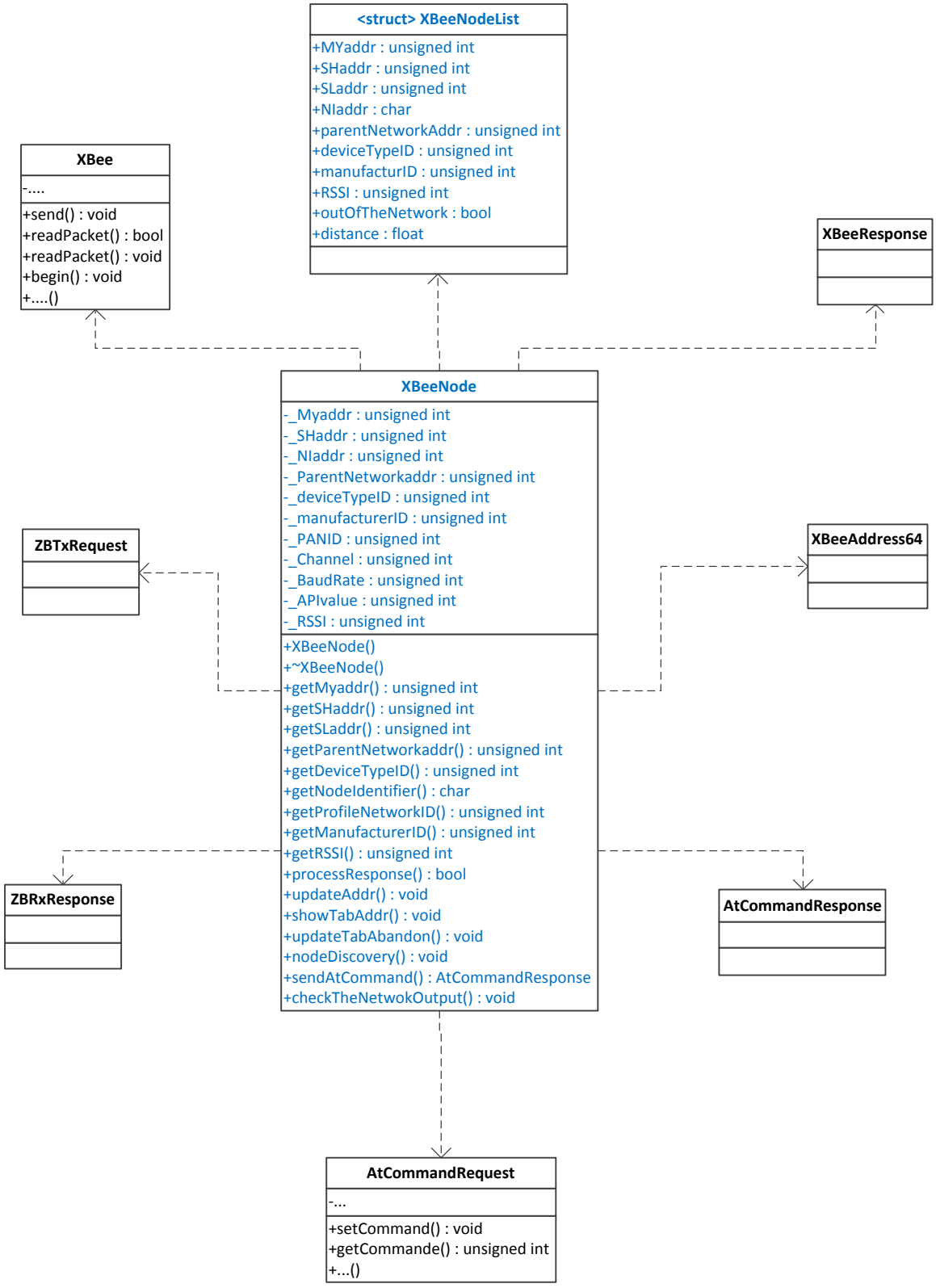

Figura F.1: Classe XBeeNode e suas dependências.

Apêndice G

Fluxograma da Função *nodeDiscovery*.

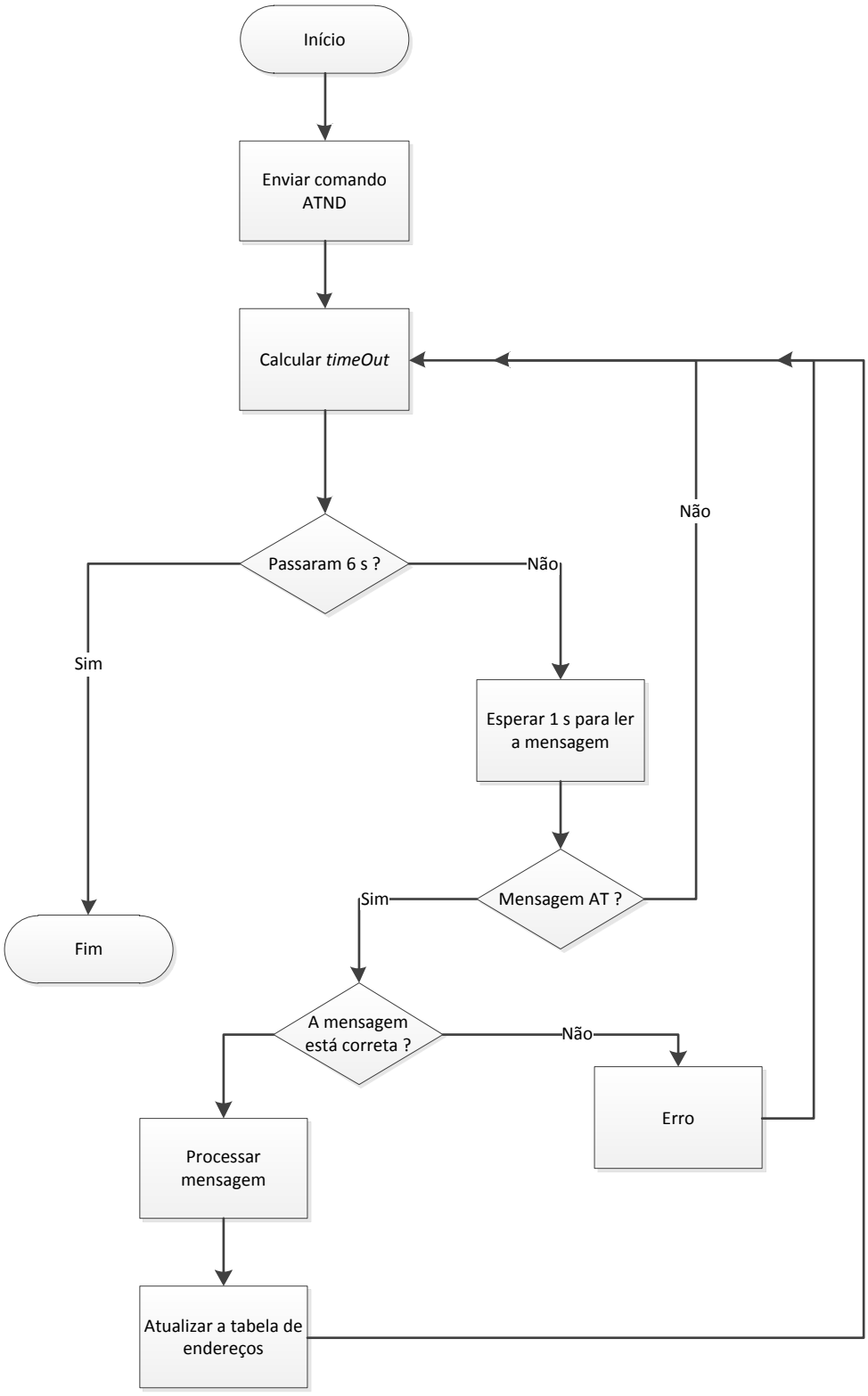

Figura G.1: Fluxograma da função *nodeDiscovery*.

Apêndice H

Firmware do Driver ROS, residente no Arduino Uno.

Algoritmo H.1 Firmware do driver ROS, residente no Arduino Uno

```
#include <Biblioteca do Omni3MD>
#include <Biblioteca do EEPROM>
#include <Biblioteca do TraxBot>
#include <Biblioteca do RobotSerialComm>
#include <Biblioteca do XBee_v1.h> // Biblioteca alterada
#include <Biblioteca do XBeeNode.h>
#include <Biblioteca do SoftwareSerial_v1.h>//Biblioteca alterada
#include <Bibliotecas Standards>
Setup Functions(); // PID motor gains, using ports, encoders scale, set I2C connection,...
Streaming Functions();
....
Main loop():
Select given action:
....
//Funções da classe
Informações da rede:
Obter o endereço da PAN;
Obter o canal de seleção;
Obter a duracção de seleção do canal;
Obter o tempo de associação de um nó;
Obter o canal de operação;
Obter o número de hops da mensagem broadcast;
Obter o timeout do comando do Node discovery(ND);

Informações do módulo:
Obter a versão do firmware;
Obter a versão do hardware;
Obter informação da tensão de alimentação;

Informações de endereçamento:
Obter a parte mais significativa do número de série;
Obter a parte menos significativa do número de série;
Obter o próprio endereço de rede;

Verificar nós que abandonam a rede;
Procurar novos nós na rede;
Enviar mensagem unicast;
Enviar mensagem Broadcast;
Receber mensagem;
```

Apêndice I

Dedução Matemática da Trilateração

Sendo (x_i, y_i) as posições conhecidas dos pontos de referência e (d_i) as distâncias estimadas entre estes e a posição desconhecida, obtemos o sistema de equações não lineares($I.1$) [LR03].

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 = & d_1^2 \\ \vdots & \vdots \\ (x_n - x)^2 + (y_n - y)^2 = & d_n^2 \end{cases} \tag{I.1}$$

Onde ($i = 1 \cdots n$) e (x, y) é a posição do nó que se pretende estimar. O sistema pode ser linearizado subtraindo a última equação nas $(n-1)$equações, obtendo o sistema de equações linear ($I.2$).

$$\begin{cases} x_1^2 - x_n^2 - 2\,(x_1 - x_n)\,x + y_1^2 - y_n^2 - 2\,(y_1 - y_n)\,y = & d_1^2 - d_n^2 \\ \vdots & \vdots \\ x_{n-1}^2 - x_n^2 - 2\,(x_{n-1} - x_n)\,x + y_{n-1}^2 - y_n^2 - 2\,(y_{n-1} - y_n)\,y = & d_{n-1}^2 - d_n^2 \end{cases} \tag{I.2}$$

Reorganizando de modo a separar as incógnitas obtem-se o sistema $Ax = b$, dada por ($I.3$) ,

$$\begin{bmatrix} 2\,(x_1 - x_n) & 2\,(y_1 - y_n) \\ \vdots & \vdots \\ 2\,(x_{n-1} - x_n) & 2\,(y_{n-1} - y_n) \end{bmatrix} \begin{bmatrix} x \\ \\ y \end{bmatrix} = \begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 + d_n^2 - d_1^2 \\ \vdots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 + d_n^2 - d_{n-1}^2 \end{bmatrix} \tag{I.3}$$

Se for restringido a 3 referências ($n = 3$), que é o caso da trilateração (ver Figura 5.8) obtêm-se o sistema ($I.4$),

$$\begin{bmatrix} 2\,(x_1 - x_3) & 2\,(y_1 - y_3) \\ 2\,(x_2 - x_3) & 2\,(y_2 - y_3) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_1^2 - x_3^2 + y_1^2 - y_3^2 + d_3^2 - d_1^2 \\ x_2^2 - x_3^2 + y_2^2 - y_3^2 + d_3^2 - d_2^2 \end{bmatrix} \tag{I.4}$$

O sistema ($I.4$), é um sistema de equações lineares, pelo que se pode utilizar, o método do

cálculo da matriz inversa para aproximar a solução do sistema. No caso do sistema $(I.4)$a matriz A é quadrada $(n \times n)$, então se o determinante for diferente de zero $(det(A) \neq 0)$ a solução do sistema será calculada utilizando a equação $(I.5)$. Os três pontos de referência para o caso de trilateração terão de ser não colineares.

$$Ax = b \Leftrightarrow \hat{x} = A^{-1}b \tag{I.5}$$

Caso a matriz não for quadrada $(n \times m)$que é o caso do sistema $(I.3)$ quando n for maior que 3, deve-se recorrer ao cálculo da pseudo-inversa $\left(\left(A^T A\right)^{-1} A^T\right)$, que é substituida na equação $(I.5)$, obtendo-se a equação $(I.6)$. Em casos em que não é possível calcular a inversa da matriz o sistema não tem solução.

$$Ax = b \Leftrightarrow \hat{x} = \left(A^T A\right)^{-1} A^T b \tag{I.6}$$

Apêndice J

Elipses de Erro da posição estimada por trilateração.

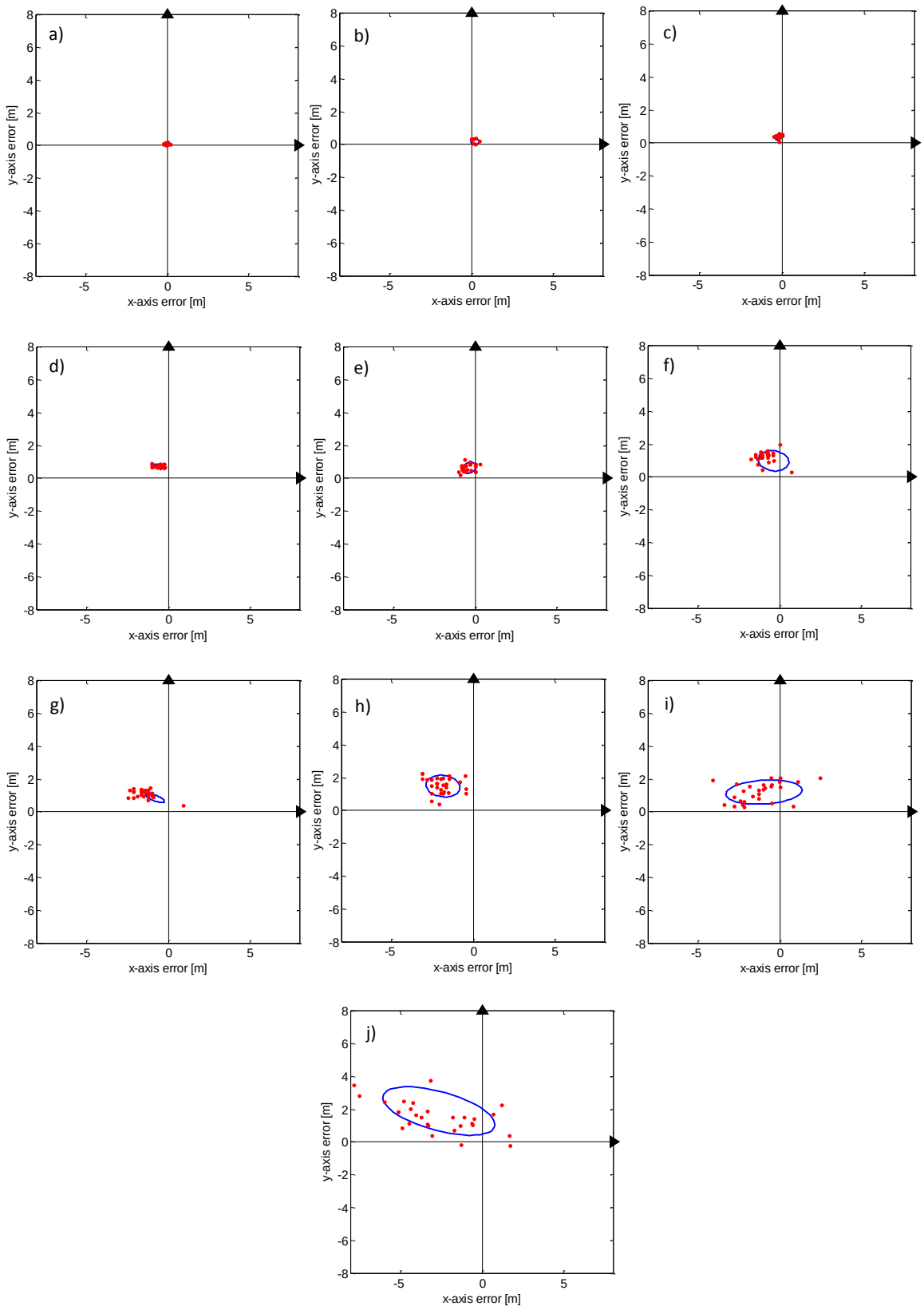

Figura J.1: Elipses de Erro da posição estimada por trilateração. De a) a j) estão representadas as estimações de 1 a 10 metros consecutivamente.

Apêndice K

Resultados numéricos de trilateração.

Posição real (m)		Posição Estimada (m)		Erro (m)	Erro (%)	Distância Estimada (m)			Distância real (m)
X	Y	X	Y			Robô1	Robô2	Robô3	
0	0	-0.06	0.15	0.16	16	0.67	1.03	0.93	1.00
0	0	0.12	0.34	0.36	18	1.48	2.04	2.21	2.00
0	0	-0.35	0.45	0.57	19	2.38	3.44	2.97	3.00
0	0	-0.69	0.68	0.97	24.50	3.2	4.85	3.96	4.00
0	0	-0.73	0.75	0.93	18.6	4.24	5.9	5.18	5.00
0	0	-1.73	1.34	1.52	25.33	4.85	7.56	6.69	6.00
0	0	-1.67	1.22	2.07	29.57	5.18	8.52	6.28	7.00
0	0	-2.22	1.64	2.75	34.37	6.69	10.71	8.03	8.00
0	0	-1.64	1.13	1.98	22	8.52	11.31	9.3	9.00
0	0	-3.01	1.25	3.26	32.6	9.56	13.29	9.57	10.00

Tabela K.1: Resultados da Trilateração.

Apêndice L

Atraso das mensagens no caso Multi-Hop.

Tamanho da mensagem (bytes)	12	24	36	48	60	72
1 hop						
Média (ms)	187,2	238,8	289,25	339,1	390,65	441,4
Desvio Padrão (ms)	1,4	1,80	1,13	1,72	1,71	1,88
2 hops						
Média (ms)	198,4	249,85	303	353,8	406,2	457,15
Desvio Padrão (ms)	1,98	2,55	3,88	2,65	3,89	2,65
3 hops						
Média (ms)	218	262	317,15	365,85	424,8	473,15
Desvio Padrão (ms)	9,31	4,07	4,89	2,30	10,92	3,74

Tabela L.1: Atraso das mensagens, no caso multi-hop.

Bibliografia

[A12] A. Araújo, "*ROSint - Integration of a mobile robot in ROS architecture*", A Dissertation presented for the degree of Master of Science in Electrical and Computer Engineering, Coimbra, July 2012.

[APC+12] A. Araújo, D. Portugal, M. Couceiro, C. Figueiredo and R. P. Rocha. *TraxBot: "Assembling and Programming of a Mobile Robotic Platform*", in Proceedings of the 4th International Conference on Agents and Artificial Intelligence (ICAART'2012), Vilamoura, Algarve, Portugal, February 6-8, 2012.

[Ard10] Arduino Uno, 2010, http://arduino.cc/en/Main/ArduinoBoardUno (Aug. 2012).

[Atmel09] Atmel, 2009, http://www.atmel.com/Images/doc8161.pdf (Aug. 2012).

[BV10] J. Biswas and M. Veloso, "*Wi-Fi localization and navigation for autonomous indoor mobile robot*s," Robotics and Automation (ICRA), 2010.

[B12] Bluetooth Technology, http://www.bluetooth.com/(Aug. 2012).

[CFL+12] Micael S. Couceiro, Carlos M. Figueiredo, J. Miguel A. Luz, Nuno M. F. Ferreira & Rui P. Rocha. *"A Low-Cost Educational Platform for Swarm Robotics*", International Journal of Robots, Education and Art, IJREA, Volume 2, Issue 1, February, pp. 1-15, 2012.

[CFP+12] Micael S. Couceiro, Carlos M. Figueiredo, David Portugal, Rui P. Rocha and Nuno M. F. Ferreira, *"Initial Deployment of a Robotic Team - A Hierarchical Approach Under Communication Constraints Verified on Low-Cost Plat-*

forms", In Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'12), October 7-12, Vilamoura, Portugal, 2012 (Accepted).

[CMR11] C. Ramya, M. Shanmugaraj and R. Prabakaran, "Study on ZigBee technology," In Proceedings of the 2011 3rd International Conference on Electronics Computer Technology (ICECT), vol. 6, pp. 297–301, 2011.

[Chopin12] "*Cooperation between Human and rObotics teams in catastroPhic INcidents*", R&D project of the Mobile Robotics Lab of the Institute of Systems and Robotics at University of Coimbra, in Portugal, http://chopin.isr.uc.pt (Aug. 2012).

[CSI07] Doyletech Corporation, D.R. Senik and Associates Inc. "*Wireless Technology Roadmap: 2006-2016 Mapping the Crucial Skills Required to Make Canada a Global Wireless Leader*", Technical Report, 2007.

[CRF+12] Micael S. Couceiro, Rui P. Rocha, Carlos M. Figueiredo, J. Miguel A. Luz & N. M. Fonseca Ferreira. "*Multi-Robot Foraging based on Darwin's Survival of the Fittest* ", In Proc. of IROS'12 - IEEE/RSJ International Conference on Intelligent Robots and Systems, October 7-12, Vilamoura, Portugal, 2012 (Accepted).

[CPP11] "ceral_port" Package, http://www.ros.org/wiki/cereal_port (Aug. 2012).

[DBA12] "W1039B030 Dual Band WLAN Antenna", http://www.farnell.com/datasheets/1493535.pdf (Aug. 2012)

[DI12] Digi International, http://www.digi.com/support/productdetail (Aug. 2012).

[ER12] Ethernet References, http://www.juniper.net/techpubs/software/erx/junose93/swconfig-physical/ethernet-references.html (Aug. 2012).

[FLW+05] W. H. Fan, Y. H. Liu, F. Wang, and X. P. Cai, "*Multi-robot formation control using potential field for mobile ad-hoc networks*," 2005 IEEE International Conference on Robotics and Biomimetics - ROBIO, pp. 133-138, 2005.

[FCP+12] A. Fernandes, M. Couceiro, D. Portugal and R.P. Rocha, "*WirelessTeams: Comparação de Tecnologias Sem Fios em Equipas de Robôs Móveis* ", Relatório técnico, Coimbra, Fevereiro de 2012.

[fitTool] Curve Fitting Toolbox Fit curves and surfaces to data using regression, interpolation, and smoothinghtt, //www.mathworks.com/products/curvefitting/ (Aug. 2012).

[FPF99] A. W. Fitzgibbon, M. Pilu and R. B. Fisher, "*Direct Least-squares fitting of ellipses*", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 5, pp. 476-480, 1999.

[GJS+07] E. Garcia, M. A. Jimenez, P. G. de Santos, and M. Armada, "*The evolution of robotics research, From Industrial Robotics to Field and Service Robotic*", no. March, pp. 90-103, 2007.

[G97] D. Gage, "*Network protocols for mobile robot systems*," Proceedings of SPIE, vol. 3210, pp. 107-118, 1997.

[GV06] J. C. Giacomin and F. H. Vasconcelos, "*Measurement Quality of Signal Strength in the Wireless Sensor Network's Communications: a Physical Layer Approach*", INFOCOMP Journal of Computer Science, April, 2006.

[GKL05] J. Thelen, D. Goense, and K. Langendoen, " *Radio wave propagation in potato fields*", 1st Workshop on Wireless Network, 2005.

[H09] J. Hu, "*Cooperation in mobile ad hoc networks*",Guide to Wireless Ad Hoc Networks, 2009.

[I07] D. International, "XBee TM Series 2 OEM RF Modules - Product Manual v1.x.2x - ZigBee Protocol For OEM RF Module Part Numbers: XB24-BxIT-00x", Julho.2007

[IEE12] IEEE 802 LAN/MAN Standards Committee, http://ieee802.org/(Aug. 2012)

[IIL09] T. Ieee, T. Ieee, and L. Mee, "IEEE 802 LAN / MAN Standards Commitee," Info, no. 802, pp. 2009-2011, 2009.

[K03] R. J. Kennelly, *IEEE standards for physical and data communications*, vol. 30, no. 2., pp. 172-5, 2003.

[K07] A. E. Turgut, F. G¨ok¸ce, C¸h. Elikkanat, L. Bayındır, and E. S¸ahin, "*Kobot: A mobile robot designed specifically for swarm robotics research*", Mar. 2007.

[L98] Lyshevski, S.E.; *Analytic solutions to Maxwell's equations: sinusoidal steady-state and transient space-time problems in transverse magnetic and transverse electric field analysis*, Mathematical Methods in Electromagnetic Theory, 1998. MMET 98. 1998 International Conference on , vol.1, no., pp.88-91 vol.1, 2-5 Jun 1998.

[LAS07] J.-S. Lee, Y.-W. Su, and C.-C. Shen, "*A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi*," IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society, pp. 46-51, 2007.

[LYC06] Y. Liu, S. Yu, and W. Chen, "*Wireless Communication Technology Based on Bluetooth and Its Application to a Manipulator*," Industrial Informatics, 2006 IEEE, pp. 1251-1256, 2006.

[LR03] K. Langendoen and N. Reijers, "*Distributed localization in wireless sensor networks: a quantitative comparison*", Computer Networks, vol. 43, no. 4, pp. 499-518, Nov. 2003.

[LVH+05] Y. Lin, P. Vernaza, J. Ham, and D. D. Lee, "*Cooperative relative robot localization with audible acoustic sensing*", 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3764-3769, 2005.

[MS10] A. L. G. Modesto and M. H. K. Sampaio, "*Controle de Sistemas Embarcados Através de Bluetooth Aplicado a Robótica Móvel com o Selvabot*," Engenharia de Computação em Revista, vol. 1, no. 4, pp. 1-4, 2010.

[MM08] J. Misic and V. B. Misic, *Wierless Personal Area Network : Performance, Interconnections and Security with IEEE 802.15.4*. Wiley, 2008.

[ML08] C.-M. Chao and K.-H. Lu, "*Load Awareness Multi-Channel MAC Protocol Design for Ad Hoc Networks*," 2008 IEEE International Conference on Sensor

Networks, Ubiquitous, and Trustworthy Computing (sutc 2008), pp. 36-43, Jun. 2008.

[MTS06] A. Martinelli, N. Tomatis, and R. Siegwart, "*Simultaneous localization and odometry self calibration for mobile robot*", Autonomous Robots, vol. 22, no. 1, pp. 75-85, Sep. 2006.

[Osi12] "*Open Systems Interconnection*", Wikipedia. Wikimedia Foundation: http://en.wikipedia.org/wiki/Open_Systems_Interconnection (Aug. 2012).

[OHK11] T. Ohta, T. Hashimoto, and Y. Kakuda, "*Self-Organizing Real-Time Service Dissemination and Collection Using Mobile Agents for Mobile Ad Hoc Networks*," 2011 14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops, pp. 199-206, Mar. 2011.

[PK09] K. Pahlavan and P. Krishnamurthy, *Networking Fundamentals: Wide, Local and Personal Area Communications*, vol. 7, no. 1. , pp. 1-641,2009.

[PR11] D. Portugal and R.P. Rocha, "*On the Performance and Scalability of Multi-Robot Patrolling Algorithms*". In Proceedings of the 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR'2011), 50-55, Kyoto, Japan, November 1-5, 2011.

[PR12] D. Portugal and R.P. Rocha, *Measuring Variables Effect to Statistically Model the Multi-Robot Patrolling Problem by means of ANOVA*, in Luis M. Camarinha-Matos, Ehsan Shahamatnia, Gonçalo Nunes (editors), Technological Innovation for Value Creation, Vol. 372, pp. 199-206, Proc. of 3rd Doctoral Conference on Computing, Electrical and Industrial Systems (Do- CEIS 12), Costa da Caparica, Lisbon, Portugal, Springer Berlin Heidelberg, Feb. 27-29, 2012.

[QGC+09] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Y. Ng,. *"ROS: an open-source Robot Operating System*", in Proc. Open-Source Software workshop of the International Conference on Robotics and Automation (ICRA 2009), Kobe, Japan, May, 2009.

[RC12] ROS concepts: http://www.ros.org/wiki/ROS/Concepts (Aug. 2012).

[R06] R. Rocha, *"Building Volumetric Maps with Cooperative Mobile Robots and Useful Information Sharing - A Distributed Control Approach based on Entropy"*, PhD Thesis, Supervisor:Jorge M.M.Dias (FCTUC) and Adriano da Silva Carvalho (FEUP), Faculdade de Engenharia da Universidade do Porto, May, 2006.

[SK10] J. Sangeetha and S. Kumar, "*A comparative study on Wi-Fi and WiMAX networks*", in Computational Intelligence and Computing Research (ICCIC), IEEE International Conference on, pp. 1–5, 2010.

[SMK+07] M. Simec, I. Mica, J. Kakalek, and R. Burget, "*Bandwidth Efficiency of Wireless Networks* ", Wireless Networks, pp. 1-15, 2007.

[SBF+06] S. Ondĝej, B. Zdenčk, F. Petr, and H. Ondĝej, "*ZigBee Technology and Device Design 1 Introduction 2 ZigBee Network*," Network, 2006.

[S03] J. Schiller, "*Mobile Communications*", 2nd ed. Edinburgh Gate: pp. 1-513, 2003.

[SM06] S. Safaric and K. Malaric, "*ZigBee wireless standard*," In Proceedings of the 48th International Symposium ELMAR - Multimedia Signal Processing and Communications, pp. 259–262, June 2006.

[SGK09] S. Gaurav, A. Ganesh and P. Key, "Performance Analysis of Contention Based Medium Access Control Protocols." In IEEE Transactions on Information Theory No. 55, Vol. 4, pp. 1665-682, 2009.

[SCM+11] M. J. Segura, F. a Auat Cheein, J. M. Toibero, V. Mut, and R. Carelli, "*Ultra wide-band localization and SLAM: a comparative study for mobile robot navigation*", Sensors (Basel, Switzerland), vol. 11, no. 2, pp. 2035-55, Jan. 2011.

[SCS11] "*serial_ communication*" Stack, http://www.ros.org/wiki/serial_ communication (Aug. 2012).

[WZA03] Z. Wang, M. Zhou and N. Ansari, "*Ad-hoc robot wireless communication*", In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, vol. 4, pp. 4045–4050, Washington D.C., October, 2003.

[ZFL+09] Y. Zhang, Z. Fang, R. Li, and W. Hu, "*The Design and Implementation of a RSSI-Based Localization System*", 5th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1-4, Sep. 2009.

[ZKS08] F. Zeiger, N. Kraemer, and K. Schilling, "*Commanding mobile robots via wireless ad-hoc networks — A comparison of four ad-hoc routing protocol implementations*", in IEEE International Conference on Robotics and Automation, pp. 590-595, 2008.

[ZA12] ZigBee Alliance, http://www.zigbee.org/ (Aug. 2012).