CISTER
Research Centre in
Real-Time & Embedded
Computing Systems

PhD Thesis

Communications and Localisation for Cooperating Autonomous Mobile Robots

Luís Oliveira

CISTER-TR-181121

2016/02/18

Communications and Localisation for Cooperating Autonomous Mobile Robots

Luís Oliveira

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail:

http://www.cister.isep.ipp.pt

Abstract

Heterogeneous teams of cooperating robots are ideal candidates for applications where the presence ofhumans is impossible or should be avoided, not only ensuring the safety of the people they replace, butalso allowing the execution of tasks otherwise impossible. Moreover, if the team is heterogeneous, i.e.robots have di_erent sensors/actuators, it is possible to optimise the use of such components reducingassociated costs while maintaining full functionality. However, one the most attractive reasons for usingsuch cooperating teams is the possibility of maximising the utility of the whole system; e.g. increasingthe e_ectiveness of surveillance by performing cooperative sensing, improving the rate of coverage insearch and rescue missions, and performing motion coordination for the transport of large parts.There are several key factors that enable such cooperation; in this work we will focus on two of them:a)exchanging information; b)tracking relative positions.For factor a), we propose a new solution for communicating data amongst a team of heterogeneousrobots, which due to mobility requirements and dynamic team composition, with nodes joining andleaving on-line, is done through a wireless medium. Thus, we developed a new wireless communicationprotocol with support for dynamic group membership, based on fully decentralised proximity (ad-hoc)communications.However, to improve the e_ciency of using the channel bandwidth the protocol uses controlledtransmissions, synchronized in a circular temporal framework of the Time Division Multiple Access(TDMA) type. One of the main innovations of this protocol is achieving distributed synchronizationwithout using a global clock while tolerating uncontrolled tra_c that is external to the team. Resortingto a consensus technique we show that this method, based on the reception instants of the messagesexchanged among the team members, assures synchronization in a vast range of operational scenarios.Moreover, the developed protocol supports multi-hop forwarding using a new algorithm that reduces theend to end delivery delay and, most importantly, drastically reduces the variability of such delay, isolatingthe communication requirements of each node throughout the whole network, significantly improving itstemporal behaviour and analysability.For the factor b) above, we developed new methods to derive relative positions from local RF communications.In particular, we use both the signal strength and time of flight paradigms and with themwe propose an innovative hybrid solution capable of improving the performance of any of them alone.These ranging paradigms are then used to produce pairwise distance measurements with which we derivethe robots positions, relative to each other. Finally, we introduce another innovation that allows usdetermining the confidence of the relative positions determined with this method as a function of theconfidence of the pairwise distances between robots.

© CISTER Research Center

U. PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Communications and Localisation for Cooperating Autonomous Mobile Robots

Luís Filipe Nunes Quaresma de Oliveira

Supervisor: Professor Doutor Luis Miguel Pinho de Almeida

Co-Supervisor: Professor Doutor Pedro Manuel Urbano de Almeida Lima

Programa Doutoral em Engenharia Electrotécnica e de Computadores

February, 2016

© Luís Filipe Nunes Quaresma de Oliveira: February, 2016

Faculdade de Engenharia da Universidade do Porto

Communications and Localisation for Cooperating Autonomous Mobile Robots

Luís Filipe Nunes Quaresma de Oliveira

Dissertation submitted to Faculdade de Engenharia da Universidade do Porto
to obtain the degree of

Doctor Philosophiae in Electronic & Computer Engineering

February, 2016

Este trabalho foi parcialmente financiado pela da FCT (Fundação para a Ciência e a Tecnologia) através da bolsa de Doutoramento SFRH/BD/74292/2010 e dos projetos PCMMC PTDC/EEA-CRO/100692/2008, CodeStream PTDC/EEI-TEL/3006/2012, PICC (acordo bilateral CAPES-FCT), e GPES (Instituto de Telecomunicações).

Aos meus pais e irmãos,
aos amigos,
e aos colegas.

Abstract

Heterogeneous teams of cooperating robots are ideal candidates for applications where the presence of humans is impossible or should be avoided, not only ensuring the safety of the people they replace, but also allowing the execution of tasks otherwise impossible. Moreover, if the team is heterogeneous, i.e. robots have different sensors/actuators, it is possible to optimise the use of such components reducing associated costs while maintaining full functionality. However, one the most attractive reasons for using such cooperating teams is the possibility of maximising the utility of the whole system; e.g. increasing the effectiveness of surveillance by performing cooperative sensing, improving the rate of coverage in search and rescue missions, and performing motion coordination for the transport of large parts.

There are several key factors that enable such cooperation; in this work we will focus on two of them: a)exchanging information; b)tracking relative positions.

For factor a), we propose a new solution for communicating data amongst a team of heterogeneous robots, which due to mobility requirements and dynamic team composition, with nodes joining and leaving on-line, is done through a wireless medium. Thus, we developed a new wireless communication protocol with support for dynamic group membership, based on fully decentralised proximity (ad-hoc) communications.

However, to improve the efficiency of using the channel bandwidth the protocol uses controlled transmissions, synchronized in a circular temporal framework of the Time Division Multiple Access (TDMA) type. One of the main innovations of this protocol is achieving distributed synchronization without using a global clock while tolerating uncontrolled traffic that is external to the team. Resorting to a consensus technique we show that this method, based on the reception instants of the messages exchanged among the team members, assures synchronization in a vast range of operational scenarios. Moreover, the developed protocol supports multi-hop forwarding using a new algorithm that reduces the end to end delivery delay and, most importantly, drastically reduces the variability of such delay, isolating the communication requirements of each node throughout the whole network, significantly improving its temporal behaviour and analysability.

For the factor b) above, we developed new methods to derive relative positions from local RF communications. In particular, we use both the signal strength and time of flight paradigms and with them we propose an innovative hybrid solution capable of improving the performance of any of them alone. These ranging paradigms are then used to produce pairwise distance measurements with which we derive the robots positions, relative to each other. Finally, we introduce another innovation that allows us determining the confidence of the relative positions determined with this method as a function of the confidence of the pairwise distances between robots.

Resumo

Equipas de robots heterogéneos a cooperar entre si são os candidatos ideais para atuar em cenários onde a presença de humanos é impossível ou deve ser evitada. A utilização destas equipas, não só permite garantir a segurança das pessoas que eles substituem, mas também permite a execução de tarefas impossíveis de realizar de outro modo. Mais, se a equipa for heterogénea, por exemplo sendo composta por robots com diferentes sensores ou atuadores, é ainda possível optimizar a utilização desses componentes, reduzindo os custos e simultaneamente mantendo a funcionalidade do sistema. Mas o que é mais atrativo nestas equipas é a possibilidade de maximizar a utilidade de todo o sistema; por exemplo maximizando a eficiência de tarefas de vigilância utilizando os vários sensores disponíveis de forma cooperativa, aumentando a velocidade com que se consegue cobrir uma certa área em operações de busca e salvamento, ou coordenando os movimentos para permitir o transporte de grandes objetos.

Há vários fatores que são essenciais para permitir a cooperação; neste trabalho nós vamo-nos focar em dois deles: a)troca de informação; b)mapeamento das posições relativas.

Quanto ao fator a), iremos investigar uma nova solução para comunicar dados entre os elementos de uma equipa dinâmica de robots heterogéneos, que devido à mobilidade dos robots, e sua entrada e saída de atividade, terá de ser realizada recorrendo a comunicações sem fios. Logo, iremos desenvolver um protocolo de comunicações que suporte uma equipa com elementos variáveis, baseada em comunicações de proximidade *(ad hoc)* descentralizadas.

Contudo, para melhorar a utilização do meio de comunicação, as transmissões serão sincronizadas recorrendo a uma estrutura temporal circular do tipo *Time Division Multiple Access* (TDMA). Uma das inovações deste protocolo é a obtenção de sincronização distribuída sem recurso a um relógio global, tolerante a tráfego externo à equipa. Recorrendo a técnicas de consenso, mostramos que este método baseado nos instantes de receção de mensagens trocadas entre os vários elementos da equipa garante a sincronização da equipa numa vasta gama de situações operacionais. Refira-se ainda que o protocolo desenvolvido suporta comunicações *multi-hop*, utilizando um algoritmo novo que reduz o atraso na entrega das mensagens, mas é de salientar que reduz de forma drástica a variabilidade desse atraso, isolando os requisitos de comunicação de cada nodo ao longo de toda a rede, melhorando significativamente o seu comportamento temporal e facilitando a sua análise.

Relativamente ao fator b), desenvolvemos novos métodos para estimar as posições relativas dos robots a partir das comunicações RF locais. Em particular abordamos os paradigmas baseados em força de sinal e em medição de tempo de propagação, e com eles propomos uma inovadora abordagem hibrida capaz de melhorar a performance de cada um dos paradigmas individualmente. Estes paradigmas serão então utilizados de forma a produzir medidas de distâncias entre todos os robots que utilizaremos para estimar as suas posições, relativas uns aos outros. Neste aspecto, outra das inovações introduzidas é uma forma de determinar a confiança das posições relativas determinadas por este método em função da confiança das distâncias medidas entre robôs.

Acknowledgments

Deixo aqui os meus agradecimentos a todos os que contribuíram para o desenvolvimento deste trabalho.

Começo, como não poderia deixar de ser, pelo meu orientador, Luís Almeida. Este trabalho só foi possível devido ao seu apoio, motivação, e acompanhamento ao longo dos últimos anos.

Agradeço também ao meu co-orientador Pedro Lima pelas interações tidas principalmente no âmbito do projeto PCMMC.

Ao Traian Abrudan, pelo apoio fundamental no estudo das questões de localização.

Aos varios os alunos do MIEEC, na FEUP, da disciplina de Sistemas Distribuidos que de variadas formas colaboraram neste trabalho.

Ao Heber Sobreira, Filipe Santos, Luís Rocha, e restantes elementos da equipa 5DPO, com quem partilhei muitos momentos de descontração e de privação de sono.

Aos muitos companheiros de laboratório, e não só, que foram circulando ao longo dos últimos anos pelas discussões e trocas de ideias, discussões sem fim, e por todo o convivio para além do trabalho.
Ana Rita Pereira, André Moreira, Aqsa Aslam, Bernardo Ordoñez, Carlos Pereira, Carmelo di Franco, Daniel Ramos, David Gessner, Frederico Santos, Julio Cano, Luís Costa, Luís Pinto, Luís Silva, Manuel Barranco, Morris Beham, Pedro Silva, Ricardo Marau, Robson Costa, e Zahid Iqbal.

Ao meu pai, à minha mãe, à minha irmã, e ao meu irmão, sem a qual nada disto teria sido possível.

A todos os meus outros amigos, que, apesar de não os poder citar a todos, foram definitivamente uma parte fundamental no sucesso desta tese.
Ana Duarte, Diogo Carvalho, Hugo Duarte, António Gomes, Diogo Vidigal, Marina Domingas, Nuno Henriques, Nuno Marujo, Pedro Brochado, Rosalina Ribeiro.

E a todos os que de alguma forma estiveram presentes na minha vida neste periodo.

A todos muito obrigado!
Luís Oliveira

Contents

Agradecimentos **ix**

List of Figures **xvi**

List of Tables **xvii**

List of Abbreviations **xxi**

1 Introduction **1**
- 1.1 Communication Amongst Mobile Robots 1
- 1.2 RF-based Relative Localisation for Mobile Robots 3
- 1.3 Thesis 4
- 1.4 Contributions 5
- 1.5 Structure of the Dissertation 6

2 An Overview of Wireless Communication Protocols for Ad-Hoc Networks **9**
- 2.1 Communication Standards 9
  - 2.1.1 IEEE802.11 9
  - 2.1.2 IEEE802.15.4 12
- 2.2 Non-standard Medium Access Control 14
  - 2.2.1 Communication protocols in wireless sensor networks 14
  - 2.2.2 Communication protocols in robotic applications 17
  - 2.2.3 Discussion 20

3 An Overview of Wireless Localisation Techniques **23**
- 3.1 RF Ranging Techniques 23
  - 3.1.1 Time based techniques 23
  - 3.1.2 Received Signal Strength based technique 25
  - 3.1.3 Hybrid techniques 25
- 3.2 Deriving Robots Positions Using RF Signals 26
  - 3.2.1 Absolute positioning 26
  - 3.2.2 Relative positioning 26
  - 3.2.3 Discussion 27

4 A Communication Protocol for Robot Ad-Hoc Networks **29**
- 4.1 Relevant Publications 30
- 4.2 Reconfigurable and Adaptive Ad-Hoc Synchronisation Protocol 31
  - 4.2.1 Problem statement 31
  - 4.2.2 Propagation of information 32

4.2.3 Slot assignment 33
4.2.4 Synchronisation in ad-hoc networks 35
4.3 Convergence of the Synchronisation Algorithm 36
4.3.1 Revisiting the problem 36
4.3.2 Bounding the maximum delay induced by synchronisation 40
4.3.3 Proving the convergence 41
4.3.4 Dropping the $\mathbf{A}_{\mathbf{\Phi}} < \pi$ assumption 43
4.4 Upper Bounds to Information Propagation Latency 45
4.5 Experimental Validation of the Synchronisation Protocol 46
4.5.1 Validating the synchronisation protocol 46
4.5.2 Impact of the synchronisation on the application 47
4.6 Conclusions 50

5 Multi-hop Communications Within a TDMA Schedule 51
5.1 Relevant Publications 52
5.2 TDMA in Mobile Robot Teams 52
5.2.1 Team end-to-end communication requirements 53
5.2.2 Packet duration in IEEE 802.11 DCF 53
5.2.3 One-hop per slot forwarding 55
5.2.4 Multi-hop per slot propagation 59
5.2.5 Multi-hop per slot with full slot isolation 61
5.3 Experimental Validation 65
5.3.1 Comparison with the one-hop per slot forwarding approach 65
5.3.2 Experimental Comparison with Token Passing 67
5.4 Conclusions 71

6 RF-based Ranging for Teams of Mobile Robots 73
6.1 Relevant Publications 74
6.2 Round-Trip Time-of-Flight Ranging 74
6.3 Received Signal Strength Indicator 75
6.4 RF-based Ranging using RT-TOF and RSSI 75
6.4.1 Filtering the RSSI readings 76
6.4.2 Online channel estimation 76
6.4.3 Extended Kalman filter for range tracking 77
6.5 Experimental Results 77
6.5.1 Setup 77
6.5.2 Results 79
6.6 Conclusion 84

7 Estimating Positions from Pairwise Distances 85
7.1 Relevant Publications 86
7.2 Estimating Relative Positions in Signal Space 86
7.2.1 The Signal Space Distance matrix 86
7.2.2 Filtering the proximity measurements 87
7.2.3 Generating signal space relative positions 88
7.3 Experimental Results on Signal Space Positioning 92
7.3.1 Sensitivity to parameter selection 93
7.3.2 Conclusions 96
7.4 Estimating Relative Positions with Real Distances 96

7.4.1 Collecting distance information from robots 96
7.4.2 Estimating positions from distances 97
7.4.3 Filtering positions with a Kalman filter 99
7.5 Simulation Results on Real Distances Positioning 101
7.5.1 Generating RSSI and RT-TOF measurements 101
7.5.2 Simulating the communications 101
7.5.3 Results 102
7.5.4 Conclusions 103

8 Conclusions 105
8.1 Thesis Validation 105
8.2 Summary of Contributions 106
8.3 Future Work 106
8.3.1 Proving synchronising of packet exchanges 106
8.3.2 Improving the analysis for immediate multi-hop forwarding 107
8.3.3 The impact of dynamic topologies on multi-hop communications 107
8.3.4 Explore the possible benefits of the non-isolated multi-hop per slot approach . . 107
8.3.5 Integration of the relative position estimator 108

A Appendix A - Middleware 109
A.1 The RTDB 110
A.2 Extending the RTDB to Ad-Hoc Scenarios 110

B Appendix B - Navigation in RSS fields 113

Bibliography 115

List of Figures

1.1 Hidden node . . . 3
1.2 A scenario of mine sweeping . . . 4

2.1 IEEE802.11 2.4GHz channel overlap . . . 10
2.2 IEEE802.11 BSS vs IBSS . . . 11
2.3 Simplified version of CSMA/CA . . . 12
2.4 IEEE802.15.4 Star/Tree vs Mesh . . . 13

4.1 Example of a snapshot with hidden robots overlapping their transmissions . . . 30
4.2 The structure of the TDMA round . . . 32
4.3 Broadcast protocol - sending and receiving procedures . . . 33
4.4 Control variables to update the connectivity matrix . . . 34
4.5 Updating the slot allocation table . . . 34
4.6 Propagating slots synchronisation . . . 35
4.7 Complete broadcast protocol - sending and receiving procedures . . . 36
4.8 Time represented as a phase . . . 37
4.9 A TDMA cycle represented as a phase . . . 38
4.10 The Arc of a set of round phases . . . 39
4.11 Classification of the robots within the Arc according to their phases . . . 40
4.12 Dependency on the point of observation . . . 42
4.13 First example of a failure to synchronise . . . 43
4.14 Second example of a failure to synchronise . . . 44
4.15 Building a tree from the topology information . . . 44
4.16 Worst-case data propagation scenario . . . 45
4.17 Illustration of the ToF ranging process . . . 46
4.18 Synchronising disjoint networks with one robot . . . 47
4.19 Synchronisation protocol operating with joining and leaving robots . . . 48
4.20 Two fully linked physical layouts . . . 48
4.21 Periodic dissemination with and without synchronisation . . . 50

5.1 One-hop per slot forwarding . . . 57
5.2 Propagating a packet “counter-slot-wise” with $N = 5$ and $n = 0..4$ . . . 58
5.3 Slot structure for five agents . . . 58
5.4 Multi-hop per slot forwarding . . . 60
5.5 Multi-hop per slot with full slots isolation . . . 62
5.6 Illustration of worst case blocking . . . 63
5.7 Simulation results . . . 69
5.8 Streaming audio streams across a multi-hop network . . . 71

6.1 Illustration of the Round-Trip Time Of Flight (RT-ToF) ranging process 74
6.2 RF-based ranging: Dotted lines apply only when RT-ToF data is available 76
6.3 Sliding window median filter ($k = 5$) 76
6.4 Communication period as seen by node 2 78
6.5 Soccer field where experiments were made 79
6.6 Error of the communication channel model 80
6.7 Error of distance between robot 1 and robot 2 82
6.8 Error of distance between robot 3 and robot 2 83

7.1 Signal Space Distance matrix 87
7.2 The filtering process for D 88
7.3 Generating Positions from the Signal Space Distance 89
7.4 Relative positions in physical and signal strength space 89
7.5 Joining weakly-connected networks using iterative methods 90
7.6 Approximating the distance for missing SSD measurement 90
7.7 Adjusting Coordinates 92
7.8 Node distribution 93
7.9 Experiments concerning MDS 95
7.10 RF-based ranging for each link 97
7.11 Simulation loop for robot i 102
7.12 Error of the position estimates 103
7.13 A zoomed in snapshot of the simulation field 104
7.14 Percentage of GT positions contained inside of the covariance ellipses 104

A.1 Original RTDB 111

List of Tables

2.1 Qualitative comparison between communication protocols 22

3.1 Qualitative comparison between ranging methods 28

4.1 Experimental results 49

5.1 IEEE802.11b/g parameters 54
5.2 OMNeT++ simulation parameters 65
5.3 Statistics of the simulation results 68

6.1 Estimation error in all three experiments 81

List of Abbreviations

AoA Angle of Arrival.

AODV Ad hoc On-Demand Distance Vector.

AP Access Point.

API Application Programming Interface.

B.A.T.M.A.N. Better Approach To Mobile Ad-hoc Networking.

BSS Basic Service Set.

CAMBADA Cooperative Autonomous Mobile roBots with Advanced Distributed Architecture.

COTS Commercial Off-The-Shelf.

CPS Cyber-Physical Systems.

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance.

CSS Chirp Spread Spectrum.

CTS Clear To Send.

DCF Distributed Coordination Function.

DI-EDF Dynamic Implicit-EDF.

DIFS Distributed Coordination Function Inter Frame Spacing.

DSDV Destination-Sequenced Distance-Vector.

DSSS Direct-Sequence Spread Spectrum.

EDCA Enhanced Distributed Channel Access.

EDF Earliest Deadline First.

F-RTS Future-Request To Send.

FFD Full-Function Device.

GPS Global Positioning System.

HCCA Hybrid Coordination Function Controlled Channel Access.

IBSS Independent Basic Service Set.
IEC International Electrotechnical Commission.
IIT Inserted Idle Time.
IMU Inertial Measurement Unit.
ISA International Society of Automation.
ISM Industrial, Scientific and Medical.

LR-WPAN Low-Rate Wireless Personal Area Network.

MAC Media Access Control.
MANET Mobile Ad-hoc Network.
MDS MultiDimensional Scaling.
MSL Middle-Size League.

OFDM Orthogonal Frequency-Division Multiplexing.

PAN Personal Area Network.
PCF Point Coordination Function.

QoS Quality of Service.

RA-TDMA Reconfigurable and Adaptive TDMA.
RF Radio Frequency.
RFD Reduced-Function Device.
RSSI Received Signal Strength Indicator.
RT-ToF Round-Trip Time Of Flight.
RT-WMP Real-Time – Wireless Multihop Protocol.
RTDB Real-Time Database.
RTS Request To Send.

SLAM Simultaneous Localisation And Mapping.
SM-IWMS Seamless Mobility within Industrial Wireless Mesh Networks.
SSD Signal Space Distance.

TDMA Time Division Multiple Access.

TDoA Time Difference of Arrival.

ToA Time of Arrival.

U-NII Unlicensed National Information Infrastructure.

UWB Ultra-Wide Band.

WICKPro WIreless Chain networK Protocol.

WLAN Wireless Local Area Network.

WMN Wireless Mesh Network.

WSN Wireless Sensor Network.

Chapter 1

Introduction

Human controlled machinery is still widely used in our societies for a large number of processes. Despite that, in a growing number of scenarios human intervention is undesirable, impossible, or insufficient, thus autonomous robotic units must be used. In recent years, several applications as diverse as search and rescue operations, surveillance, cleaning, and transportation of large volumes have favoured autonomous robots to perform several of the needed tasks. In some cases robots can assist in tasks that would be impossible to be performed by a person alone, in other cases robots can replace people in dangerous tasks, thus protecting people from being exposed to dangerous situations, and in other cases robots can replace people in a repetitive or boring task that they simply do not want to perform. However, one single robot may not be enough. In those situations, the solution researchers are focusing their work involves several robots cooperating as a team. This allows to maximise the utility of the whole system by, for example, increasing the effectiveness of surveillance by performing cooperative sensing [1, 2], using relative positions to improve the coverage in search and rescue [3, 4], or performing motion coordination for the transport of large parts [5]. Moreover, if the team is heterogeneous, i.e. robots have different sensors/actuators, it is possible to reduce costs by sharing such components while maintaining performance or, in some cases, increasing performance, e.g. some robots with short range cameras and others with long range cameras, or one robot with an expensive actuator together with multiple sensor robots.

There are several key factors that enable such cooperation, in this work we will focus on two of them:

- A wireless communication protocol for exchanging data
- An RF-based relative positions manager

1.1 Communication Amongst Mobile Robots

Already in the 1990s it was well known that sharing of information is one of the key features to enable co-operation in autonomous and mobile multi-robot teams, which due to the inherent mobility requirements, is usually done through a wireless medium. For example in [6], it is shown that even the communication of small amounts of data can greatly benefit the performance of a team containing two robots. Despite that, as it is pointed out in [7], this topic is still far from settled.

To begin with, the wireless medium is known to raise several issues that can affect timeliness and reliability [8]. For example, during transmission, signals are reflected, diffracted, and scattered, which leads to a significant increase of bit-error rates and packet losses; the wireless medium is an open communication environment, consequently interference from external traffic is unavoidable, not only making the bandwidth limited but also variable; and the absence of a collision detection mechanism requires acknowledgement packets to ensure the proper delivery of data, thus reducing the reliability of broadcast packets that do not receive an acknowledge.

When the constraints on the movement of the robots are reduced, new challenges must be tackled. For example, communications through an Access Point (AP) are not desirable because it would hinder formations, such as line-formations, that due to the distance between the robots may not allow one-hop (direct) communications between all the team members; the necessity of having a dynamic team with members joining and leaving the network to perform small missions would also, probably, require using several AP, either moving with the team or in a previously built infrastructure, which would not only increase the total cost, but also be impractical or impossible to build beforehand. Therefore, in the general case, distributed ad-hoc wireless communications protocols are better suited for mobile robot teams, allowing them to communicate data amongst them without a centralised device. In spite of being a desirable option to multi-robot teams, ad-hoc networks raise even more issues. One of the very well known problems is the hidden node; consider three nodes A, B, and C as depicted in 1.1. In this situation B can hear both A and C, but C and A cannot hear each other. If A is sending to B, node C cannot detect the transmission, thus, if C then begins transmitting, the transmission from C will collide with the transmission from A corrupting one or both transmissions as they arrive in B. The hidden node problem is specially problematic due to the lack of synchronisation when the nodes compete for the opportunity of transmitting. While in networks with AP all the nodes hear the end of a transmission, thus synchronise for backoff purposes, in ad-hoc networks the channel perceived by one node is different from that perceived by another node rendering such mechanisms ineffective. One possible solution is to implement a time-triggered scheme to avoid internal team collisions, e.g. Time Division Multiple Access (TDMA) schemes. On one hand, these schemes help to reduce the problems caused by members of the team simultaneously trying to access the medium, and are appropriate to periodic state information sharing. On the other hand, the delay introduced by such schemes may be too high for event-triggered information that must be distributed quickly, thus requiring a separate management mechanism.

Despite the difficulties inherent to the wireless medium, there have been tremendous improvements – specially in relevant areas such as mobile ad-hoc networks (MANETs) [9] and Cyber-Physical Systems (CPS) [10]; using available Commercial Off-The-Shelf (COTS) technologies that range from Bluetooth and ZigBee to WiFi, WAVE, and WiMax or LTE, it is possible to provide cheap short (100m) to mid-range (>10km) communications, in some cases with Quality of Service (QoS), and different bandwidth options. In order to provide these options some approaches attempt to avoid internal team collisions with a self-synchronisation algorithm [11], others act on the inter-frame space [12] to prioritise high priority traffic, and others use polling master-slave solutions [13] to request transmissions. Nevertheless, in some cases, there is an inherent impossibility to provide QoS guarantees, in such cases, we are constrained to best-effort solutions [14].

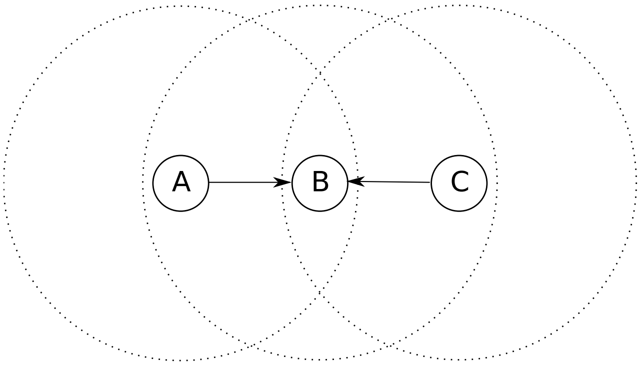


Figure 1.1: Hidden node – A is transmitting to B when C begins its transmission to B, corrupting one or both transmissions.

1.2 RF-based Relative Localisation for Mobile Robots

Beyond data exchanges, another service commonly needed to support cooperation amongst robots, and that can also be supported through wireless communications, is localisation. From cooperative tracking to coordination of movements, knowing the position of a robot can dramatically improve the performance of many algorithms. In some situations, it is possible to find the positions of robots using an infrastructure that allows every robot to extract its own absolute position. However, building an infrastructure is costly and it is probably unavailable in urgent scenarios. For outdoors, Global Positioning System (GPS) may be a possible solution; however, it is satellite dependent, thus it is not available everywhere, such as in indoor spaces and street canyons.

Alternatively, it is possible to derive relative positions from relative distances obtained through local communication, which may not carry extra costs if the communication hardware is already present. There are many methods to measure distances from local communications, however two assume particular significance. The first one is Received Signal Strength Indicator (RSSI) based ranging that not only is ubiquitous, but is also able to produce fast measurements, i.e. if several units receive a message from another one, all of them can obtain the RSSI value from that message. The disadvantage is that the RSSI is a measurement of signal strength, thus dependent on the propagation medium, antennas, and obstacles. The second method is RT-ToF measurements, where one unit measures the time a message needs to reach the destination and return, thus obtaining the distance that separates the two units as long as the speed of propagation is known. The problems of using RT-ToF ranging with mobile units are that it is only possible to range one robot per ranging operation, thus making this method less responsive to fast robot dynamics, and the necessity of using specialised hardware. Despite that, the ranging operation produces a distance in meters that is accurate enough to be used for localisation.

Consider a small team of heterogeneous robots working in a relatively small area. Each of the robots is equipped with identical basic platform and communication module so that they can communicate in a predefined channel and move in a coordinated pattern. For the sake of cost, robots may have different sensing or actuating components, which means that some tasks have to be accomplished by specific robots. When one special event is detected by one robot, it may have to notify another robot which is far from the event area, but equipped with specific actuating component, to deal with such event. For example, in a mine sweeping application, it is advisable to spread a team of robots with mine detecting capability and equip only a small portion of them with sweeping ability, thus reducing the cost of equipment. A typical scenario is depicted in Figure 1.2. The mine field is divided into smaller areas according to the robots sensing range and the robots sweep the areas one by one with certain formation to guarantee coverage. When mines are detected, a robot with sweeping ability is informed to approach the specific spot. For both maintaining the formation and relocating the sweeping robot, relative positions have to be managed. In both cases, it is important to know the relative positions in order to make the decisions involving moving robots from one place to another.

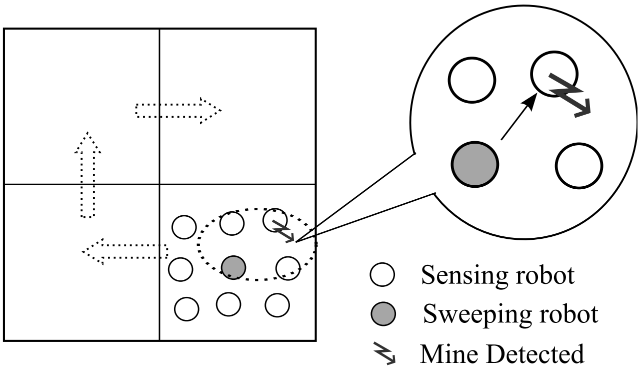


Figure 1.2: A scenario of mine sweeping: Only the sweeping robot is equipped with the hardware required to disable a mine, thus the necessity of navigating to the position where the mine was detected.

1.3 Thesis

In this thesis we make three claims. Firstly, we claim that ad-hoc wireless communication protocols, even those already relying on well known standards with robust random medium access arbitration mechanisms, can benefit from organising the transmissions in time with a TDMA framework. Specifically, by accounting for variability in team composition, we can change transmission schedules in order to minimise the number of internal team contention and collisions, and we can maximise the available bandwidth for each of the active team members. We validate this claim in ad-hoc scenarios with a thoroughly modified version of a previously developed protocol, the Reconfigurable and Adaptive TDMA (RA-TDMA) protocol, in order to be able to use it in ad-hoc networks. This protocol allows to synchronise transmissions in an efficient way, without requiring explicit transmission control mechanisms or global clock synchronisation, and only sharing a small amount of membership information.

Secondly, we claim that adequate multi-hop message forwarding mechanisms can significantly reduce the message delivery delays imposed by typical TDMA schedules. We validate this claim within IEEE802.11 ad-hoc networks using a small amount of global information, namely the TDMA round structure and topology information, by extending the Ad-hoc RA-TDMA protocol. The RA-TDMA protocol underutilises a significant portion of the bandwidth available, therefore we use that portion to transmit data in a multi-hop fashion, allowing robots to reach beyond their local neighbourhood in their own slot. We compare our approach with a traditional TDMA transmission schedule and show that we can reduce message propagation delay, and especially its variability, with minimum overhead.

Finally, our third claim is that using the wireless transceiver, only, it is possible to perform relative localisation of the team members even in the presence of propagation medium changes. To support this claim we developed two different localisation algorithms, one based solely on RSSI and another based on a hybrid approach using RSSI and RT-ToF. The first approach is a lightweight, low accuracy approach that localises robots without modelling the propagation medium, thus it is only enough to keep connectivity or for basic robot navigation. The latter approach, is a higher precision approach that, by increasing communication and computing requirements, is able to estimate the propagation channel parameters, and calculate relative positions with confidence areas, that we believe to be helpful for more complex sensor fusion and navigation techniques.

1.4 Contributions

The main contributions of this work, as stated in the section above, are an ad-hoc wireless communication protocol, namely the Ad-hoc RA-TDMA, and the relative localisation service.

The Ad-hoc RA-TDMA wireless communication protocol, is based on the RA-TDMA approach proposed in [11] for infrastructured networks, and extends this approach to ad-hoc networks where the team members can form an ad-hoc network in any configuration. Unlike other approaches, that rely on clock synchronisation services, or on external communication triggering mechanisms (tokens), our proposal uses the neighbour robots transmissions and a small amount of global information, namely the number of active members and their id numbers, to create and maintain a synchronised schedule of transmissions. More specifically, when robots listen to the medium and receive a message from a neighbour robot, they automatically update their neighbourhood information and (re)synchronise their transmissions.

The Ad-hoc RA-TDMA protocol is only efficient, and planned, for transmitting state data, however, in most ad-hoc robotic applications, the ability of sending data to one non neighbour robot in a multi-hop fashion is vital. For that reason we developed an extension to the protocol, where the empty portion of each robot slot can be used for multi-hop transmissions. The main contribution of this part of the work is the message transmission and forward approach, namely the in-slot forwarding mechanism that allows for intermediate robots to immediately forward packets not addressed to them, within the message source slot. This has a significant positive impact on the delivery delay of multi-hop messages, since the number of TDMA cycles they have to wait in robot egress queues is reduced, when compared to traditional TDMA approaches.

This protocol was presented in the following publications:

- [15] L. Oliveira, L. Almeida, F. Santos, A loose synchronisation protocol for managing rf ranging in mobile ad-hoc networks, in: T. Rfer, N. Mayer, J. Savage, U. Saranl (Eds.), RoboCup 2011: Robot Soccer World Cup XV, Vol. 7416 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 574—585.

- [16] L. Oliveira, L. Almeida, P. Lima, Multi-hop routing within TDMA slots for teams of cooperating robots, in: IEEE World Conference. on Factory Communication Systems - WFCS, 2015.

The second main contribution is the relative localisation service which using information collected solely by the wireless transceiver, is capable of estimating the relative positions of the team robots. Our work covers two different approaches to RF-based ranging, a very coarse RSSI-based ranging approach, and an hybrid approach using both RSSI and RT-ToF to estimate the path loss model on-line, and obtain a higher ranging precision.

Then, we filter this ranging information, and we propose a technique based on Multidimensional Scaling to obtain relative position information. Using the first ranging approach we can only obtain a very coarse representation of the robots positions, nevertheless, we claim that this coarse representation is enough to perform basic coordination of the nodes such as keeping connectivity within the network or perform basic navigation (Annex B). Using the second more robust approach we are able to provide significantly more useful information, namely, unlike the first approach where we work with RSSI in dB, we measure distance in meters, thus providing estimates with a direct physical mapping. Moreover, in addition to the positions estimate, we provide an estimate of the covariance of the positions, allowing the user to define confidence regions around the estimate, which, to the best of our knowledge, is one of the first works computing confidence regions associated to MDS-computed position estimates. Finally, our proposed scheme, unlike previous work, does not use sensors other than the RF transceiver module to compute relative positions, does not make assumptions on the dynamics of the robots, and does not assume pre-installed anchor nodes or robots with known positions.

This protocol was presented in the following publications:

- [17] Oliveira, L.; Di Franco, C.; Abrudan, T.E.; Almeida, L., "Fusing Time-of-Flight and Received Signal Strength for adaptive radio-frequency ranging," Advanced Robotics (ICAR), 2013 16th International Conference on , vol., no., pp.1,6, 25-29 Nov. 2013

- [18] L. Oliveira, H. Li, L. Almeida, T. E. Abrudan, Rssi-based relative localisation for mobile robots, Ad Hoc Networks 13 (2014) 321–335.

- [19] L. Oliveira, L. Almeida, Rf-based relative position estimation in mobile ad-hoc networks with confidence regions, in: RoboCup 2014: Robot World Cup XVIII, Springer, 2015, pp. 383–394.

1.5 Structure of the Dissertation

With this first chapter, we presented the scope of this work and what we propose to achieve. The remaining structure of this dissertation is as follows.

Chapter 2 presents an overview of wireless standards and non-standardised protocols present in the literature. The focus of this thesis is on mobile robots, therefore in this chapter the focus will be on analysing different ad-hoc wireless communication technologies, and assessing their applicability to scenarios where mobile robots may be used.

Chapter 3 presents recent work developed in the scope of localisation of mobile robots, with special focus on RF-based techniques. Special attention will be given to mobile robot localisation, despite that, related areas, such as wireless sensor networks, will also be considered.

Chapter 4 presents the Ad-hoc RA-TDMA wireless communication protocol, which is responsible for managing the synchronisation of team members. Special focus will be given to the team management and slot allocation algorithm, and to the proof of convergence of the synchronisation. Finally we present some results with the experimental validation of our algorithm.

Chapter 5 presents the multi-hop unicast extension to the Ad-hoc RA-TDMA wireless communication protocol, which manages the communications between non-neighbour robots. Different approaches are studied in terms of imposed end-to-end delay, and simulation results are presented.

Chapter 6 introduces the problem of ranging robots using the wireless transceiver, both using Received Signal Strength and Round-Trip Time of Flight. An hybrid solution is proposed where both the measurements are used to improve ranging accuracy, while being able to track the mobile robots highly dynamic movements. Experimental results are presented.

Chapter 7 follows up with a technique to obtain relative positions estimates from pairwise ranging information. Here the focus is on merging the ranging information from the different robots, and filter the result in order to provide a 2D map of the relative positions. This chapter covers both coarse low accuracy RSSI-only based positioning, and high accuracy hybrid RSSI/ToF relative positioning. Finally, some simulation results are presented to validate our approach.

Chapter 8 presents the conclusions of this work, and puts forwards several research lines that remain open for future work.

Finally, we still include two annexes. Annex A briefly addresses the instantiation of the RTDB middleware on top of the Ad-hoc RA-TDMA protocol, to facilitate the access to remote data from the application side. In turn, Annex B summarises previous works from the group related to navigation based on RF signals.

Chapter 2

An Overview of Wireless Communication Protocols for Ad-Hoc Networks

In this chapter we present and discuss previous work in the topics related to wireless communication protocols within mobile robot teams. In the first section we present some of the technologies widely used for ad-hoc networking, namely the IEEE802.11 and the IEEE802.15.4 communication standards. Then we will explore other non-standardised protocols, some of them trying to improve the performance of these standards, and some proposing complete new protocols.

2.1 Communication Standards

Standard protocols are usually designed for supporting many different conditions in broad application domains. Therefore, using them as a basis and develop improvements on top of them is a common practice. When restricting our choice to standard Radio Frequency (RF) communication solutions that are widely available in the market, i.e., COTS, and also widely used within the Robotics community, two solutions stand out. Small teams of agents that are more computing capable and typically have IEEE 802.11 RF interfaces. In the case of swarms, the agents are typically simpler and use technologies that are less energy consuming and can be embedded in smaller nodes. Probably the most used RF technology in this case, today, is IEEE802.15.4.

In this work, we developed our protocols on top of these two widely used standards. Thus, we present here a brief discussion of their main features.

2.1.1 IEEE802.11

Popularized as WiFi by the Wi-Fi Alliance, a trade association that promotes Wireless Local Area Network (WLAN) technology and certifies products, the IEEE 802.11 standard [20], defines the physical and Media Access Control (MAC) layers used in over-the-air communications. The first version of IEEE 802.11 was released in 1997. Since then, multiple amendments were introduced, improving the base standard in speed, security, and in QoS. A new version of the standard was released in September 2007

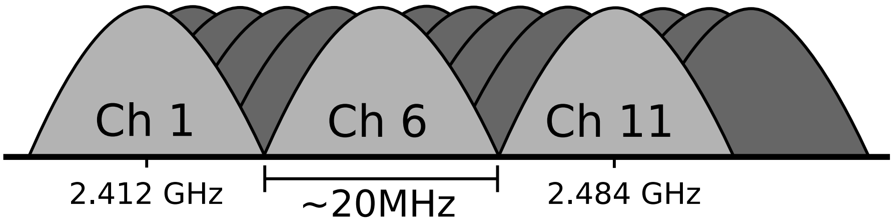


Figure 2.1: IEEE802.11 2.4GHz channel overlap – Only a maximum number of three channels (1, 6, and 11) can be used simultaneously without overlap.

(including the amendments a, b, d, e, g, h, i and j), and later, in 2012, the current version [20] was released (including 10 new amendments).

Within IEEE 802.11 three frequency bands are allowed, where the most used are the Industrial, Scientific and Medical (ISM) band at 2.4GHz, and the Unlicensed National Information Infrastructure (U-NII) band at 5GHz. The third frequency band (used by the amendment y) is only available in the United States of America (USA) and operates in the frequency range 3.65GHz – 3.70GHz. Both the IEEE 802.11b and IEEE 802.11g use the ISM band where many house-hold appliances also operate such as microwave ovens, wireless telephones, and Bluetooth devices, which may occasionally cause interference. In order to tolerate interference IEEE 802.11b/g uses Direct-Sequence Spread Spectrum (DSSS) and Orthogonal Frequency-Division Multiplexing (OFDM) modulation, respectively. IEEE 802.11a uses the U-NII band and OFDM modulation. Both 2.4GHz and 5GHz bands are divided in channels. The former contains 14 overlapping channels with 5MHz separation, which limits the number of non-overlapping channels to 3, see Figure 2.1. Conversely the latter provides 19 and 20 independent channels (in Europe and USA respectively) with 20MHz separation.

In IEEE 802.11 there are two types of devices, station and AP. The former (stations) are the endpoints, the producers and consumers of data, and the latter (AP) are central devices whose objective is to mediate the message exchanges between stations. Concerning the topology (Figure 2.2), most IEEE 802.11 networks form a star topology in a mode called Infrastructure Basic Service Set (BSS) where the repeater is the AP. However, the standard also defines another type of network called Independent Basic Service Set (IBSS), also known as ad-hoc mode, that conversely to the Infrastructure BSS, all the stations form a mesh network and interact directly with each other. This makes this type of network more flexible, consequently, it is more useful for specific purpose, short duration applications, for example, the execution of a mission by a team of robots.

In IEEE 802.11, all the stations in the network share the same channel. Consequently, the medium is half-duplex and stations must contend for the medium. This makes an arbitration mechanism vital to avoid losing transmissions by collision (simultaneous transmission). Without the capacity to listen to what is being transmitted, the sending node cannot detect a collision occurrence. Therefore it must rely on mechanisms that try to avoid this, specially with the recent increase in wireless devices, where everything from computers to wrist watches carries a WiFi device. And as a consequence, there is a high potential for excessive load on WiFi networks leading to an increase of collisions and substantial

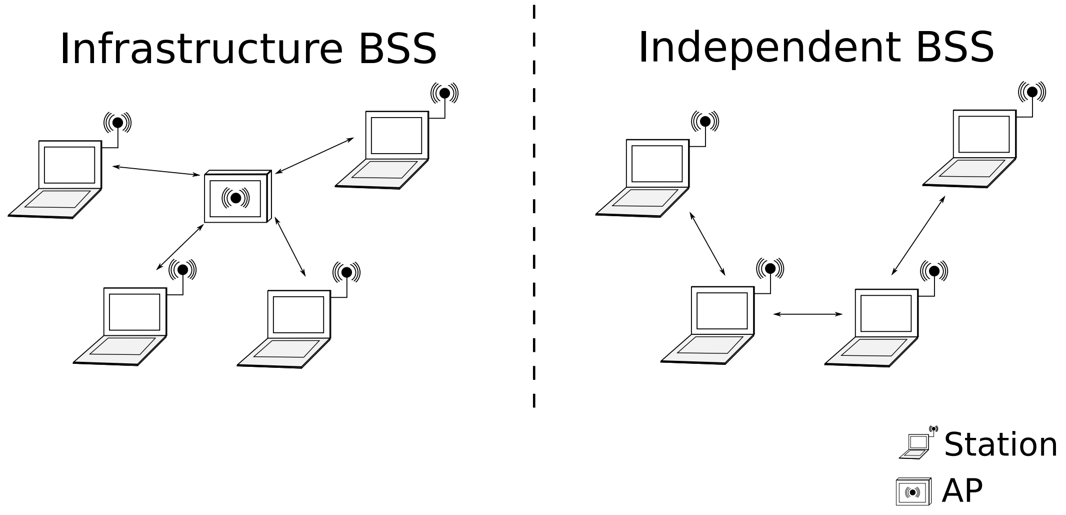


Figure 2.2: IEEE802.11 BSS vs IBSS – (left) The Infrastructure BSS forms a star topology (the AP is the central device). (right) The Independent BSS forms a mesh topology (stations communicate directly)

degradation of network performance [21]. This is particularly undesired in robotic systems due to a potential negative impact on system performance.

In order to mitigate this phenomenon, there are multiple techniques, some of which are integrated into the IEEE 802.11 protocol itself, such as the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) method, the scheduled traffic with Point Coordination Function (PCF) or Hybrid Coordination Function Controlled Channel Access (HCCA), which unfortunately are hardly supported by COTS WiFi equipment, and using QoS extensions with Enhanced Distributed Channel Access (EDCA) such as traffic differentiation in QoS classes [20, 22]. This latter case is useful when there are different kinds of traffic, some of which is more important than other. Unfortunately, this is not the case we are considering, since we need to handle multiple devices communicating with similar level of importance but it may help, for example, in separating high rate feedback control traffic from low rate multimedia streams or even file transfers for logging and configuration. IEEE 802.11 also defines a negotiation protocol using a request Request To Send (RTS) and a transmission clearance Clear To Send (CTS), optionally introduced to help solving the hidden node problem. However, in ad-hoc networks this mechanism in known to have limited usefulness [23].

The most general and widely supported method to arbitrate the medium is CSMA/CA. Using CSMA/CA, before sending any data, stations listen to the channel for a predefined amount of time. If during this time the medium remains constantly free of transmissions, then the transmission can begin. Conversely, if the medium is found busy at any point during this period, a waiting process based on random waiting intervals begins. A simplified diagram of this process is depicted in Figure 2.3.

CSMA/CA is known to be particularly effective with a lightly loaded medium, where it is known to solve the problem of collisions and improve quality of communications. As shown in [11], when the medium load increases, it creates variable and unpredicted delays. Nonetheless, even when more

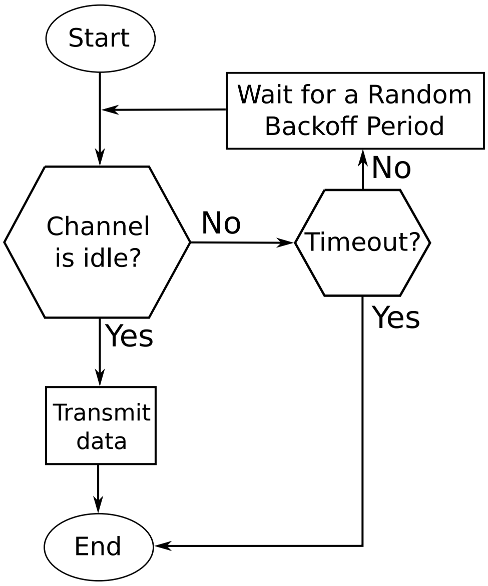


Figure 2.3: Simplified version of CSMA/CA – Before sending any data, stations listen to the channel for a predefined amount of time. If the medium remains free the transmission begins, otherwise random waiting intervals begins and the process is restarted.

specialised and robust implementations are used for medium arbitration, CSMA/CA is still quite useful is solving collisions created by unpredictable or uncontrollable interferences.

Since it became the de facto standard for consumer electronics, every computer, particularly laptops, comes equipped with an IEEE 802.11 device. That, allied with laptop computers being the core computing platforms in many mobile autonomous robots, made this standard the main communication protocol used by the Robotics community. Consequently, we will also use this protocol later on, in the validation of some parts of this thesis.

2.1.2 IEEE802.15.4

IEEE 802.15.4 [24] is a wireless communications standard targeting applications that require low data rate and long battery life at low cost, typically referred as Low-Rate Wireless Personal Area Network (LR-WPAN). IEEE 802.15.4 chip vendors typically sell microcontroller-based integrated radios that allow very small and compact systems.

The standard defines the physical and MAC layers. Some of its most distinguishing features include selectable data rates of 851kbps, 250kbps, 100kbps, 40kbps, and 20kbps, star/tree or mesh topologies, short (16bit) or extended (64bit) addresses, non-persistent CSMA/CA arbitration, automatic acknowledgement of unicast transfers and link quality indication. Moreover, the protocol supports synchronisation based on beacons that allows contention-free communication using Guaranteed Time Slots as well as low duty-cycle communications for very low power operation. In the frequency domain the standard offers different profiles with different channels and modulation options. The most common options are 16 channels in the 2.4GHz band, 10 channels in the 915MHz band and 1 channel in the 868MHz band, all with DSSS modulation. Optionally, there is a profile with 14 overlapping Chirp Spread Spectrum (CSS)

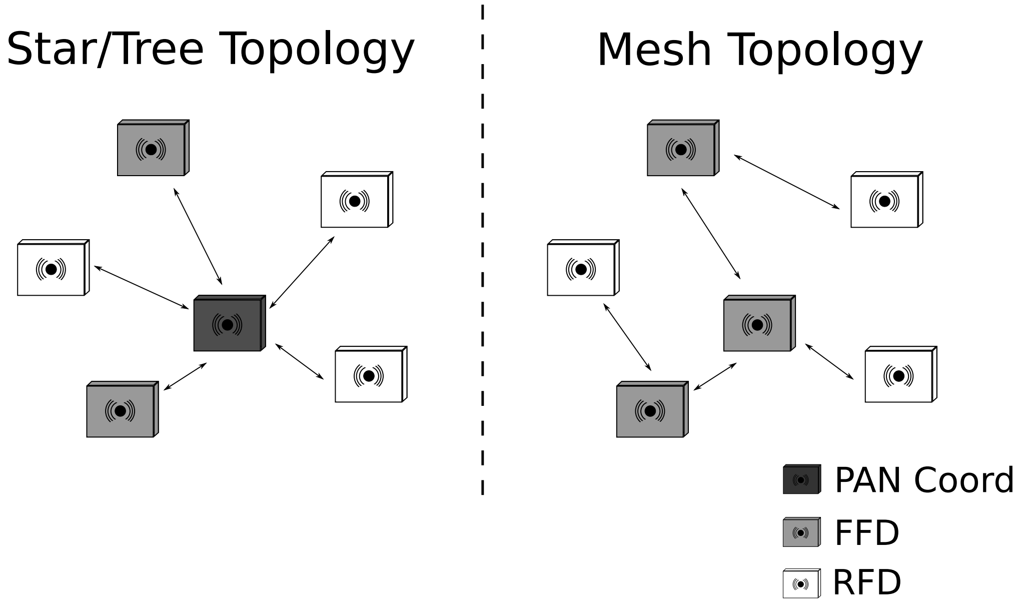


Figure 2.4: IEEE802.15.4 Star/Tree vs Mesh – (left) The star/tree topology forms clusters and connect to the PAN coordinator. (right) The mesh topology is peer-to-peer with non-restricted topologies.

channels in the 2.4GHz band, another with 16 channels in Ultra-Wide Band (UWB) band (500MHz and 3GHz to 10GHz), and a few other with special frequencies for use in China.

Concerning the topology, the standard defines two types of nodes, namely Full-Function Device (FFD) and Reduced-Function Device (RFD). The former FFD can either communicate their own data, in this case they are ordinary network nodes, or they can relay messages of other nodes, in which case they take up the role of coordinators. In the whole Personal Area Network (PAN) one coordinating FFD is selected to the PAN coordinator, concentrating network management functions. On the other hand, the latter (RFD) are ordinary simple nodes with very modest resource and communication requirements and which can only communicate with FFD. RFD can never act as coordinators.

Using these two types of nodes, an IEEE 802.15.4 PAN can be built as a mesh, a.k.a. peer-to-peer with non-restricted topologies, or as a star/tree with clusters (Figure 2.4). In this latter case, the cluster heads are coordinating FFD and the root of the tree (or star hub) is typically the PAN coordinator. In any case, one coordinating FFD is always needed to serve as PAN coordinator.

One important aspect is that the standard does not define a network layer, thus routing and multi-hop communications require additional layers not specified in the standard. ZigBee, WirelessHART, ISA100 and MiWi are examples of high level layers, built on top of the IEEE 802.15.4, to offer services tailored to specific application domains.

Among these, ZigBee is probably the most common high level specification, enhancing the standard by adding network and security layers and an application framework. In particular, ZigBee supports a large number of interoperable specifications for different application domains, from Health Care to Home Automation, Smart Energy, Telecom Services and Building Automation and Retail Services.

WirelessHART aims at industrial process monitoring, control and assets management. It was launched by the HART Foundation to complement and enhance the old HART Protocol (cabled). ISA100 has very similar aims as WirelessHART but was developed by the International Society of Automation (ISA), as a publicly available specification approved by the International Electrotechnical Commission (IEC).

Finally, MiWi is a proprietary protocol designed by Microchip Technology for use in PIC and dsPIC micro-controllers. Its distinguishing feature is its small protocol stack, approximately 90% smaller than ZigBee's, which allows its use on very memory-constrained devices. The MiWi specification and protocol stack are freeware but only run on Microchip micro-controllers.

The IEEE 802.15.4 standard and the higher layers referred above were primarily designed to support Wireless Sensor Network (WSN). However, they can also be used to equip autonomous mobile robots, particularly small robots that are energy constrained and move in relatively small areas. In this thesis, some of the experiments were in fact developed over this standard. Other previous examples can be found in the references in Annex B.

2.2 Non-standard Medium Access Control

As it is shown in [6], communication of data amongst a team of robots may have a positive effect on the team global performance. In such cases, the performance enhancement is always limited by the quality of the communication channel. This quality can be hindered by many factors when using a wireless medium. For example, when comparing with cabled communication, the reliability of packet delivery is orders of magnitude lower, with frequent packet losses. Moreover, the medium is shared and open raising security and safety issues. Particularly, the medium can be polluted with interference or occupied by external traffic, effectively reducing the bandwidth available for team communications. Therefore, achieving a good global performance requires a good communication channel leading to the need for enhanced communication protocols that tackle the issues referred above.

In this section we start by surveying protocols developed for WSN given the amount of work developed in this domain for ad-hoc wireless communication. We then state the limitations of these protocols for use in teams of mobile robots and we present a survey of protocols specifically developed for this domain.

2.2.1 Communication protocols in wireless sensor networks

Exchanging information in ad-hoc networks has already been the subject of research in many areas. Among these, wireless sensor networks has been one of the most active in the last years. Due to the easy access to hardware level, common approaches completely or partially replace standard MAC layers in favour of specifically enhanced ones. Thus, there is a huge amount of very different approaches to MAC that can be found in this area. Here we present some of most representative MAC techniques developed for WSN, and motivate the need for different approaches when focusing on mobile robots.

2.2.1.1 Communications with schedules

Some of the initial approaches to WSN MAC protocols were based on low duty cycle periodic sleep and listening, and were designed for ad-hoc deployments where applications are expected to remain inactive for large periods of time, with only some periods of activity when events ocurred. Two well known examples are S-MAC [25] and T-MAC [26].

Both of these protocols were designed with energy conservation and self-configuration as the primary goals, consequently, fairness and latency are deemed less important. In order to save energy, a low duty cycle periodic sleep and listening is implemented, therefore nodes avoid idle-listening by only listening to the medium when packets are expected.

The S-MAC protocol uses a fixed listening time that may lead to periods of idle listening when there is no data to be transmitted. To avoid that, the T-MAC protocol listens until there is a silent period greater than a predefined time. The outcome of this strategy is that if there is no data to be transmitted, then the nodes will sleep earlier. If, on the other hand, there is data to be transmitted, nodes will stay awake for a longer time to receive it. In both cases neighbouring nodes synchronise their sleep schedules, or, in rare cases, adopt multiple schedules to keep network connectivity. To prevent nodes from never synchronising, from time to time they continuously listen for a complete period.

Another important result from T-MAC, is that unlike S-MAC it does not avoid overhearing. S-MAC switches off the radio when it learns from a control packet (RTS/CTS) that the next transmission is not meant for it. In the T-MAC protocol, the authors claim that this mechanism of avoiding overhearing has a deteriorating effect on throughput because awaking nodes are not aware of the medium, and might interfere with ongoing communications involving hidden nodes. Consequently, T-MAC nodes do not sleep during data transfers. Also, as an additional feature to improve throughput, the T-MAC protocol defines Future-Request To Send (F-RTS) packets that inform a node that there will be a transmission destined for it in the near future, thus keeping it awake even if it does not detect traffic in the medium.

2.2.1.2 Communications without schedules

A different approach is presented with the protocols B-MAC [27] and X-MAC [28]. In these protocols all nodes have independent schedules, therefore no exchange of schedule data is required. Instead, the nodes signal that they want to transmit data by transmitting a preamble.

The solution implemented by B-MAC is to transmit a preamble with duration greater than the duration of the sleep period. If a node awakes and detects the preamble, it will remain awake to receive the data, otherwise it will go back to sleep. However, since the preamble must be always transmitted, it will impose a reduction in throughput, which can be exacerbated when the data must be delivered in a multihop fashion. Moreover, not all radios support such long preambles.

The authors of X-MAC detected these problems, therefore they presented the following solution. Instead of using a long preamble, X-MAC proposes to use a series of shorter preambles containing the address of the destination node. Consequently, if the destination node awakes, it can respond with an acknowledge, thus starting the data transfer. Also, if a node that is not the destination awakes it can resume the sleep cycle. In addition, if a node that wants to transmit to the same destination hears the preamble, it will know that the destination will be awake in the end of the data transfer. Therefore it can start to transmit the data once the previous transfer completes, without sending the preamble. However, in both protocols, if an event is detected by several neighbour nodes, a high contention period will be generated over the medium. This will result in a large number of collisions and a large and variable delay in the transmission of the messages.

A similar approach is presented by the RI-MAC protocol [29]. However, instead of transmitters initiating the transmission, that role goes to the receiving nodes. To do this, nodes that have data to transmit listen to the medium waiting for the intended receiver. Once a receiver wakes up, it sends a beacon that informs a sender that it is listening. The senders will then contend for the medium and send the messages. When the receiver successfully receives a message it sends another beacon to initiate a new transmission. This beacon also serves as an acknowledge. Despite improving the performance when comparing to X-MAC, the problem remains. When all nodes have data to transfer, there will be a high number of nodes trying to transmit with high probability of collisions and a large and variable delay.

In the RC-MAC protocol [30], the authors improve the RI-MAC protocol by taking advantage of the tree structure to organise the transmissions of their children. To do this, they piggyback the ID of the next sender on the acknowledgement packets, sent after the first contention based transmission. Nonetheless, the management of the several tree levels is not coordinated, which may lead to large delays.

2.2.1.3 Communications using TDMA

A very well known technique used to avoid the contention in the communication medium created, for example, by the previously presented protocols is TDMA.

A simple approach is presented in the Funnelling-MAC protocol [31], which uses an hybrid approach between CSMA/CA and TDMA. Since the network forms a tree for data collection, it is known that the traffic increases as the nodes approach the sink node. Therefore, the protocol uses CSMA/CA as the default access control mechanism, and only in the high traffic area it defines nodes that synchronise their clock and use TDMA.

The TreeMAC protocol [32], takes advantage of the knowledge that WSN typically perform many-to-one communications, thus forming a tree from leaf nodes to a sink node, thus using an hierarchical frame assignment scheme. In this protocol bandwidth is divided in TDMA rounds, where each round is comprised of several frames. Each node is given a portion of these frames, for both its own and its children transmissions. In order to solve the hidden node problem, TreeMAC divides each frame in three sub-slots, where each one is assigned to a different hierarchy level. However, this protocol also requires global time synchronisation.

The Z-MAC protocol [33] also uses TDMA schedules to reduce contention under low and high traffic conditions. In both conditions, each node is the owner of a slot where it has the highest priority, however, other nodes can still transmit in its slot with a lower priority (slot stealing) implemented with a different backoff window. A slot can only be stolen as long as the network is in low traffic mode, or high traffic mode but the node is not a two-hop neighbour (avoiding hidden nodes). Despite that, this protocol requires clock synchronisation between neighbouring transmitters. Therefore the authors implement a synchronisation protocol that after an initial global clock synchronisation, only sends synchronisation packets proportionally to their outbound traffic. On the initial phase of configuration this protocol performs several operations, e.g. neighbour discovery, slot assignment (with spatial slot re-utilisation), and global clock synchronisation. These operations are only expected to run again if significant topology changes occur. Unfortunately significant topology changes is a scenario that is expected to occur frequently in mobile networks.

Another protocol that leverages the benefits of TDMA to reduce contention over the medium is iQueue-MAC [34]. The main design goal of this protocol is to support wireless communications within cluster-tree networks, specifically two level networks that are organised in different clusters. These clusters are connected amongst themselves through their heads (routers), and each cluster head then connects many sensors and is responsible of managing their transmission schedules. Under low traffic conditions, each cluster head uses a beacon to support low duty cycle with a phase for contention based access using non-persistent CSMA/CA (like IEEE 802.15.4). However, in each of the packets sent to the cluster head, nodes communicate the status of their transmission queues, allowing it to detect nodes with queued traffic (unlike RC-MAC transmitter queue sizes is not known). Whenever this happens, i.e. under high traffic, the cluster head (unlike Z-MAC static schedules) dynamically schedules guaranteed time slots to those nodes, thus avoiding collisions and countering the effects of medium overload. However, the cluster structure of this type of protocols is not adequate for teams of robots where mobility creates ever changing and odd shaped topologies, where information is to be shared across different end points.

2.2.2 Communication protocols in robotic applications

Wireless sensor networks have indeed explored many different protocols and medium access methods, however, latency and throughput is rarely an issue, and mobility, when considered, is very limited. Consequently, the robotics wireless communication community, had to tackle these problems using different techniques, where energy efficiency is a secondary concern, and message throughput and delay is the primary focus.

2.2.2.1 Dynamic Implicit-EDF for teams of mobile robots

In the work presented in [35] the authors propose a communication protocol for Mobile Ad-hoc Network (MANET) comprised of robots. This is an enhancement of a previous technique proposed for scheduled wireless sensor networks called Implicit-Earliest Deadline First (EDF) [36]. The proposed protocol, which we refer to as Dynamic Implicit-EDF (DI-EDF), requires global clock synchronisation to achieve real-time traffic scheduling while coping with dynamic team membership and dynamic traffic requirements. Robots are allowed to join the team, leave the team, or simply crash, and therefore the protocol can dynamically manage the addition, modification, and removal of messages and their requirements. This is done through a consensus process that, after ensuring that the bandwidth requirements after the change can be accommodated, transmits the new configuration throughout the network, in bounded time. Each message is transmitted in a slot of constant duration $T_{\mathtt{slot}}$. Therefore, in order not to waste bandwidth the time $T_{\mathtt{slot}}$ must be set according to the maximum message duration. However, the impact of global clock synchronisation errors must be taken into account when calculating the maximum slot duration. The presence of external traffic in this protocol is not supported. Consequently, if a node outside the team starts transmitting without respecting the protocol, the message deadlines will not be respected.

In order to reach (and trigger) the consensus, each robot transmits periodically every T_{sync} a message containing its network requirements, the status of the network, the local clock value, and other information related to the consensus process. This periodic message is transmitted in a round-robin fashion, i.e. if in period n robot k transmits the message, then in round $(n+1)$ robot $(k+1)$ transmits, and in round $(n+N-1)$ robot $(k+N-1)$ transmits, etc.. The period of this message can be adjusted according to the system requirements. However, although larger periods reduce the bandwidth requirements, they also reduce the dynamics supported by the protocol. Conversely, smaller periods decrease the bandwidth requirements but increase the supported dynamics. Despite that, according to the authors, when using 10 robots communicating at 1Mbps , with $T_{\mathtt{sync}} = 20\mathtt{ms}$, and $T_{\mathtt{slot}} = 1\mathtt{ms}$, this message uses 10% of the bandwidth, and the robots may need up to $2s$ to reach a consensus.

2.2.2.2 Real-Time – Wireless Multihop Protocol

The Real-Time – Wireless Multihop Protocol (RT-WMP) [37] family of protocols are designed to support hard real time traffic in MANET, containing up to ten to twenty units, based on a token passing scheme. The unicast version of the protocol is presented in [38] and runs in three phases. In the first phase the token circulates the network to decide which is the node with the highest priority ready message (Priority Arbitration Phase). Then, in the second phase, the token is passed to the unit with the highest priority, thus giving it permission to transmit (Authorisation Transmission Phase). Finally, in the third phase, the authorised node transmits the message to the destination and finishes the round (Message Transmission Phase). The token pass scheme translates in a very high overhead, detrimental for the throughput. In this protocol the worst case scenario would require $2n-3$ hops for phase one, $n-1$ hops for phase two and $n-1$ hops for phase three; totalling $4n-5$ hops, albeit only the last phase transmits the message data, where n is the number of robots.

This number of hops can have even a greater impact if data is to be delivered to several robots (multicast). For that reason, this protocol was extended in [39] by adding to it multicast capabilities (RT-WMP-PME). To do that, in every frame of the protocol a tail is added which contains a multicast message (if there is one). In this case, however, a multicast message may need to traverse the whole network to be delivered to *all* robots, which translates in a much higher number of hops $(6n-13)$, specifically $4n-9$ PAP hops, $n-2$ ATP hops, and $n-2$ MTP hops. In order to reduce this time, it was also proposed to use only the multicast protocol (RT-WMP+), where unicast would be treated as a special case of multicast. The advantage of the latter would be removing the three phase scheme, since only a continuous PAP phase would be used. This means that using this scheme all the messages now carry data, and that only $4n-9$ PAP hops are required for delivery. However, when compared to the initial RT-WMP in terms of unicast end-to-end delivery delays, it is better only when dealing with high data rates, short payloads, or a small number of nodes.

Due to the inherent problems of the wireless medium, a message transmitted may not reach its destination, the receiver may crash or move in the process. In order to avoid such situations from creating unexpected protocol behaviour, the sending node listens to the medium for the next transmission, which also works as implicit acknowledge, but for as long as a timeout interval, only. If during that timeout interval the next transmission is not heard a failure is assumed and appropriate measures are taken. However, if this protocol needs to coexist with external traffic, this timeout may not be respected for a reason other than failure. Therefore, the protocol is extended in [40] to support external traffic. In this case, the timeout can be extended according to detected external traffic, bounded by a maximum time to avoid a potentially infinite extension. Although this protocol is able to respect the real-time requirements it proposes to, it suffers from some drawbacks inherited from the method it uses. The multicast feature is the one more likely to be used by cooperating robots, i.e. from formation control to cooperative tracking all robots are interested in most of the sensory information from all or several other robots. However the delay of such messages in the transmission queue may be in the range of 225-350 milliseconds. Finally, one of the drawbacks of this protocol is that if a token is lost, e.g. the robot crashes after the acknowledge, until that event is detected and a recovery routine is performed, the whole network stops. Therefore the crash of a single node affects the whole network.

2.2.2.3 WIreless Chain networK Protocol

The WIreless Chain networK Protocol (WICKPro) protocol [41, 42] is another token based approach that aims to provide hard real-time guarantees for Wireless Mesh Network (WMN) in chain topologies. Assuming chain topologies is a typical assumption when working in tunnels or other tight spaces where robots can only move in one direction. Consequently, this protocol is only applicable to a very limited set of scenarios. This work defines a macro-cycle that is subsequently divided in micro-cycles. Then, within each micro-cycle the protocol works similarly to a TDMA schedule where each slot is explicitly started and ended using a token, this token carries the transmission schedule that is managed by the token master (the node that is in the root of the chain). Nevertheless, albeit an effective solution to manage access to the medium, using a token could lead to high delays, for example when a token is lost.

2.2.2.4 Reconfigurable and Adaptive TDMA

The RA-TDMA [11, 43] is a protocol developed within the Cooperative Autonomous Mobile roBots with Advanced Distributed Architecture (CAMBADA) [44] robotic soccer team for the RoboCup Middle-Size League (MSL). Thus, due to the rules of that league,, this protocol was created to be used with a managed network. Despite that, the fact that the protocol relies on a TDMA scheme, makes it interesting for ad-hoc applications by mitigating the hidden-node problem. The fact that in robotic soccer according to the rules of this league, two different teams (usually using different protocols) need to share a single wireless channel, makes one of the focus of this communication protocol the resilience to external traffic. For that purpose, the protocol maintains a loose synchronisation amongst the team members, i.e. without global clock synchronisation, which adapts to delays in the packet transmissions, and reconfigures according to the number of active robots. This protocol uses a predefined TDMA round period T_{up} that sets the responsiveness of the communications. Then, that round is divided in N slots, and each slot is allocated to a single robot, where N is the number of active robots. Since each robot only has one slot per TDMA round all data is piggybacked in a single packet and broadcast, thus minimising the number of transmissions and consequently the contention at the medium access. This dynamic number of slots maximises the space between communications originating in the team, consequently, even if external traffic causes a delay in a transmission, the contention amongst team members is virtually eliminated.

In order to maintain the TDMA round without global clock synchronisation, this protocol synchronises on the reception of messages, which can be affected by clock drifts, external traffic, or other interference (e.g. operating system). To cope with that the protocol absorbs the delays by shifting the phase of the round, up to a bounded value, thus also delaying the following team transmissions. This not only keeps the transmissions synchronised, but also (in situations where the interference is periodic or near periodic) helps to avoid external traffic. Experiments presented in [11] show, without synchronisation, the robots may, at some point, transmit very close to each other in time, leading to high transmission delay even without external interference (some experiments showed delays up to 20ms without and 30ms with interference). Conversely, when the protocol is used, this value is greatly reduced (to 7ms without and 10ms with interference, in the same experiments).

As mentioned before this protocol was created to work in managed networks, which simplifies some of the communication issues. First of all, since in a managed scenario all robots are at one-hop distance from each other, i.e. not counting the access point, every robot can communicate directly with all the team members, the propagation of information takes one TDMA round. Adding to that the team membership is implicit, any robot that is within the reach of the AP it is part of the team.

2.2.2.5 Seamless Mobility within Industrial Wireless Mesh Networks

Another example of an infrastructure based approach is the Seamless Mobility within Industrial Wireless Mesh Networks (SM-IWMS) [45] protocol, where in order to provide QoS guarantees for communicating mobile robots, the authors propose to use a WMN as a replacement for a wired backbone. The main objective of this work, is to provide a dependable communication protocol for mobile robots, therefore, there is an emphasis on link failure detection, message recovery mechanisms, and admission control with the objective of avoiding network overloads. The focus of this work is naturally interesting for all types of wireless networks, since link failure affects the performance of all solutions. Nevertheless, this thesis is more focused on the management of the transmission instants, which is an orthogonal problem that is not considered in SM-IWMS, where the authors rely on standard protocols. Moreover, we assume that we have no fixed infrastructure to rely on.

2.2.3 Discussion

Looking back at the WSN communication protocols, we can see that they are appropriate to the applications they were designed to, but they make some assumptions that are not valid in high mobility robotic networks. First of all it cannot be assumed that there is a flow of information to one single sink node. In mobile robot teams it is rather common to find *n*-to-*n* communication patterns. Also, assuming mobility in WSN is either non-existing or very limited, for example considering only a few mobile nodes within an essentially static network [46]. High mobility implies high limitations to message validity, frequently leading to high data rate requirements (e.g. position). Consequently, not only delays in message delivery are very important, but also the amount of traffic is higher than in WSN, e.g. every robot wants to transmit some information every computation period which is usually no longer than 100ms. This increases the need for fast and reliable delivery[1], specially when there is uncontrollable external traffic. Finally, while the main focus of WSN communication protocols is to save energy, in mobile robots, the presence of motors and generally high power processors deem the energy wasted in communications negligible. To sum up, the existing protocols designed for WSN cannot be efficiently applied to high mobility robot teams.

The protocols designed for robot communications, naturally attempt to cover those limitations. The first protocol, DI-EDF, provides real-time guarantees, however, it does not provide support for high dynamics, requires global clock synchronisation, and is not resilient to external traffic. The RT-WMP on the other hand, in addition to providing real-time guarantees, is resilient to external traffic, and supports highly dynamic reconfiguration. Despite that, it is subject to long message delivery delays, specially in cases where there are link failures or robot crashes. For example, since the system relies on the token, if the robot with the token crashes, the system takes some time to regenerate the token, or if a link fails, the delay of the token will affect the whole network. The WICKPro protocol presents the same token related problems, and in addition, is was designed for very specific robot applications (chain topologies). The SM-IWMS, similarly to other protocols designed for using WMN (Wireless Mesh Networks as an infrastructure), is not adequate for the applications we aim for, where robots need to be deployed immediately. The RA-TDMA, provides a protocol that is extremely resilient to external traffic, trying to adapt to it, can bound the transmission time, and uses broadcast to make the message propagation as efficient as possible. Despite that, it does that in a managed environment, thus assuming that the network is fully connected. This is not the case in ad-hoc scenarios, in which the network may be not fully-connected, therefore both operational data as well as the membership information have to be propagated by the team members to cover the whole network and reach consensus, which is known to be a complex problem in a distributed dynamic network. Nevertheless, it seems an an effective, despite challenging, solution to ad-hoc networks, which was one of the motivations for this thesis. In particular, extending RA-TDMA to ad-hoc networks allows an interesting comparison between an event-triggered (RT-WMP) and a time-triggered (RA-TDMA) approach in ad-hoc networks, highlighting the pros and cons of each one. Table 2.1 shows a qualitative comparison among the referred protocols focusing on some relevant key aspects.

---

[1] in the sense that most messages are delivered without retransmission

Table 2.1: Qualitative comparison between communication protocols

Protocol	Medium Access	Resilience to external traffic	Real-time guarantees	Support for mobility	Topology
S-MAC	CSMA	No	No	Limited	Mesh
T-MAC	CSMA	No	No	Limited	Mesh
B-MAC	CSMA	No	No	Yes	Mesh
X-MAC	CSMA	No	No	Yes	Mesh
RI-MAC	CSMA	No	No	Yes	Mesh
RC-MAC	TDMA/CSMA	Yes	No	Yes	Cluster tree
Funnelling-MAC	TDMA/CSMA	Yes	No	Yes	Cluster tree
TreeMAC	TDMA/CSMA	No	No	Yes	Cluster tree
Z-MAC	TDMA/CSMA	Yes	No	Limited	Mesh
iQueue-MAC	TDMA/CSMA	No	No	Yes	Cluster tree
DI-EDF	TDMA	No	Yes	Yes	Mesh
RT-WMP	Token-pass	Yes	Yes	Yes	Mesh
WICKPro	Token-pass	Yes	Yes	Yes	Chain
RA-TDMA	TDMA/CSMA	Yes	Yes	Yes	Star
SM-IWMS	Not defined	Yes	Yes	Yes	Mesh infrastructure

Chapter 3

An Overview of Wireless Localisation Techniques

The location of the nodes in a network of mobile robots is frequently an essential information to put in practice a diversity of coordination algorithms, such as team formation and path planning. For example, the work in [47] explores using feedback laws to control multiple robots together in a formation. However, this work assumed that each robot has the ability to measure the relative position with respect to its closest neighbours. Also, in [48], the robots path is computed to ensure that the network partition never occurs during the robots motion, but the knowledge of global location (e.g. GPS) is assumed to be available at each robot. The work in [49] explores the sensor relocation in order to deal with sensor failure or respond to new events. Methods of finding redundant sensors and moving sensors to specific areas are proposed, assuming that sensors are placed into grids and global information is shared to support relocation planning. None of these works, however, consider the practical position management of mobile robots or sensors.

In this chapter we present a brief survey of the most common techniques used to derive a notion of localisation in teams of mobile robots, using the RF interface. We first discuss ranging methods to measure pair-wise distances and then we analyse how to deduce localisation based on the ranging information. We conclude the chapter with a discussion of the presented techniques and we draw the motivation for our work in this topic.

3.1 RF Ranging Techniques

Ranging means measuring distances. In this section we briefly discuss different existing methods to carry out ranging using RF communications. We broadly classify them in time-based and RSSI-based techniques, with a few hybrid cases.

3.1.1 Time based techniques

Time-based methods essentially measure intervals of time and convert them to distances or angles knowing the speed of propagation of the RF waves through air. Since this speed is very close to the speed of light in vacuum, for relatively short distances and angles the involved time intervals are rather short. Thus, measuring these intervals typically requires special hardware support and specific techniques, such as computing cross-correlation of signals. Nevertheless, these measuring limitations lead to limitations in the achievable precision. These ranging techniques have been frequently used in static sensor positioning [50] where the dominant time-based techniques to obtain distance measurements include the Time of Arrival (ToA), the Time Difference of Arrival (TDoA), Angle of Arrival (AoA), and the RT-ToF, which we describe next.

3.1.1.1 Time of Arrival

Time of Arrival measures the time a given message needs to travel between the sender and several receiving nodes [51, 52, 53]. To do that, the sending node timestamps the message with the global time and sends it. Then, the receiving nodes timestamp the arrival using the global clock and calculate the time it took the message to get to them. Knowing that the waves travel at the speed of light, the nodes calculate the distance between themselves and the sender. This method requires that all the nodes are very well synchronised to guarantee a good precision of the global clock. In fact, just a small error in the clocks can translate into a large distance error.

3.1.1.2 Time Difference of Arrival

This method, similarly to ToA measures the time a given message needs to travel between nodes, but unlike ToA it measures the time difference between receiving times in different receivers. This method requires at least four receiving nodes to determine the position of the sender through multilateration and measure its distance, i.e. tracking the sender. Just the nodes involved in tracking need to be synchronised, thus partially solving the global clock synchronisation issues [54, 55]. Nevertheless, this method only brings advantages to situations where some nodes can be easily synchronised to track the sender. In the work presented in [56], the clock synchronisation limitation is solved by sending, simultaneously, both an RF and an acoustic signal. Using both signals, namely the difference on their time of arrival, and the difference in the propagation speed of both waves, this approach calculates the time on which the signal was transmitted. However, this simply replaces the error caused by clock drift, by the error of the estimation of the speed of sound, which depends heavily on air temperature. Nonetheless, in this thesis we are interested in RF *only* solutions, so we will not focus on these approaches.

3.1.1.3 Round-Trip Time Of Flight

RT-ToF is a technique that eliminates the need for global clock synchronisation [57]. This technique is very similar to ToA in that it allows measuring the time a message needs to travel between communicating nodes. However, in order to remove the need for global clock synchronisation, instead of measuring the time of one-way trip, it measures the time that a message needs to go to the receiver and return to the transmitter. The precision of this method is highly dependent on the precision of the time taken in the receiver to process the incoming message and send it back to the sender. Therefore, for high precision, this needs to be done in hardware. Another feature of this method is that, since the ranging operation involves two nodes at a time, it may take a long time to range several nodes, limiting the applicability to fast moving robots.

3.1.1.4 Angle of Arrival

This technique involves measuring the angle from which the received message comes from, thus obtaining the direction of the transmitter node [58, 59]. Typically, this method uses at least three receivers in known non-co-linear positions that are synchronised. Then it works similarly to TDoA but the time differences are used to compute the angles of arrival with respect to each pair of receivers. Then, through multiangulation, it is possible to determine the relative position of the sender and thus its distance.

3.1.2 Received Signal Strength based technique

Another method to perform ranging between communicating nodes is to measure the RSSI. As the name implies, RSSI-based methods, obtain range estimation from the strength of the received RF signal [60]. In open space and without interference there is a well known logarithmic relationship between RSSI and distance, however, in the presence of interference, reflection, and refraction, this relationship can be easily destroyed. Nevertheless, since the referred time-based methods require specialised and expensive hardware for accurate time measurements or for precise global time synchronisation, it might be appealing to pursue an RSSI-based approach in cases in which the application only requires coarse localisation, either for navigation or topology estimation purposes. Due to their dependence on the operational environment, RSSI-based methods require calibration based on a priori channel measurements[61, 62]. This is the main problem of these methods, since those measurements may be unavailable or unreliable, i.e. either there is no previous knowledge or there were changes in the environment. For this reason, some methods perform online channel estimation using measurements with and between predefined statically located anchor nodes [63, 64] which, however, is not compatible with unknown environments. Other methods perform callibration with external sensors [62] thus requiring extra equipment.

3.1.3 Hybrid techniques

As the work in [65] shows, hybrid techniques can greatly improve localisation accuracy. In that work, the authors compared localisation using combinations of ToA, TDoA, and RSSI, showing that RSSI has limited usefulness whenever time-based techniques are available. On the other hand, the work in [15] showed that ToF ranging requires a long time to range each pair of robots in a team while RSSI allows several receivers to “range” one transmitter simultaneously, thus making RSSI appealing for applications with mobile robots where the dynamics of the movements are not negligible. Therefore, solutions such as [66] use an hybrid approach fuses RSSI and round-trip time-of flight measurements. However, this work assumes that channel parameters are estimated in advance. As part of this thesis we will also propose a hybrid RSSI/RT-ToF method but which estimates the channel parameters online, thus adapting to changing environments.

3.2 Deriving Robots Positions Using RF Signals

In order to support global cooperative application we typically need to estimate the topology of the network. The ranging methods defined in the previous section are essentially steps to reach this goal. In some cases, such as TDoA and AoA, some relative position information is already discovered. However, this is not the general case with ranging methods that just provide pairwise distances. In this section we will discuss the generation of positioning information from ranging values. We will generally classify the localisation methods in absolute and relative positioning. The former localises each robot in a fixed physical frame, e.g., a building or a generic geographical area, while the latter localises the robots in a team with respect to each other, only, thus creating a floating topology with respect to the physical space the team is in.

3.2.1 Absolute positioning

A common approach to absolute localisation is Simultaneous Localisation And Mapping (SLAM). A famous example is the FastSLAM algorithm [67], where a particle filter is employed to track several possible paths of the robot, and extended Kalman filters to estimate the positions of landmarks. Another example is Wifi-SLAM [68] where an automatic fingerprinting technique that exploits landmarks on the radio map is proposed. By fusing RSSI information with inertial data from an Inertial Measurement Unit (IMU) unit, it is possible to detect loop closure and to build the environment map and locate the user. Another interesting work is presented in [69], where authors propose a range only SLAM. This work proposes a technique using dead reckoning to track robot movement and ultrasound ranging equipment to measure the distance between itself and some beacons. The beacons have unknown positions but are able to measure the distance between themselves. MapCraft [70] assumes that a physical map containing the walls and doors is already available. Then, data from different sensors is fused and matched to the map to estimate positions. The work of [71] proposes a method of estimating nodes positioning using a Maximum Likelihood Estimator (MLE). The authors collect RSSI and ToA measurements of static nodes and, together with the location of four anchor nodes with known location, they estimate the positions of the nodes. In order to obtain distance measurements from RSSI values, the authors estimate the parameters of the path loss model before runtime with data collected in prior experiments. In general, these techniques require the ability of measuring some static features, such as landmarks, walls, or the RSSI fingerprints of certain access-points.

3.2.2 Relative positioning

A very popular solution for calculating nodes relative positions from ranging information is the MultiDimensional Scaling (MDS) algorithm [72, 73] that minimises the dissimilarities of a connectivity matrix up to a rigid formation. For example, in [74], the authors propose a method of deriving the network topology from the RSSI data using MDS. The method presented is not a physical accurate localisation system, first because of the RSSI behaviour, but also because the work does not consider any propagation model. Nevertheless, the method presented is precise enough for its application scope, i.e., helping wireless connected units manage the relative positions without central supervision. In [75, 76] the authors propose two methods of estimating nodes positioning using an improved version of MDS to obtain node locations based on the distance information called MDS-MAP(C) and MDS-MAP(P). The former, after running classic MDS uses an extra step to transform the relative positions to a global frame using anchor node information. The latter, applies MDS-MAP(C) to a set of nodes up to a maximum of n hops. Then, each of the generated maps is stitched together with the neighbours until a map of the full network is produced. Finally relative positions are adjusted to a global frame using anchor node information. Also in the same work, the authors present an extended version of both algorithms, MDS-MAP(P,R) and MDS-MAP(C,R). These variants apply a final refinement step, using least-squares minimisation to make the distances between neighbouring nodes better match the provided measured distances. Despite performing better than the regular MDS, this solution carries extra computational cost, that may not be worth in small sized networks with just a few hops. In order to improve results under unknown line-of-sight/non-line-of-sight (LOS/NLOS) conditions and scarce ranging information, [77] uses another variant of MDS based on Weighted Least Squares algorithm, whose weights are assigned according to the reliability of the ranging measurements. The work in [78] provides a complete theoretical analysis of node localisation using MDS. However, most of the existing work assumes known position of anchor nodes, which can be unavailable in many scenarios. Adding mobility to sensor nodes makes localisation more complex and uncertain. The work in [79] employs Monte Carlo Localisation methods to improve accuracy. Nodes that know their own location – called seeds – and nodes with unknown location form a network, where at least one kind of sensors is moving. Despite the improved localisation, Monte Carlo solutions are computationally expensive, consequently very hard to implement in real-time. Another approach to mobile robots localisation using MDS can be found in [80]. In this work node positioning is improved with extra hardware, i.e. known robot movement, thus increasing the cost. And finally, in the work presented in [81] a method of estimating the positions of moving nodes in an anchor-less scenario is proposed. The authors use relative velocity, calculated based on dead-reckoning, together with RSSI measurements to provide a position estimate of a team of mobile nodes without resorting to anchor nodes.

3.2.3 Discussion

Time based ranging techniques using wireless transceivers typically require some degree of clock synchronisation to be able to perform well. The Round-Trip Time of Flight, conversely, does not need clock synchronisation, and thus seems very interesting in our scope. Nevertheless, this ranging method is done explicitly between two communicating nodes. Therefore, the rate at which it can be done is quite low, and since it requires a query and a response its bandwidth requirements can not be neglected. RSSI based methods, on the other hand, require no clock synchronisation and use no extra bandwidth, but the need of a channel model for higher accuracy may be a problem. Hybrid solutions are much more attractive, since multiple technologies can be combined to preclude each other disadvantages. In this work we will focus specifically in using the higher accuracy of RT-ToF measurements to improve the accuracy of a faster RSSI-based distance estimator by recurrent online recalibration. Table 3.1 shows a qualitative comparison among the referred methods focusing on some relevant key aspects.

Table 3.1: Qualitative comparison between ranging methods

Method	Requires global clock	Accuracy	Degree
ToA	Yes	High – Depends on clock synchronisation	one to many
TDoA	Amongst receivers	High – Depends on clock synchronisation	one to many
AoA	Amongst receivers	High – Depends on clock synchronisation	one to many
RT-ToF	No	High – Requires specialised hardware	one to one
RSSI	No	Low – Depends on channel model	one to many

As for obtaining positions from the ranging information, we aim at obtaining relative positions using solely the RF transceiver. Therefore the use of landmarks or objects with known positions is outside of our objective. Algorithm simplicity is also a concern, since many robots have limited computing resources, thus MDS is a good match to our aims. Moreover, when localising robots, and beyond the estimation accuracy, it is important to be able to provide information about the confidence of that estimation. To the best of our knowledge there are no works computing confidence regions associated to MDS computed position estimates. Consequently, this is a topic worth exploring, which is part of this thesis.

Chapter 4

A Communication Protocol for Robot Ad-Hoc Networks

Whenever new information is produced in a robot, and that information is important for other robot actions, intuition tells us that it should be distributed immediately. Albeit beneficial, since fast delivery to the interested robots maximises the utility of that information, this strategy has a pitfall, increased contention for the wireless medium. Contention in the network happens when multiple wireless nodes try to send their messages within a short time interval. As an example, consider that one event in the physical world generates some data in a robot, e.g. an intruder entering a protected region of space. If multiple robots detect that event, then all of them would try to transmit their data at the same time, leading to an increase in the number of collisions and unpredictable delivery delays. Another example comes from within the robots themselves, many of the applications running within the robots involve repetitive processes such as image capture and analysis, control loops, and sensor acquisition, consequently, they often produce periodic data. Since the periods are governed by different internal clocks, that drift from each other, these processes will eventually occur in very short intervals in multiple robots, again creating multiple simultaneous transmissions and all the problems that derive from them [15][82].

When only a small number of robots compete for the medium, standard mechanisms can solve these multiple accesses, however, when many robots are competing for the wireless medium, these mechanisms become ineffective, thus leading to multiple packet collisions, and consequently packet losses. Even if we have robust retransmission mechanisms, these collisions will increase the number of re-transmissions and the amount of backlog stored in the transmission queues. Consequently, the end-to-end delay of all packets will be increased. Moreover, if we consider multi-hop networks, the interference caused by simultaneous transmissions can be rather strong as packets from robots hidden from each other are sent within small time intervals (Figure 4.1). Note that in these circumstances, called critical interference periods [83], the typical Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) arbitration mechanism is ineffective since hidden robots do not hear each other.

In order to solve the problem of critical interference periods, we developed a protocol that, not only manages periodic communications, but also deals with some common situations in dynamic environments, namely, joining and leaving of team members, dynamic topology changes, hidden robots inside

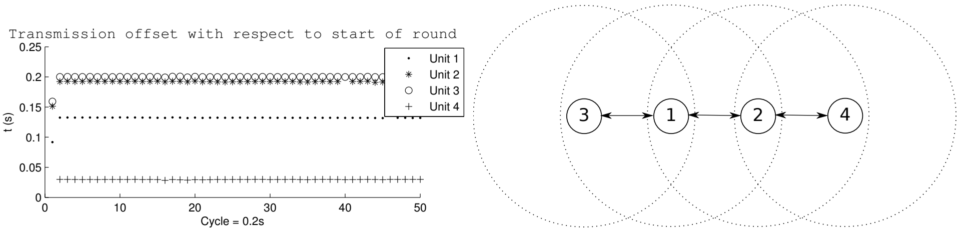


Figure 4.1: Example of a snapshot with hidden robots overlapping their transmissions – robots 2 and 3 may interfere with each other in the intersection of their ranges causing robot 1 to lose both packets

the team, and transmissions external to the team. Moreover, our protocol propagates the synchronisation needed through the network and a consensus algorithm ensures convergence of the synchronisation. According to the definition in [84], a consensus means reaching an agreement regarding some quantity of interest that is dependent on the state of all agents. Similarly, a consensus algorithm is defined as a rule that defines how neighbouring agents interact, namely how they exchange information, in order to reach a consensus. In the literature, we find two types of consensus algorithms; consensus using explicit exchanges of information and consensus using *implicit* information, i.e., using sensors to measure the state of other agents. A common application, where explicitly shared information is used, is multi-vehicle control, e.g. the *rendez-vous* problems presented in [85] and [86]. In those examples, robots share their current position with other team members to reach a consensus on a cooperatively agreed position. In our approach, we want to synchronise transmissions, however, because we do not use a global clock, there is no definition of an absolute time reference. As a consequence, it is not possible to use an explicit form of consensus, instead, we approach the problem implicitly with the packets reception times. This has been used before in [87], where synchronisation of a periodically emitted pulse is achieved by a group of agents. In this approach, no information about the network is collected (neither number of members nor topology), therefore, the agents cannot transmit in different phases. Instead, whenever an agent senses the pulse coming from a neighbour, it delays or advances its own pulse to try to match it. Due to their approach, in some cases agents can be delayed indefinitely, thus some stochastic misfires have to be introduced to avoid those situations. In our protocol, we collect information about the team composition, consequently, we use that information and avoid those issues.

In this chapter we describe such protocol, which is an extended version of the Reconfigurable and Adaptive TDMA (RA-TDMA) protocol already developed for infrastructured networks. In particular, we extend the protocol for ad-hoc operational scenarios and develop a consensus protocol that ensures convergence of the synchronisation.

4.1 Relevant Publications

The materials in this chapter were partially published in the following paper that presents the main mechanisms of the proposed protocol. The proof of convergence of the synchronisation mechanism is part of another paper currently in preparation.

- [15] L. Oliveira, L. Almeida, F. Santos, A loose synchronisation protocol for managing RF ranging in mobile ad-hoc networks, in: T. Rfer, N. Mayer, J. Savage, U. Saranl (Eds.), RoboCup 2011: Robot Soccer World Cup XV, Vol. 7416 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 574–585.

4.2 Reconfigurable and Adaptive Ad-Hoc Synchronisation Protocol

Our approach follows the RA-TDMA protocol [11] in which the team robots transmit in a fixed and predetermined periodic round. Each robot transmits once per round, and the size of the round sets the reactivity of the communications. The round is divided in a dynamic number of slots according to the current number of robots in the team. Similarly, we also use an underlying medium access protocol that provides CSMA/CA arbitration, reducing the collisions with transmissions of robots external to the team, and even among team robots while the slot structure of the Time Division Multiple Access (TDMA) round is being adjusted. The main purpose of RA-TDMA, which we keep in our protocol, is to separate the transmissions of the robots in the team in time as much as possible, without using global clock synchronisation. This is done synchronising on the receptions of the packets sent by the other robots in the team that, as shown further on, will allow robots to reach a consensus on the structure of the communication round.

There are also several approaches to reaching a consensus, e.g. mean-consensus where agents agree on a mean value of all the states, and max/min-consensus where agents agree on the maximum/minimum value of all the states. Since moving towards the mean or the minimum value implies the possibility of anticipating the next transmission, thus consuming more bandwidth, we want to avoid these forms of consensus. Consequently, using an approach similar to the original RA-TDMA, we will explore the max-consensus problem. Unlike the original proposal, however, our protocol must cope with ad-hoc networks and dynamic topology, which requires new approaches to the propagation of the information in the network, to the agreement on the slots structure and assignment at each instant, and to the synchronisation itself. In this section we will provide a detailed explanation of those mechanisms.

4.2.1 Problem statement

In our proposal, we consider a team $\mathcal{T}$ of N_{max} robots numbered between 0 and $N_{max} - 1$, i.e. $\mathcal{T} = \{r_i\}$ where $i \in [0, N_{max} - 1]$. At each time instant, N of those robots are active, therefore, we define $\mathcal{T}_A$ as the set of size N that contains all of the active robots, i.e. if robot r_i is active then $r_i \in \mathcal{T}_A$.

Within this framework, our objective is to provide a protocol capable of synchronising transmissions of the active robots in order to maintain a TDMA round. The size of the TDMA round period, is configured to the same value for all robots ($T_{\mathtt{up}}$), thus defining how often the information is updated in each robot. Consequently, the current number of active team robots N is a fundamental parameter for the proposed synchronisation protocol, since each robot autonomously divides this period in a number of slots equal to N, where each slot has the duration of $t_{\mathtt{slot}} = T_{\mathtt{up}}/N$. Each slot n is uniquely assigned to one robot, and even though spatial re-utilisation would be possible, our approach does not include it to avoid a strong dependency on the robots spatial topology.

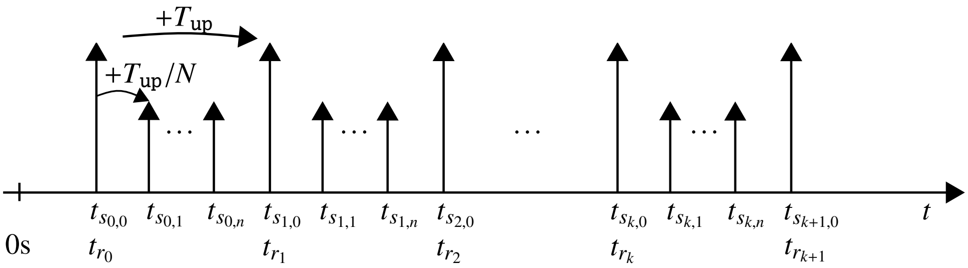


Figure 4.2: The structure of the TDMA round: In this figure t_{r_k} represents the beginning of round k and $t_{s_{k,n}}$ represents the beginning of slot n of round k; As depicted on the top, the round duration is $\mathtt{T}_{\mathrm{up}}$, and the slot duration is $\mathtt{T}_{\mathrm{up}}/N$ where N is the number of active robots.

Using the round structure, as depicted in Figure 4.2, we can write the time at which round k starts (t_{r_k}) as in Eq. 4.1, where t_ϕ represents the offset of round zero, i.e. $t_\phi = t_{r_0}$. Then, we can write the time $t_{s_{k,n}}$ of each slot n in a round k relatively to the beginning of that round as in Eq. 4.2, where n is the number of the slot. Note that $\forall_k : t_{s_{k,0}} = t_{r_k}$ and that $\forall_k : t_{s_{k,N}} = t_{r_{k+1}}$.

$$t_{r_k} = t_\phi + k \times T_{\mathrm{up}}\texttt{, where } k \in \mathbb{N}_0 \tag{4.1}$$

$$t_{s_{k,n}} = t_{r_k} + n \times \frac{T_{\mathrm{up}}}{N}\texttt{, where } n \in [0, N) \tag{4.2}$$

The structure of the round, will eventually allow us to measure the delays of packets transmitted by each robot, but before that, we need to assign a slot to each robot, and track the value N. Remember that N changes throughout the lifetime of the team, whenever a robot becomes active, inactive, crashes, or simply moves away from the range of the team.

4.2.2 Propagation of information

In order to keep information produced in different robots coherent within the team, we need to be able to share and maintain this information throughout the network. In our approach this is done with a broadcast protocol that disseminates a set of shared variables, each having one single producer and multiple consumers. This protocol makes use of a set of controls that regulate the updating of those variables in order to enforce consistency between the copies at the producer and consumers. They ensure that newer produced data eventually reaches all copies of each variable at the consumers as well as that stale information is detected and removed following a robot crash, departure from the team, or simply a link rupture. These controls are the following:

1. Local time-stamps, indicating the freshness of the data
    - one time-stamp per shared variable
    - time-stamps are refreshed when their respective variable is updated (t_u), allowing to control the age
    - a variable is erased if not updated after a pre-set variable-dependent validity interval ($t_{\mathtt{val}}$)

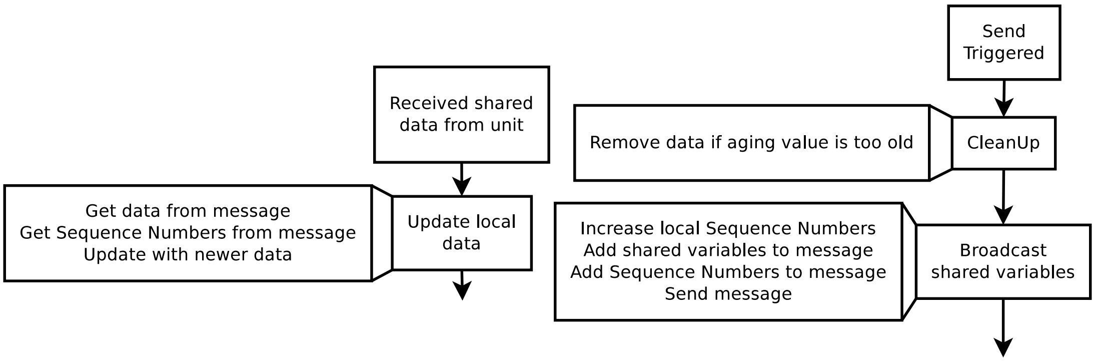

Received shared data from unit
Get data from message
Get Sequence Numbers from message
Update with newer data
Update local data
Send Triggered
Remove data if aging value is too old
CleanUp
Increase local Sequence Numbers
Add shared variables to message
Add Sequence Numbers to message
Send message
Broadcast shared variables

Figure 4.3: Broadcast protocol - sending and receiving procedures

2. Sequence numbers indicating between copies of the same shared variable which is the one containing fresher information
    - one sequence number per shared variable
    - each sequence number is increased by the producer robot right before it is sent together with the new information
    - larger number corresponds to newer data

Finally, note that each robot cleans up its own variables, i.e., removes (deletes) stale information, every time it broadcasts them, just before transmission. This means removing all variables for which $t_{\texttt{now}} \geq t_u + t_{\texttt{val}}$ (Figure 4.3), where $t_{\texttt{now}}$ is the current broadcast instant.

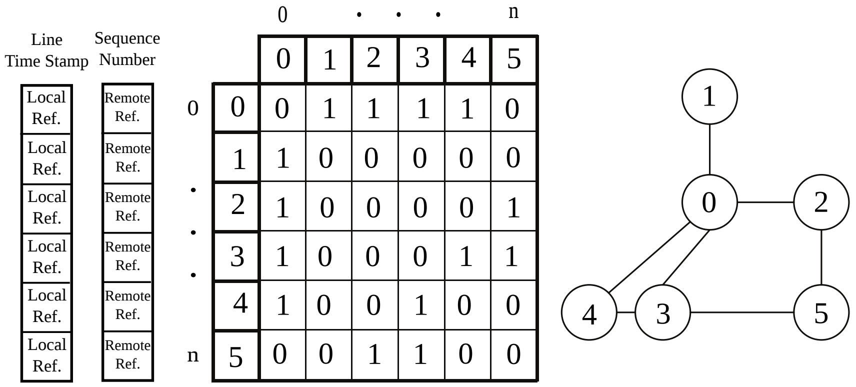
4.2.3 Slot assignment

As previously stated, each robot autonomously divides the TDMA round in a number of slots equal to N, each uniquely assigned to one robot. To do that, we resort to a slot allocation table, that is derived locally based on the knowledge retrieved from the topology of the network.

To share the topology of the network, similarly to [35], we create in each robot r_n a connectivity matrix $M^n_{N\times N}$, whose element (i, j) can represent a connection (1), or a lack of a connection (0) from robot r_j to robot r_i (i.e., r_i listens to r_j). Each robot r_i writes in the i^{th} line, only, so that the i^{th} line contains the view r_i has of the network, and the j^{th} column presents the vision the network has of r_j. Since each of these lines will be a separate shared variable, each has an associated time-stamp and sequence number (Figure 4.4).

From the topological information contained in the connectivity matrix $M^n_{N\times N}$, robot r_j is considered to be active by robot r_i if there is a bidirectional link between them, i.e. $r_j \in \mathcal{T}_A$ iff $M^n(i, j) = M^n(j,i) \neq 0$. Moreover, considering a robot active is transitive, i.e., if robot r_i considers r_o active, and robot r_o considers r_j active, then robot r_i also considers robot r_j active. When r_i does not consider r_j active directly (with a direct link) or indirectly (through transitivity) then, robot r_j is considered absent from the network and is not in $\mathcal{T}_A$.

Naturally, this mechanism requires all connectivity matrices of connected robots to be consistent, which is enforced within a bounded interval (see Sect. 4.4) by the updating rules shown previously.



	0	1	2	3	4	5
0	0	1	1	1	1	0
1	1	0	0	0	0	0
2	1	0	0	0	0	1
3	1	0	0	0	1	1
4	1	0	0	1	0	0
5	0	0	1	1	0	0

Figure 4.4: Control variables to update the connectivity matrix – Top (from left to right) vector containing the local age of each line of the matrix, vector containing the sequence number of each line of the matrix, the connectivity matrix, topology of the network

This interval sets a limit on the rate of topology changes that our protocol is capable of handling. Faster rates may prevent the team to reach consistent connectivity matrices, and therefore reach the same slot allocation table, i.e. reach a consensus. In order to simplify consistency when updating the table we use the same strategy as in RA-TDMA, based on a unique identifier (ID) per robot. Whenever the table is changed we sort the list of active robots by growing identifiers and assign them to the N slots in order, starting from slot 0.

Figure 4.5 depicts a situation in which $\mathcal{T}_A = \{r_0, r_4\}$ and robot r_3 becomes active, changing the active team to $\mathcal{T}_A = \{r_0, r_3, r_4\}$. Due to our slot allocation mechanism, initially, robot r_0 owns slot s_0, and robot r_4 owns slot s_1, however, as robot r_4 becomes active, the new slot allocation will change so that robot r_0 owns slot s_0, robot r_3 owns slot s_1, and robot r_4 owns slot s_2. In our example, robot r_3 is only connected to robot r_0, consequently, each robot has an incoherent perception of the team composition. Only when robot r_0 transmits its matrix to robot r_4, the knowledge of a new active robot is propagated, r_3 in this case, which, in turn, allows robot r_4 to build a consistent table. Since the team is dynamic, in order to refer to an active team member independently of its ID, we say that a robot has a logical ID n, i.e. robot l_n, iff that robot owns slot s_n. For example, in the right side of Figure 4.5, $\mathcal{T}_A = \{l_0, l_1, l_2\}$.

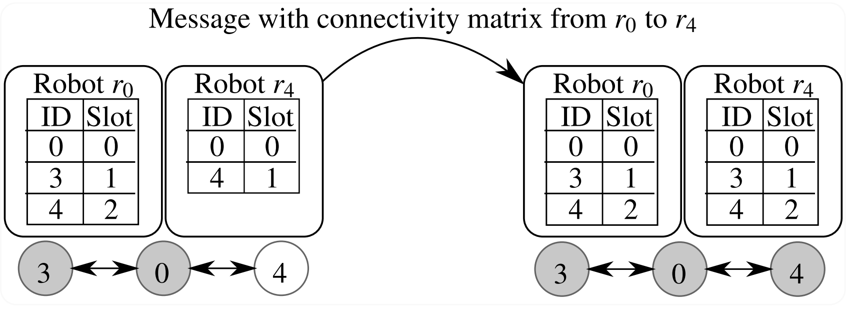


Figure 4.5: Updating the slot allocation table: Robot r_3 joins the network and tells robot r_0 (left), robot r_0 transmits the connectivity matrix to robot r_4 carrying the knowledge of a new team member (right).

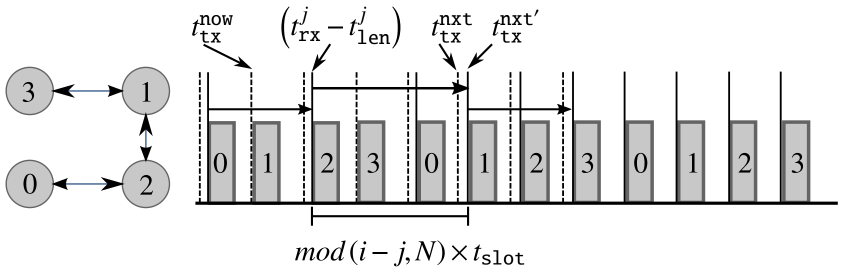

$t_{\mathtt{tx}}^{\mathtt{now}}$
$\left(t_{\mathtt{rx}}^{j} - t_{\mathtt{len}}^{j}\right)$
$t_{\mathtt{tx}}^{\mathtt{nxt}}$
$t_{\mathtt{tx}}^{\mathtt{nxt}'}$
3
1
0
2
0 1 2 3 0 1 2 3 0 1 2 3
$mod(i-j,N) \times t_{\mathtt{slot}}$

Figure 4.6: Propagating slots synchronisation – In the example robots need to propagate the synchronisation through their neighbourhood to the global network. The marked time points refer to the variables required for algorithm 1 when robot l_1 ($i = 1$) receives the message from robot l_2 ($j = 2$).

4.2.4 Synchronisation in ad-hoc networks

Beyond the consistency of the slot allocation table, the robots must also agree on the start of their slots. This particular aspect is also handled similarly to RA-TDMA but in a localised fashion in which each robot synchronises in each round with the robots in range, only, using the packets received from them. This synchronisation propagates to the whole network through any connection path, in a flooding fashion. In the beginning of each slot, each node sets the start of the next slot as one round later ($t_{\mathtt{tx}}^{\mathtt{nxt}} = t_{\mathtt{tx}}^{\mathtt{now}} + T_{\mathtt{up}}$). Then, when robot l_i receives a packet from robot l_j at $t_{\mathtt{rx}}^{j}$ and transmission time $t_{\mathtt{len}}^{j}$, where $t_{\mathtt{len}}^{j}$ is the time required to transmit b bits at B Mbps, algorithm 1 is executed to possibly adjust the start of the next slot. This causes a phase shift of the whole TDMA round.

Algorithm 1 Re-synchronisation of robot l_i upon reception of packet from robot l_j

```
$t_{\mathtt{tx}}^{\mathtt{nxt}'} = \left(t_{\mathtt{rx}}^{j} - t_{\mathtt{len}}^{j}\right) + mod(i-j,N) \times t_{\mathtt{slot}}$
$t_{\mathtt{tx}}^{\mathtt{nxt}} = max(t_{\mathtt{tx}}^{\mathtt{nxt}'}, t_{\mathtt{tx}}^{\mathtt{nxt}})$
```

Figure 4.6 shows the synchronisation mechanism where the initial slots are marked with dashed lines. A delay in robot l_0 is noticed by robot l_2 that delays its next slot setting a new time-frame, marked with full lines. Robots l_1 and l_3 are still unaware of this delay and keep their initial slots. Once robot l_2 transmits in the adjusted slot, robot l_1 is made aware of this adjustment and will synchronise. Finally, robot l_3 will also synchronise after receiving a packet from robot l_1.

Figure 4.7 shows the complete sending-receiving procedures of our ad-hoc broadcast and synchronisation protocol. In each round each node will receive at most once from each of its neighbours, aggregate all received variables with its own, possibly resynchronise the round time-frame, and, when the time comes, each node updates its own variables and transmits them.

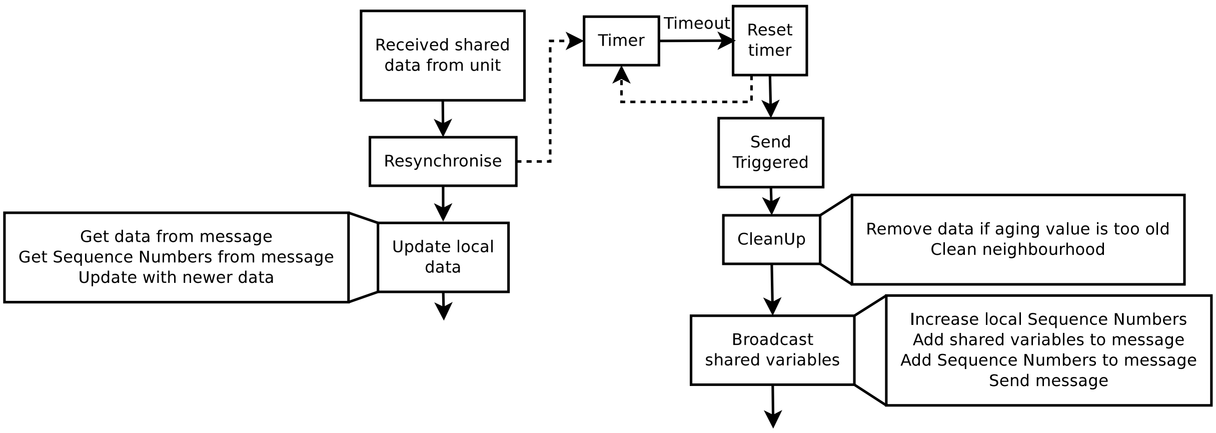

Received shared data from unit
Resynchronise
Update local data
Get data from message
Get Sequence Numbers from message
Update with newer data
Timer
Timeout
Reset timer
Send Triggered
CleanUp
Remove data if aging value is too old
Clean neighbourhood
Broadcast shared variables
Increase local Sequence Numbers
Add shared variables to message
Add Sequence Numbers to message
Send message

Figure 4.7: Complete broadcast protocol - sending and receiving procedures

4.3 Convergence of the Synchronisation Algorithm

In the previous section, we have shown an approach to synchronise packet exchanges between a team of mobile robots in wireless ad-hoc networks. Here we look into the convergence properties of the algorithm, considering different topologies and different starting conditions in order to test the robots ability to synchronise packet exchanges. Curiously, we found that under certain starting conditions and constraints, robots were guaranteed to converge to a synchronised state, however, given other starting conditions, convergence was not guaranteed. Thus we now analyse the ability of our algorithm to reach synchronisation, specifically, we analyse the starting conditions that allow our algorithm to converge to a solution, and we show some situations where our approach is not robust enough to synchronise communications. Finally, we conjecture that using a specific heuristic solves those issues. The proposed technique, includes limiting synchronisation to a subset of neighbouring robots. Namely, we propose to use the connectivity matrix to locally build a spanning-tree that, by removing loops within the synchronisation process, allows robots to synchronise their rounds.

4.3.1 Revisiting the problem

For the sake of clarity, we will now restate our synchronisation consensus problem. In particular, due to the round cyclic nature, we adopt a phase (θ) to represent the alignment of the rounds, which can have any value in the unit circle that corresponds to the period T_{up} (Eq. 4.3 and Figure 4.8). This representation, as it will be shown, abstracts the absolute rounds in time and helps us defining some variables that we will use to prove the convergence of our solution.

$$\theta(t) = \frac{2\pi}{T_{\text{up}}} t \tag{4.3}$$

For simplicity we consider that angles vary within the interval $[-\pi, \pi]$, except where explicitly stated otherwise. To accomplish this, given an angle $\bar{\gamma} \in \mathbb{R}$ we normalise the angle to $\Gamma(\bar{\gamma}) \in [-\pi, \pi]$ as in Eq.

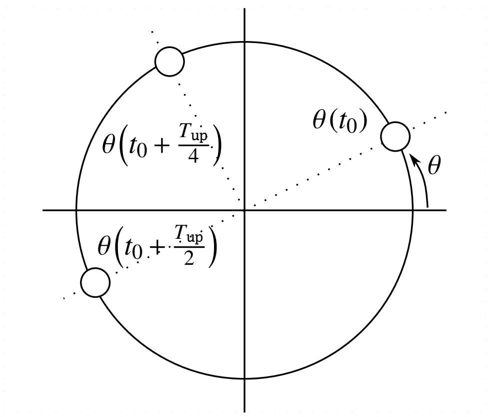

$\theta\left(t_0 + \frac{T_{up}}{4}\right)$
$\theta(t_0)$
θ
$\theta\left(t_0 + \frac{T_{up}}{2}\right)$

Figure 4.8: Time represented as a phase – As time progresses the value of θ increases (i.e. moves anti-clockwise) until it wraps around after one TDMA cycle.

4.4, where the operator $\lfloor\rceil$ represents the round function. In order to avoid cluttering the formulas we omit this step.

$$\Gamma(\bar{\gamma}) = \bar{\gamma} - 2\pi \left\lfloor \frac{\bar{\gamma}}{2\pi} \right\rceil \tag{4.4}$$

Using the definition in Eq. 4.3, we can write the rounds starting phase, called round phase – ϕ, as in Eq. 4.5.

$$\phi = \theta(t_{r_k}) \tag{4.5}$$

Replacing t_{r_k} from Eq. 4.1 into Eq. 4.5, we get Eq. 4.6[1].

$$\begin{aligned} \phi = \theta(t_{r_k}) &= \frac{2\pi}{T_{\text{up}}} \times t_{r_k} \\ &= \boxed{\frac{2\pi}{T_{\text{up}}} \times t_\phi} + \boxed{\frac{2\pi}{T_{\text{up}}} \times k \times T_{\text{up}}} \\ &= \boxed{\theta\left(t_\phi\right)} + \boxed{2\pi k} \end{aligned} \tag{4.6}$$

Note that after normalisation, $\forall_k : \phi = \theta(t_{r_k}) = \theta\left(t_\phi\right)$, as shown in Eq. 4.7.

$$\begin{aligned} \phi &= \Gamma\left(\theta(t_{r_k})\right) \\ &= \Gamma\left(\theta\left(t_\phi\right)\right) + \underbrace{\Gamma(2\pi k)}_{0} \\ &= \theta\left(t_\phi\right) \end{aligned} \tag{4.7}$$

---

[1] Boxes are used simply to highlight different terms.

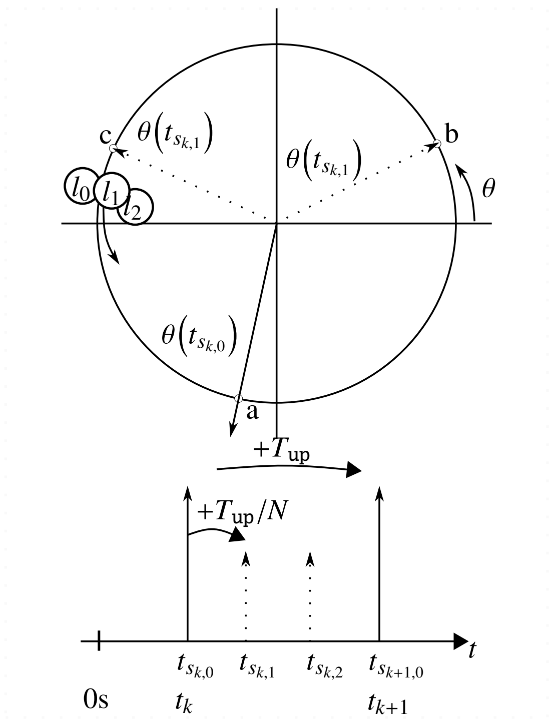


Figure 4.9: A TDMA cycle represented as a phase – (bottom) The previously presented TDMA cycle in time. (top) The equivalent representation in terms of phase. As the time of each robot progresses, represented by the circles (l_0,l_1,l_2), their phase moves around the circle. As each circle passes the slot it owns – namely point a, b, and c respectively – the robot starts its slot.

Similarly, the phase at which the slot n starts (σ_n) can be calculated solving Eq. 4.3 for $t = t_{s_{k,n}}$ (Eq. 4.2), obtaining Eq. 4.8.

$$\begin{aligned}\sigma_n &= \theta\left(t_{s_{k,n}}\right) = \frac{2\pi}{T_{\text{up}}} \times t_{s_{k,n}} \\ &= \boxed{\frac{2\pi}{T_{\text{up}}} \times t_{r_k}} + \boxed{\frac{2\pi}{T_{\text{up}}} \times n \times \frac{T_{\text{up}}}{N}} \\ &= \boxed{\phi} + \boxed{n \times \frac{2\pi}{N}}\end{aligned} \tag{4.8}$$

Figure 4.9 depicts both the time axis (bottom) representing the previously presented TDMA cycle, and the corresponding new approach (top). As the time of each robot progresses, represented by the circles (l_0,l_1,l_2), their phase moves around the circle. As each circle passes the slot it owns – namely point a, b, and c respectively – the robot starts its slot.

Let ϕ_i be the round phase of robot l_i, where robot l_i is the owner of slot i. Also, let Φ be the set of round phases of all active robots within the team, i.e. $\Phi = \{\phi_0, \phi_1, \ldots, \phi_{N-1}\}$. Similarly, let $\mathcal{N}_i$ be the neighbourhood of robot i and Φ_i the round phases of all robots that are neighbours of robot l_i including its own phase, i.e. $\Phi_i = \left\{\phi_i \cup \left\{\phi_k : M^i(i,k) \neq 0 \wedge M^i(k,i) \neq 0\right\}\right\}$ where M^i is the connectivity matrix in robot l_i.

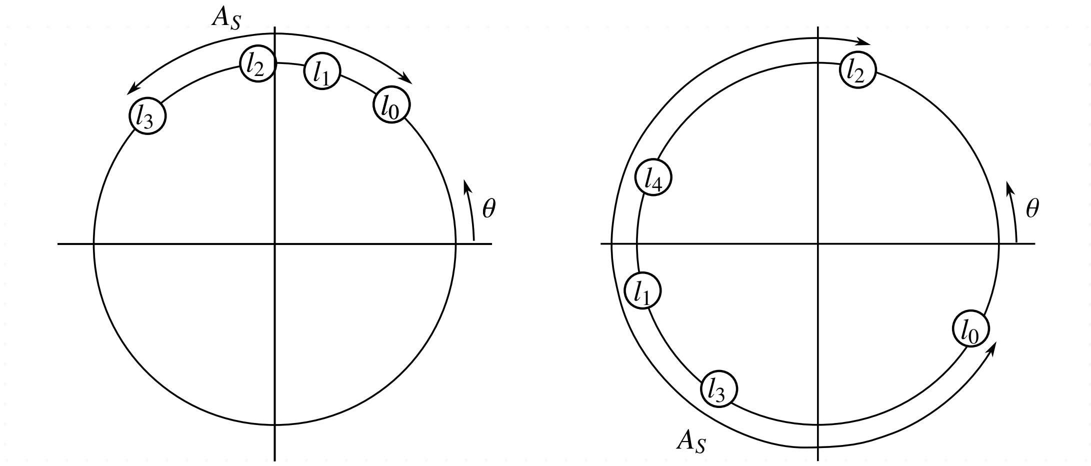


Figure 4.10: The Arc of a set of round phases – In this picture we show an example of the Arc of two sets of round phases. On the left a network which is almost synchronised, i.e. with a small Arc; on the right a network very unsynchronised, i.e. with a large Arc

Let S be a set of round phases with size $|S|$. Similarly to the work in [87], we define the Arc(S)[2], denoted A_S, as the shortest phase interval that contains all the phases of set S, see Figure 4.10. The complementary of this arc is the pair of consecutive phases that are further apart from each other, $A^*_S = 2\pi - A_S$. In order to measure A^*_S, we first need to sort the set $S = \{\phi_0, \phi_1, \ldots, \phi_{|S|-2}, \phi_{|S|-1}\}$ with growing phase, thus obtaining $S' = \left\{\phi'_0, \phi'_1, \ldots, \phi'_{|S|-2}, \phi'_{|S|-1}\right\}$, where $\phi'_0 < \phi'_1 < \cdots < \phi'_{|S|-2} < \phi'_{|S|-1}$. Using this ordered set, we can calculate the clockwise distances[3] between every pair of consecutive sorted phases $D_{S'} = \left\{d'_{0,1}, d'_{1,2}, \ldots, d'_{|S|-2,|S|-1}, d'_{|S|-1,0}\right\}$. Consequently, the distance, $d'_{i,j}$ normalised in the interval of $[0, 2\pi]$, between the phases of robot l_i and l_j can be calculated as in Eq. 4.9.

$$d'_{i,j} = \left(\phi'_j - \phi'_i\right) - 2\pi \left\lfloor \frac{\phi'_j - \phi'_i}{2\pi} \right\rfloor \tag{4.9}$$

Finally, we can obtain A^*_S as in Eq. 4.10 and, using this result, solve Eq. 4.11 and calculate A_S.

$$A^*_S = max(D_{S'}) \tag{4.10}$$

$$\begin{aligned} A_S &= 2\pi - A^*_S \\ &= 2\pi - max(D_{S'}) \end{aligned} \tag{4.11}$$

Using the A_S, we can measure the state of synchronisation of the network, since the smaller the value of the Arc, the smaller is the difference between the phases of the robots. An example is depicted in Figure 4.10.

---

[2] The authors call it Diameter, but we found this name not representative of what it is.

[3] We define the clockwise distance between two phases as the angle separating them in a clockwise direction.

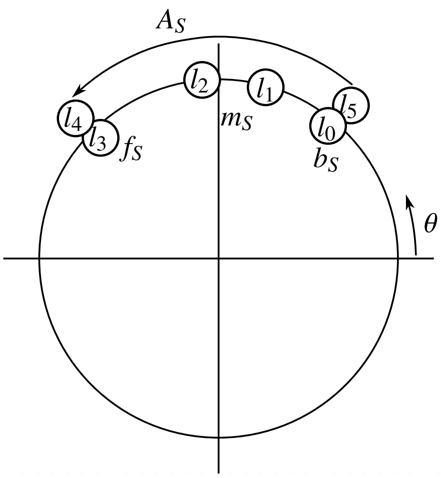


Figure 4.11: Classification of the robots within the Arc according to their phases – Robots can either be $\mathbf{f}_S$, $\mathbf{b}_S$, or $\mathbf{m}_S$, if they are in the front, back, or middle of the Arc respectively.

Within A_S, a robot can be in one of three relative positions in terms of its round phase (ϕ), as depicted in Figure 4.11:

- $\mathbf{f}_S$ – A robot that has the most advanced round phase within the arc A_S
- $\mathbf{b}_S$ – A robot that has the earliest round phase within the arc A_S
- $\mathbf{m}_S$ – A robot that is neither a $\mathbf{b}_S$ nor a $\mathbf{f}_S$

Note that, robots can only be classified in one of the categories above, therefore, they will first be tentatively classified as $\mathbf{f}_S$ robots, then as $\mathbf{b}_S$ robots, and only then as $\mathbf{m}_S$ robots. Consequently, if all have the same round phase, all robots will be $\mathbf{f}_S$ robots.

4.3.2 Bounding the maximum delay induced by synchronisation

The solution presented so far does not impose a limit on the delay between two consecutive communications, thus potentially allowing robots to transmit inside the slot of another robot. In order to avoid such a situations, we now introduce a limitation on the delay each robot could induce in one of its TDMA rounds when adjusting its phase. This limitation forces some modifications in the way we express the synchronisation method in the previous section.

When a robot l_j reaches its slot, and sends a packet ($\mathtt{P}_j$) to its neighbourhood, those neighbours need to calculate its round phase (ϕ_j). The measurement of the round phase of robot l_j by its neighbour l_i, starts when the latter receives the packet from the former at time $t^j_{\mathtt{rx}}$. Using this information, robot l_i can calculate robot l_j slot start ($t_{s_{k,j}}$) as in Eq. 4.12, where $t^j_{\mathtt{len}}$ is the packet transmission time, i.e. the time required to transmit b bits at B Mbps. Using this value in Eq. 4.8, we can calculate the phase of slot j in Eq. 4.13 and, from this, the phase (ϕ_j) of the round of robot l_j in Eq. 4.14.

$$t_{s_{k,j}} = t^j_{\mathtt{rx}} - t^j_{\mathtt{len}} \tag{4.12}$$

$$\sigma_j = \theta\left(t_{s_{k,j}}\right) = \phi_j + j \times \frac{2\pi}{N} \tag{4.13}$$

$$\phi_j = \sigma_j - j \times \frac{2\pi}{N} \tag{4.14}$$

When robot l_i reaches its own slot, it will have collected all the round phases of all of its neighbours $\mathcal{N}_i$ and solved the max-consensus problem as written in Eq. 4.15. Note that since $\phi_i - \phi_i = 0$, δ_i is never negative.

$$\delta_i = \max_{k \in \mathcal{N}_i \cup \phi_i} (\phi_k - \phi_i) \times \frac{T_{\text{up}}}{2\pi} \tag{4.15}$$

Lastly, we define a delay saturation value Δ, that can be set according to the application requirements, and set the new round phase (ϕ_i^{nxt}) for robot l_i based on its current phase ϕ_i^{now} Eq. 4.16, effectively limiting the amount of delay a robot can induce in its own communication, i.e., bounding the maximum phase adjustment per round.

$$\phi_i^{\texttt{nxt}} = \phi_i^{\texttt{now}} + \min(\Delta, \delta_i) \times \frac{2\pi}{T_{\text{up}}} \tag{4.16}$$

4.3.3 Proving the convergence

From the previous sections, it is clear that if all robots agree on the value of ϕ, i.e., $\phi_i = \phi \; \forall_{i=0..N-1}$, then the communication rounds are synchronised, or equivalently, all robots l_i in the team ($i = 0..N-1$) are $\mathbf{f}_\Phi$ robots. Therefore, in order to prove that communications synchronise, we need to show that using our protocol the number of $\mathbf{f}_\Phi$ robots tends to increase for all i, and never decreases.

For this purpose, lets first consider that all round phases start within an interval of π, i.e. $A_\Phi < \pi$. In this situation, as we depict in Figure 4.12, independently of the observation reference, as long as this reference is within the arc A_Φ, the robot with the maximum phase is the same for all robots. In addition, lets consider a period where external interference does not occur, therefore the only source of delay is an initial set of delays then modified by the protocol operation. The proof of convergence that follows is then subject to these starting conditions.

First, consider an arbitrary robot l_i in the team and let $|F|$ be the number of $\mathbf{f}_\Phi$ robots, $|M|$ be the number of $\mathbf{m}_\Phi$ robots, and $|B|$ be the number of $\mathbf{b}_\Phi$ robots. Since $A_\Phi < \pi$, any $\mathbf{f}_\Phi$ robots cannot detect a round phase that is more advanced than their own. If this is the case for robot l_i, it will not delay its own communication for round phase adjustment. Consequently, both $|F|$ and A_Φ will remain unchanged. If, on the other hand, robot l_i is a neighbour of a $\mathbf{f}_\Phi$, it will observe a phase more advanced than its own. Therefore, it will delay its own communications, increasing its phase according to Eq. 4.16, and reducing the difference to the round phases of the $\mathbf{f}_\Phi$ robots until that difference is zero. At that point, it becomes a $\mathbf{f}_\Phi$ robot, and $|F|$ increases. This reasoning can then be repeated with the new $\mathbf{f}_\Phi$ robots and their neighbours, until $|M| = 0$ and $|B| = 0$. Note that, as all $\mathbf{b}_\Phi$ robots delay their communications, the value of A_Φ decreases, reaching zero when all robots become $\mathbf{f}_\Phi$ robots, and $|F| = |\Phi|$. This trend occurs in all robots l_i, with $i = 0..N-1$, eventually leading to the synchronisation of the whole team.

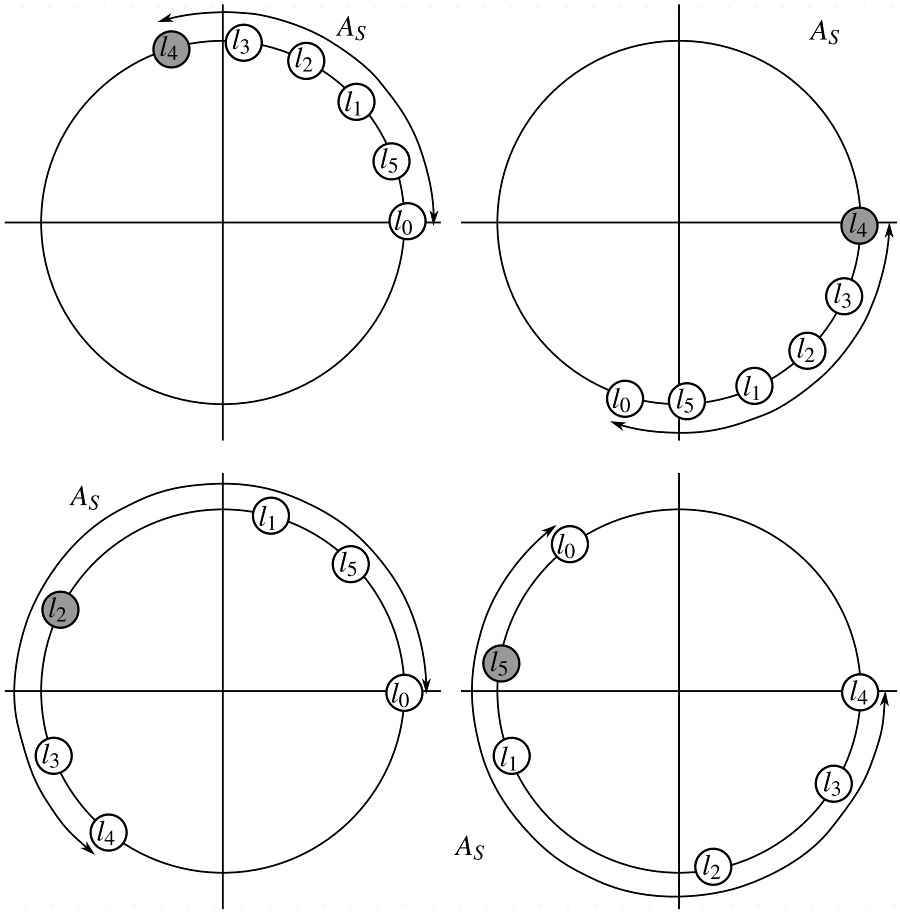

A_S
l_4
l_3
l_2
l_1
l_5
l_0
A_S
l_4
l_3
l_2
l_1
l_5
l_0
A_S
l_1
l_5
l_2
l_0
l_3
l_4
l_0
l_5
l_4
l_1
l_3
A_S
l_2

Figure 4.12: Dependency on the point of observation – If all robots are within an Arc of π (top), then from the point of view of any of them, the most advanced phase is the same (dark grey circle); conversely, if the Arc is larger than π (bottom), the same does not apply.

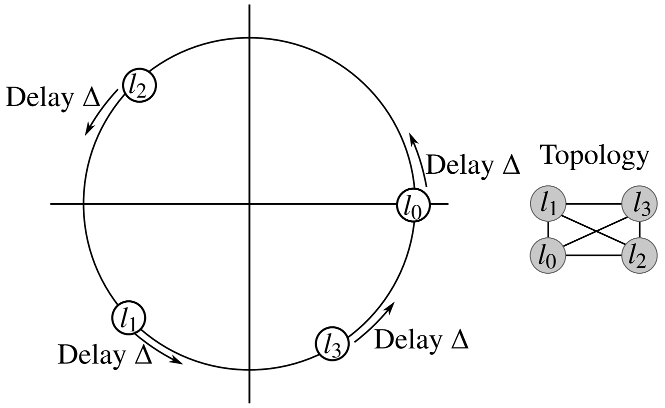


Figure 4.13: First example of a failure to synchronise – When there is a maximum delay of Δ in each round, and all robots measure a delay above that maximum value, all robots delay their own transmission by the same amount. Therefore, the relative distances of the phases is kept constant, and synchronisation is never reached.

4.3.4 Dropping the $A_\Phi < \pi$ assumption

Through extensive testing we realised that in particular situations when $A_\Phi >= \pi$ then synchronisation was not reached. An example is shown in Figure 4.13, in which all robots detect delays that are superior to the maximum allowed delay correction (Δ) per round, therefore, they all delay their communications by the same amount of time. The resulting behaviour is that their relative phases do not change, thus the robots remain unsynchronised transmitting with a period equal to $T_{\text{up}} + \Delta$, and only clock drifts or packet failures can introduce phase changes that may eventually bring them to a point in which synchronisation is again possible.

In addition to the previous case, we also found that some specific topologies, with loops as depicted in 4.14, where all nodes have a neighbour that has a larger round phase than their own, can also lead to synchronisation failure. In those situations, each robot l_i will try to synchronise with robot l_{i+1} and robot l_{N-1} will try to synchronise with robot l_0. Consequently, all the robots will delay their transmission in the round which, similarly to the previous problem, inhibits their ability to synchronise their rounds.

In order to reach synchronisation even in such cases, we modified our protocol to avoid loops that hinder the ability to reach a consensus. Our solution, is to use the topology information contained in the connectivity matrix and build a tree using the available connections, Figure 4.15. Since the matrix is available in all robots, each one of the team members can build its own tree locally following a common set of rules. First of all, the root of the tree is always set as robot l_0, then we add all neighbours of robot l_0 to the tree, and repeat the process for the neighbours of the neighbours until all the robots are added. The actual order in which they are added is irrelevant, as long as it can be described as a rule that can be replicated locally in each robot. Finally, note that the tree does not limit the propagation of information; it is only used to select a subset of neighbours with which the robot synchronises, instead of synchronising with all.

Nevertheless, forcing a tree reduces the speed of dissemination of the synchronisation information within the network, and thus reaching all robots, thus increasing the amount of time required to syn-

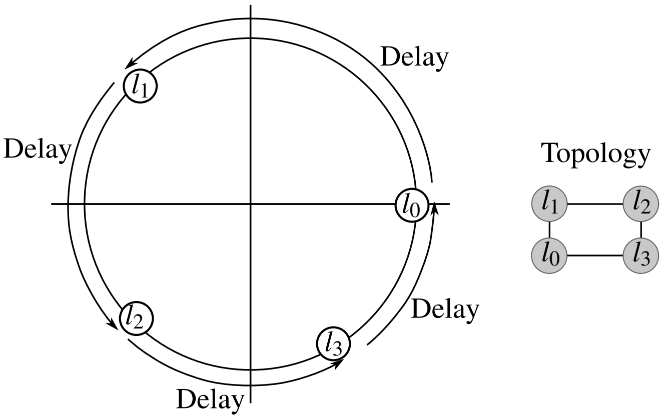

Delay
Delay
Delay
Delay
l_1
l_0
l_2
l_3
Topology
l_1
l_2
l_0
l_3

Figure 4.14: Second example of a failure to synchronise – Even when there is no limit to the amount of delay per round, some cases are not synchronised using this approach. In this example, each robot l_0, l_1, l_2, and l_3 will changes its initial phase to the value of robots l_1, l_2, l_3, and l_0 respectively.

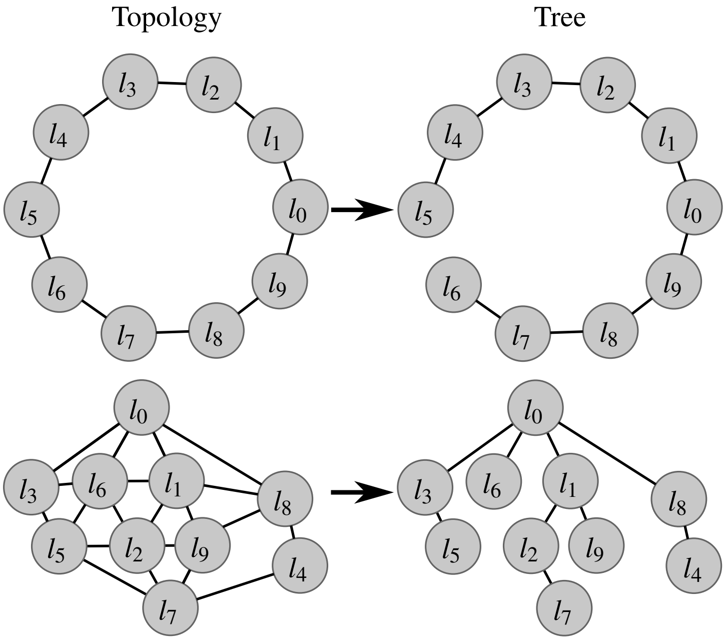

Topology
Tree
l_0
l_1
l_2
l_3
l_4
l_5
l_6
l_7
l_8
l_9

Figure 4.15: Building a tree from the topology information – Using the topology information from the connectivity matrix we can build a tree. Then, using this tree, we limit the neighbours with which robots can synchronise their communications.

chronise communications. If robots were able to measure the value of A_Φ directly, then it would be simply a matter of using the tree only when $A_\Phi >= \pi$. However, since robots can only observe their neighbourhood, they have a limited vision of the network, and robot l_i can only measure A_{Φ_i}. Therefore, as a safer upper bound, we share the Arc measured by each robot and calculate $\Sigma_i = \sum_{n \in [0,N-1]} A_{\Phi_n} \geqslant A_\Phi$. Specifically, robot l_i will use the tree if $\Sigma_i \geqslant \pi$.

As an implementation detail, for the sake of stability of the synchronisation process, an hysteresis was added to this mode change. This is, only after a pre-determined number of TDMA rounds where the robot observes $\Sigma_i \geqslant \pi$ will it start using the tree. Similarly, only after the same number of rounds where $\Sigma_i < \pi$, will the robot stop using the tree.

Finally, we could not prove that using the tree method ensures synchronisation with all possible phase scenarios, yet. However, we conjecture that it does, since we ran extensive simulations with random phases and synchronisation was always achieved.

4.4 Upper Bounds to Information Propagation Latency

Despite the unreliability of the wireless medium, it is reasonable to consider the medium lossless for the purpose of establishing some baseline properties that are intrinsic to the protocol. Here we analyse the conditions that maximise the information propagation latency in the absence of packet losses.

First we analyse the worst case topology for information sharing. This situation is similar to the one reported in [35] and corresponds to the case in which all robots form a line but sorted such that the identifiers decrease in the direction of the propagation of the information. For example, Figure 4.16 shows the worst-case topology for propagating information from robot 4 to robot 0. In this case we will need one initial round for robot 4 to transmit its new information, such as a joining node, which will be received by robot 3 that will transmit it to robot 2 in the following round until the information gets to robot 1. At that point, one slot is enough to finally transfer the information to robot 0. The total worst-case latency is $(N-2) \times T_{\mathtt{up}} + t_{\mathtt{slot}}$. If any two robots switch position, or if there are any parallel paths, the latency will be lower.

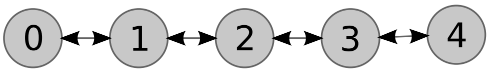


Figure 4.16: Worst-case data propagation scenario

Finally, we compute the maximum age that a given information might develop in the network. Let us consider $max_{\mathtt{val}} = t_{\mathtt{val}}/T_{\mathtt{up}}$ as the number of rounds in the information validity interval. Suppose now that a given information is $max_{\mathtt{val}}$ old when transmitted by robot n in a worst-case line topology. This information will arrive at robot 0 in less than $N-1$ rounds, thus within $max_{\mathtt{val}} + (N-1)$ rounds after its generation. Then, robot 0 will keep it for another $max_{\mathtt{val}}$ rounds before removing it. Therefore, the maximum time interval that a piece of stale information can remain in the network before being removed is upper bounded by $2 \times max_{\mathtt{val}} + (N-1)$.

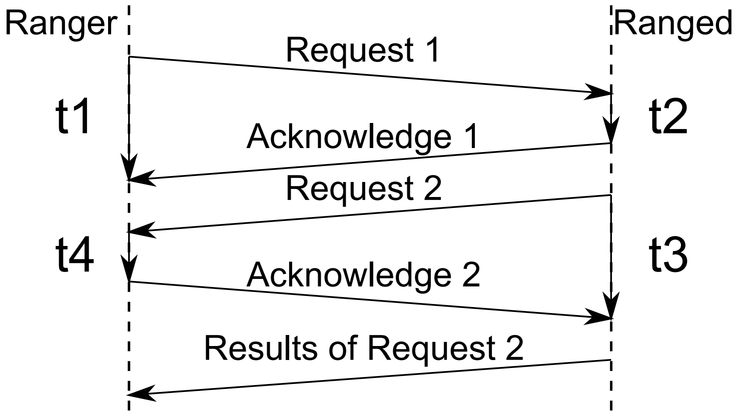


Figure 4.17: Illustration of the ToF ranging process – The robot on the left ranges the robot on the right.

4.5 Experimental Validation of the Synchronisation Protocol

In this section we show an experimental validation carried out with a Nanotron's nanoLoc development kit [88]. This kit includes 5 nodes, each using an Atmega128l μC, communicating in the 2.4GHz ISM band according to IEEE 802.15.4 with a chirp modulation, which allows RF ranging using Round-Trip Time Of Flight (RT-ToF). This ranging method works by measuring the time a packet needs to reach the destination and return, without requiring global clock synchronisation. As implemented by the nanoLoc development kit [88], the ranging is done in two phases for increased accuracy (Figure 4.17). The first phase measures $r_1 = V \times (t_1 - t_2)/2$ and the second one measures $r_2 = V \times (t_3 - t_4)/2$, where V is the propagation speed of the RF signal. Finally, r_2 is sent back and the values are averaged, thus the whole ranging procedure returns $\overline{d} = (r_1 + r_2)/2$.

We organise these experiments in two sections, firstly showing the synchronisation capabilities of this algorithm in a small indoors environment and secondly, exhibiting the improvements of the ranging process due to using the synchronisation protocol. In all cases, we define $max_{val} = 10$ rounds.

4.5.1 Validating the synchronisation protocol

We start by setting $T_{\text{up}} = 500ms$ and activating robots 1, 3 and 4 which run the protocol. Robot 0 is used for monitoring purposes, only. As shown in Figure 4.18, there are two disjoint subnetworks, one with robot 1 and the other with robots 3 and 4. Note that the protocol allows each subnetwork to synchronise internally independently of each other (Figure 4.18 left plot, up to round 26). Then, at that point, robot 2 is switched on and connects to both subnetworks, joining them, thus allowing the synchronisation to propagate across. After a short transient of 2 rounds, all robots are synchronised with their transmissions separated as much as possible (125ms).

Figure 4.19 shows a case with 5 robots and $T_{\text{up}} = 200ms$, where several consecutive network reconfigurations occur, with nodes being switched on and off thus joining and leaving. In the beginning, robots 0, 2 and 4 form a network and robot 3 joins at round 24 causing a resynchronisation from 3 to 4 slots. At round 45 robot 1 also joins. Then, at round 60, robot 4 leaves, which causes a resynchronisation later on,

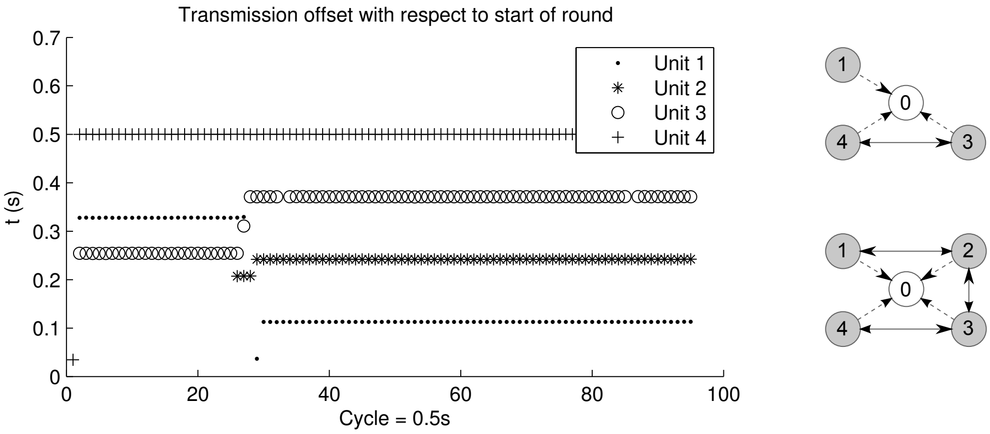


Figure 4.18: Synchronising disjoint networks with one robot – Initially, robots 1, 3 and 4 are activated but only the latter two are connected. Then, in round 26, robot 2 is switched on and connects to both robots 1 and 3, joining all in one single network. This allows the synchronisation to propagate across all robots. Note: robot 0 is used for monitoring purposes, only.

after $max_{val} + 1 = 11$ rounds, which is when it is removed by the nodes it was connected to. The same happens when robot 2 and later robot 3 leaves the network. All protocol timings were verified.

4.5.2 Impact of the synchronisation on the application

In the following experiments, we set up a fully linked network with 5 nodes but using two different physical layouts (topologies) aiming at analysing the impact the synchronisation has on the ranging performance, both in terms of accuracy and of failure rate.

Concerning accuracy, we used the physical layout of Figure 4.20a with a separation of $1m$ between every two consecutive robots. Robot 0 ranged every other robot and logged the results. We analysed 3500 rounds of operation with, and another 3500 without, synchronisation, with $T_{\mathrm{up}} = 200ms$, and calculated the average measurement error when synchronised ($\overline{d_e^S}$) and when unsynchronised ($\overline{d_e^U}$), and the respective standard deviations ($\overline{\sigma_e^S}$ and $\overline{\sigma_e^U}$). The ranging results showed that the accuracy was similar in both cases with a negligible difference on the average errors, $\left|\overline{d_e^S} - \overline{d_e^U}\right| < 0.01m$. The difference in terms of standard deviation of the distance measurements was also small, $\left|\overline{\sigma_e^S} - \overline{\sigma_e^U}\right| < 0.1m$. This is expected since the synchronisation does not affect the ranging that actually takes place. Thus, the obtained distances are similar in both situations.

Concerning ranging failure rates, we used the layout shown in Figure 4.20b but we used robots 1 to 4 only. Two runs of 5000 consecutive rounds were logged using a $t_{up} = 200ms$ and for both cases, with and without synchronisation. The percentage of failed ranges for each robot is shown in Table 4.1a. Then we repeated these experiments after having switched on robot 0, which was programmed to send a 127B packet every $20ms$ ($64.3kbps$), without synchronisation, just to create interference. The percentages of failed ranges for each robot, are shown in Table 4.1b.

The results of the synchronised experiments show a residual percentage of range failures that is similar for all robots, between 3% and 4.4%. The results with interference show a minor degradation,

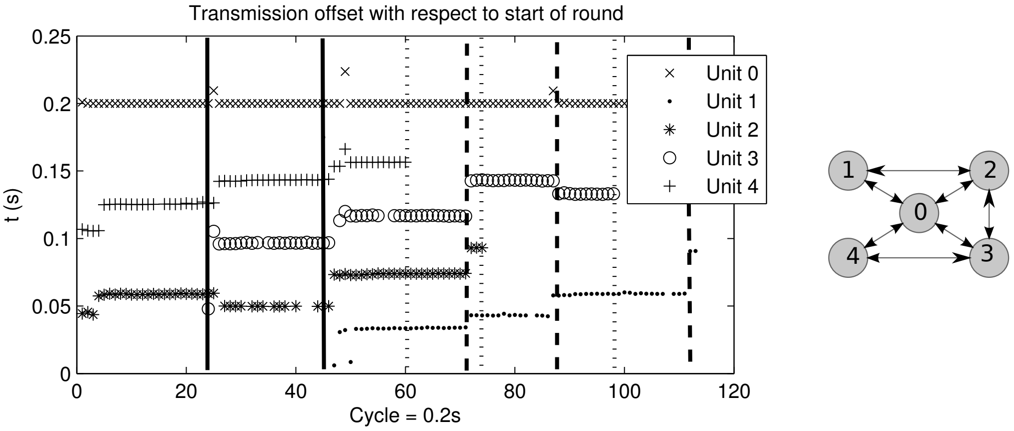
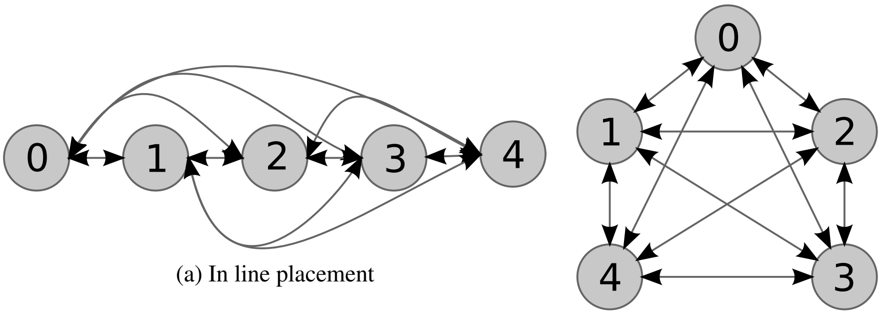


Figure 4.19: Synchronisation protocol operating with joining and leaving robots



(a) In line placement

(b) Arbitrary positions

Figure 4.20: Two fully linked physical layouts

Table 4.1: Experimental results using topology in Figure 4.20b

(a) Baseline measurements

$t_{up} = 200ms$
5000 samples
Interference: No

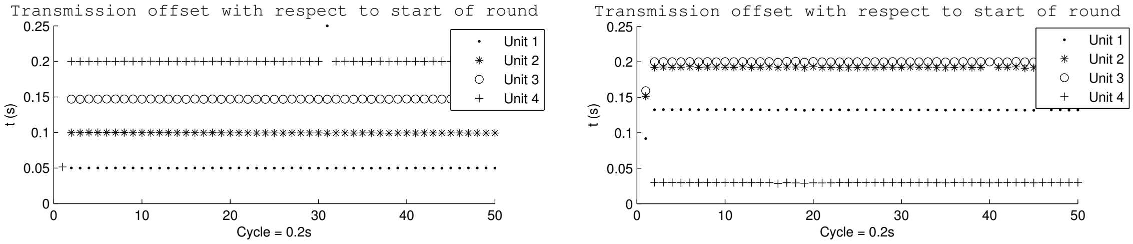
Synchronisation	Robot	Error Rates	
		Run 1	Run 2
Yes	1	3.82%	2.98%
	2	3.70%	3.53%
	3	4.36%	3.54%
	4	4.42%	3.52%
MEAN		3.73%	
STD		0.47%	
No	1	4.20%	3.92%
	2	13.64%	23.94%
	3	11.82%	24.54%
	4	3.62%	14.70%
MEAN		12.55%	
STD		8.48%	

(b) Measurements with noise

$t_{up} = 200ms$
5000 samples
Interference: $127Bytes/20ms$

Synchronisation	Robot	Error Rates	
		Run 1	Run 2
Yes	1	4.16%	3.92%
	2	4.32%	3.96%
	3	3.54%	3.50%
	4	4.46%	4.90%
MEAN		4.09%	
STD		0.47%	
No	1	4.36%	12.00%
	2	16.14%	4.88%
	3	10.82%	5.94%
	4	4.28%	13.02%
MEAN		8.93%	
STD		4.62%	

increasing the percentages of losses from 3.5% to 4.9%. This is expected because of the relatively weak interference that would affect the ranging only occasionally. On the other hand, the results without synchronisation show substantial degradation on certain nodes, in one case going up to 25% ranging failures. Nevertheless, even without synchronisation it is still possible to find robots exhibiting range failures similar to the synchronised case. This is easily explained looking at Figure 4.21. In fact, without synchronisation some robots will end up transmitting almost at the same time, which causes a critical interference period with high number of failures due to collisions, while other robots will transmit very far apart, thus similarly to the synchronised case. For example, in Figure 4.21b robot 1 has a very high clearance from the other robots while robots 2 and 3 are transmitting very close to each other. This log corresponds to the baseline Run 2 experiments without synchronisation in Table 4.1a. With synchronisation the team robots do not practically interfere with each other. Consequently, the average range failure rate without synchronisation and without interference is 3.3 times higher than that obtained with synchronisation (12.5% compared to 3.7%), and the standard deviation of the robots failures rates is one order of magnitude higher without than with synchronisation (8.48% and 4.62% compared to 0.47%). However, the actual degradation of the situation without synchronisation depends on too many factors, such as starting conditions, and is thus very difficult to characterise accurately.



(a) Synchronised (b) Non-Synchronised

Figure 4.21: Periodic dissemination with and without synchronisation

4.6 Conclusions

In this chapter we proposed a novel synchronisation / broadcast protocol for ad-hoc dynamic networks. The protocol is fully distributed and symmetrical in the sense that all nodes play a similar role. We proved, using a consensus approach, that under certain assumptions the network achieves synchronisation, allowing the nodes to transmit periodically in disjoint slots in a TDMA fashion that is globalised throughout the network. Then we conjectured that the restrictive assumptions made in the consensus proof could be dropped with a proposed heuristic. Despite absence of formal proof, the proposed heuristic always led to synchronisation in extensive simulations. Finally, the proposed protocol was experimentally validated in a ranging application using a RT-ToF method that integrates the dissemination of ranging data through the network in an effective way contributing to an improved relative localisation service. Experimental results with IEEE 802.15.4 nanoLOC nodes validate the properties of the protocol, namely its ability to enforce synchronisation in ad-hoc scenarios even when a single path connects different nodes, its ability to acquire and efficiently disseminate ranging information thorough the network, as well as its effectiveness in reducing the failure rates of the ranging operations.

Chapter 5

Multi-hop Communications Within a TDMA Schedule

The ability to send data to distant robots is very important to several common applications such as teleoperation of robots for inspection in difficult to access areas [42], and the streaming of multimedia links, particularly in search and rescue scenarios [37]. However, due to the limits of the wireless medium, often times the communication range is not enough for those applications. The solution is forwarding packets in a multi-hop fashion. There are many challenges when designing multi-hop routing algorithms for ad-hoc, on one hand the discovery and maintenance of routes, and on the other hand, the management of transmissions on the wireless medium caused by multiple concurrent users.

As for the first, it has been widely studied with a large variety of options [89]. Some of them search the routes proactively. For example, the Destination-Sequenced Distance-Vector (DSDV) routing protocol [90], one of the first designed for mobile ad-hoc networks and Babel [91]. The Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.) protocol [92] takes a slightly different approach, where destination nodes send packets across the network, and the sources receive them to decide the best next-hop based on how many packets were received by which neighbour. Others are reactive, and only search for the routes when needed. Such as the Ad hoc On-Demand Distance Vector (AODV) routing protocol [93]. Implementing both reactive and proactive components, the biologically inspired AntHocNet [94] algorithm behaves reactively until a new route is requested. Once a new route is requested, it sends "ants" that travel from the source to the destination node through multiple routes. When they arrive, the "ants" are sent back to the source node, where the different routes are compared according to a metric, and a route is selected. Then, once the route is established, "ants" are proactively transmitted in order to maintain (or possibly discover new) routes. Finally, the work in [95] presents a protocol that is based on the routing protocol found in [96], but focused on the route creation and selection. In this work, no routes are pre-established, the agent that produces the video broadcasts the frames, then the several receivers set a random wait time (proportional to the perceived link quality) to retransmit the frame to the next hop. This process, eventually creates a route, that is then used to send unicast packets.

In this work, we focus in the transmission management problem, namely we address the problem of congestion of the medium leading to increased collisions and substantial degradation of network performance [11]. We use the topology information that is already provided by the Reconfigurable and Adaptive TDMA (RA-TDMA) protocol presented in the previous chapter to generate minimum hop-count routes. In order to mitigate this phenomenon, many protocols, including standards such as the IEEE 802.11 and IEEE 802.15.4, already employ the well-known, Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). However, applied to ad-hoc networks, this technique may behave poorly in the presence of hidden nodes, and even the RTS/CTS fails, since the disruption range is larger than the reception range [23]. Despite that, several multi-hop routing protocol implementations, still rely on this approach to transmit data.

Many alternatives exist to mitigate interference amongst mobile agents by trying to avoid contention at the medium access. Some of the protocols rely on token passing [97, 98] and each node can only communicate when in the possession of the token. The token is used to carry out a global priority-based arbitration, thus supporting real-time guarantees. In general, this protocol offers low end-to-end delays with low communication load or for high priority packets. Conversely, large delays can occur when a token is lost or under high communication loads for lower priority traffic, and the token circulation for arbitration imposes a significant overhead.

As opposed to the token passing approach in [97, 98], we will use a Time Division Multiple Access (TDMA) protocol. TDMA protocols have less maintenance overhead, and are less prone to failure (e.g. the loss of a token). Nonetheless, the TDMA schedules impose large latencies, for which we propose a solution within this chapter.

5.1 Relevant Publications

The work presented in this chapter has been partially published in the following paper. An extension including the worst-case end-to-end latency analysis is under preparation for submission to a relevant journal.

- [16] L. Oliveira, L. Almeida, P. Lima, Multi-hop routing within tdma slots for teams of cooperating robots, in: IEEE World Conference. on Factory Communication Systems - WFCS, 2015.

5.2 TDMA in Mobile Robot Teams

Using TDMA schedules introduces long delays on communications [99], exacerbated when transmitting information to agents beyond the one-hop neighbourhood. Many authors assign links to slots instead of agents, i.e., a slot given for each packet transmission instead of for each node. For example the work of [99] the authors study the impact of link scheduling in the queuing delay, for example the slot order allocation. Also the work in [100] studies the impact of several link scheduling techniques (breadth-first vs depth-first) in WirelessHART [101]. Another example can be found in [102], where the authors interpret delay as a cost of transmission order of the link and formulate an optimisation to find a transmission schedule with the min-max delay across a set of multiple paths. However, when agents are mobile, links are created and destroyed many times making these approaches inadequate due to schedules being recalculated and redistributed frequently.

Therefore, and following the RA-TDMA protocol, we allocate a TDMA slot per agent, not per link. The Ad-hoc RA-TDMA protocol is a common temporal multiplexing scheme for periodic communications that provides each agent with an exclusive fixed duration transmission window (or slot). These slots are organised in a round that repeats continuously, consequently, with this scheme, medium access collisions are precluded. We build upon this protocol, which generates TDMA rounds without global clock synchronisation, but leaves open the issue of propagating information beyond the one-hop neighbourhood. In the beginning of a slot, our protocol (Ad-hoc RA-TDMA) sends a multicast packet to all one-hop neighbours of the slot owner. Since agents know when to expect this packet, in the event of a delay they also delay their own transmissions in order to maintain synchronisation. Other than its intended synchronisation purpose, this packet is also used to track and disseminate the network topology, which can be used to perform routing decisions. Once this synchronisation packet is sent, the remaining slot time is free. We use this remaining time for both local (with neighbours) and remote (multi-hop) data transmissions, considering the application multi-hop communication requirements. Albeit vital for the protocol, RA-TDMA is largely tolerant to sporadic losses of these synchronisation packets.

In fact, reducing delay on multi-hop communication networks is not new or exclusive to wireless ad-hoc networks. Recently, the work in [103] presented a solution for reducing delivery delays on multi-hop switched Ethernet networks that resembles our proposal for immediate forwarding. However, the mobility and half-duplex characteristics of wireless networks preclude the direct use of this work in our application context.

5.2.1 Team end-to-end communication requirements

In this work, we consider a set ($\mathcal{J}$) of P periodic unicast packets, where each packet is defined as in 5.1.

$$\mathcal{J} = \{p_i = (C_i, T_i, D_i, Sr_i, Ds_i), i = 1..P\} \tag{5.1}$$

In our model, C_i represents the total transmission time of packet p_i across one hop, T_i its generation/activation period, and D_i represents its relative deadline (in this work we consider constrained deadlines, i.e., $D_i \leq T_i$). In addition, Sr_i and Ds_i represent, respectively, the source and destination of packet p_i. Note that our protocol supports multi-packet packets but we leave fragmentation for an upper layer and consider single packets at this level, only.

5.2.2 Packet duration in IEEE 802.11 DCF

In this work we use the IEEE 802.11b/g Distributed Coordination Function (DCF) as the Media Access Control (MAC) layer protocol, nonetheless the work we present can be applied on top of many other technologies. For completeness, we start by reviewing the time required to transmit packets in an IEEE 802.b/g ad-hoc environment. The packet durations presented in this section are for IEEE 802.11b. Nevertheless, we also use them as a slightly pessimistic approximation for IEEE 802.11g networks.

In order to transmit a multicast packet, since there is no acknowledgement, the time needed for the complete transmission is simply the time required for transmitting a data frame (Eq. 5.2). The

Table 5.1: IEEE802.11b/g parameters

	IEEE 802.11b (only)	IEEE 802.11b/g (mixed)	IEEE 802.11g (only)
Parameter	Duration (μs)	Duration (μs)	Duration (μs)
$t_{\mathtt{DIFS}}$	50	50	28
$t_{\mathtt{SIFS}}$	10	10	10
$t_{\mathtt{backoff}}$	620	620	135
$t_{\mathtt{preamble}}$	192	192	26

frame, however, is composed of several terms, the Distributed Coordination Function Inter Frame Spacing (DIFS) which is used to assess if the medium is busy, a random backoff time in case the medium was busy when sensed, the preamble which signals the beginning of the frame, and the data transmission itself, see (Eq. 5.3). For simplicity, we consider that the data transmission bitrate is known and set to be constant.

$$t_{\mathtt{mcast}}(\mathtt{size},\mathtt{bitrate}) = t_{\mathtt{frame}}(\mathtt{size},\mathtt{bitrate}) \tag{5.2}$$

$$t_{\mathtt{frame}}(\mathtt{size},\mathtt{bitrate}) = t_{\mathtt{DIFS}} + t_{\mathtt{backoff}} + t_{\mathtt{preamble}} + \frac{\mathtt{size} \times 8}{\mathtt{bitrate}} \tag{5.3}$$

If instead the packet is unicast, then after the data frame, an acknowledgement is transmitted, (Eq. 5.4). The acknowledgement duration includes the Short Inter Frame Space (SIFS), the transmission of a preamble, and 14 Bytes of acknowledgement data (Eq. 5.5).

$$t_{\mathtt{ucast}}(\mathtt{size},\mathtt{bitrate}) = t_{\mathtt{frame}}(\mathtt{size},\mathtt{bitrate}) + t_{\mathtt{ack}} \tag{5.4}$$

$$t_{\mathtt{ack}} = t_{\mathtt{SIFS}} + t_{\mathtt{preamble}} + \frac{14\mathtt{B} \times 8}{1\mathtt{Mbps}} \tag{5.5}$$

For IEEE 802.11b/g, using a long preamble, the duration of those parameters is presented in Table 5.1. Note that the backoff time is actually a random number calculated in a window that doubles each time a collision is detected. As referred before, in this work we assume a TDMA network, and thus, collisions are not expected. For that reason we assume that the maximum possible value to be $135\mu s$ (for IEEE 802.11g) or $620\mu s$ (for IEEE802.11 b), corresponding to the maximum initial random value.

5.2.2.1 Useful slot duration in worst case ad-hoc RA-TDMA

When communicating within an ad-hoc mobile network of robots, specifically if using the RA-TDMA implementation presented in the previous chapter above, there are certain particularities that must be taken into account. To begin with, all robots are considered to have similar bandwidth requirements, therefore slots are equally distributed amongst robots with duration t_s (Eq. 5.6), where $T_{\mathtt{up}}$ is the round duration and N is the number of robots. Adding to that, the beginning of each slot is reserved to the transmission of the synchronisation packet with $O_{\mathtt{RA\text{-}TDMA}}$ data bytes and transmitted at the lowest multicast bitrate of 1Mbps. Consequently, the time in each slot useful for transmitting unicast packets (t_{use}) is calculated as shown in (Eq. 5.7).

$$t_s = \frac{T_{\mathrm{up}}}{N} \tag{5.6}$$

$$t_{use} = \frac{T_{\mathrm{up}}}{N} - t_{\mathtt{mcast}}(O_{\mathtt{RA\text{-}TDMA}}, \mathtt{1Mbps}) \tag{5.7}$$

Algorithm 2 Sending packets: “t_ucast” represents the time it takes to transmit a packet of size “size” at a bitrate of “bitrate”, see (Eq. 5.4); “find_route” is the call to the routing algorithm; “end_slot” represents the time when the slot ends;

```
Input:  packet p
 1: r = find_route(p.source(),m.destination())
 2: next_hop = r.first()
 3: slot_remaining_time = end_slot − current_time
 4: msg_time = t_ucast(m.size(),bitrate)
 5: max_hops = floor(slot_remaining_time ÷ msg_time)
 6: if (is_multihop_per_slot) then
 7:     end_of_tx = current_time + r.length() × msg_time
 8: else if (is_unihop_per_slot) then
 9:     end_of_tx = current_time + msg_time
10: end if
11: if (max_hops > 0) then
12:     send_packet(p)
13:     if (end_of_tx < end_slot) then
14:         schedule_next_send(end_of_tx)
15:     end if
16: end if
```

5.2.3 One-hop per slot forwarding

The *one-hop per slot* propagation strategy is the simplest forwarding approach to TDMA communications. Using this method, when the time comes for an agent to transmit its packets (Algorithm 2) it will look at its queue to select the next packet (p) to transmit. Then, using the topology information contained in the TDMA protocol packet, and, for example, Dijkstra’s algorithm, it chooses the route to reach the destination (line 1); for the scope of this work we use the minimum hop count metric. After determining the next hop (line 2), it calculates the time remaining in the current slot, the amount of time required to transmit the packet, and uses that information to check how many packets fit in the slot (lines 3-5). This allows to calculate the time at which the transmission is expected to end (line 9). Finally, using all this information, if the packet fits in the slot the agent will transmit it and schedule the next transmission.

Using this method each robot transmits packets within its own slot, only, as normal in a TDMA approach. Therefore, when a one-hop neighbour of that agent receives a packet (Algorithm 3), it will check if it is the final destination of that packet. If the agent is the final destination it will process the

Algorithm 3 Receiving packets

```
Input: packet p
if (p.destination() == myself) then
    process_packet()
else if (is_multihop_per_slot) then
    r = find_route(myself,m.destination())
    next_hop = r.first()
    slot_remaining_time = end_slot − current_time
    msg_time = t_ucast(m.size(),bitrate)
    max_hops = floor(slot_remaining_time ÷ msg_time)
    if (max_hops > 0) then
        send_packet()
    else
        queue_packet()
    end if
else if (is_unihop_per_slot) then
    queue_packet()
end if
```

packet. Conversely, if the agent is only an intermediate hop, it will queue the packet and forward it in its own slot. This approach is illustrated in Figure 5.1.

5.2.3.1 Comments on the end-to-end delay

Due to the multi-hop nature of ad-hoc networks, agents need to forward information. However, due to mobility, topology changes occur frequently throughout the lifetime of the network, thus links are constantly being created and destroyed. Being unable to know at design time which topology changes may occur, we must consider the worst case topology to get an upper bound on the resulting end-to-end delay. This topology is depicted in Figure 5.2, being comprised of agents in a line formation and sorted by their IDs (see [35] for a formal proof). This topology results from a coupling between the network diameter (number of hops between the two farthest nodes) and the circulation of the slots in the TDMA round.

Looking at Figure 5.2, if we wish to send information from the last node (4) to the first one (0), assuming there is no other information being transmitted in the network, this transmission will use $(N-2)\times(N-2)+(N-1)$ slots where N is the number of agents (5 in this case). This delay corresponds to the following: initially agent 4 sends its packet to agent 3, the latter needs to wait for $N-2$ slots until its slot arrives so that it can forward the packet to robot 2, and so on until it reaches the destination at robot 0. Thus, in total we have $N-2$ intervals (one for each forwarding robots, 3 to 1) of $N-2$ slots plus the $N-1$ slots in which the packet is actually transmitted. In this example, this delay amounts to 13 slots.

It is clear that this value is already rather pessimistic in practice, since the worst-case topology may seldom occur. Even with line topologies, if we consider a packet sent in the other way, i.e., from robot 0 to robot 4, the end-to-end delay is reduced to $N-1$ slots (4 in this case).

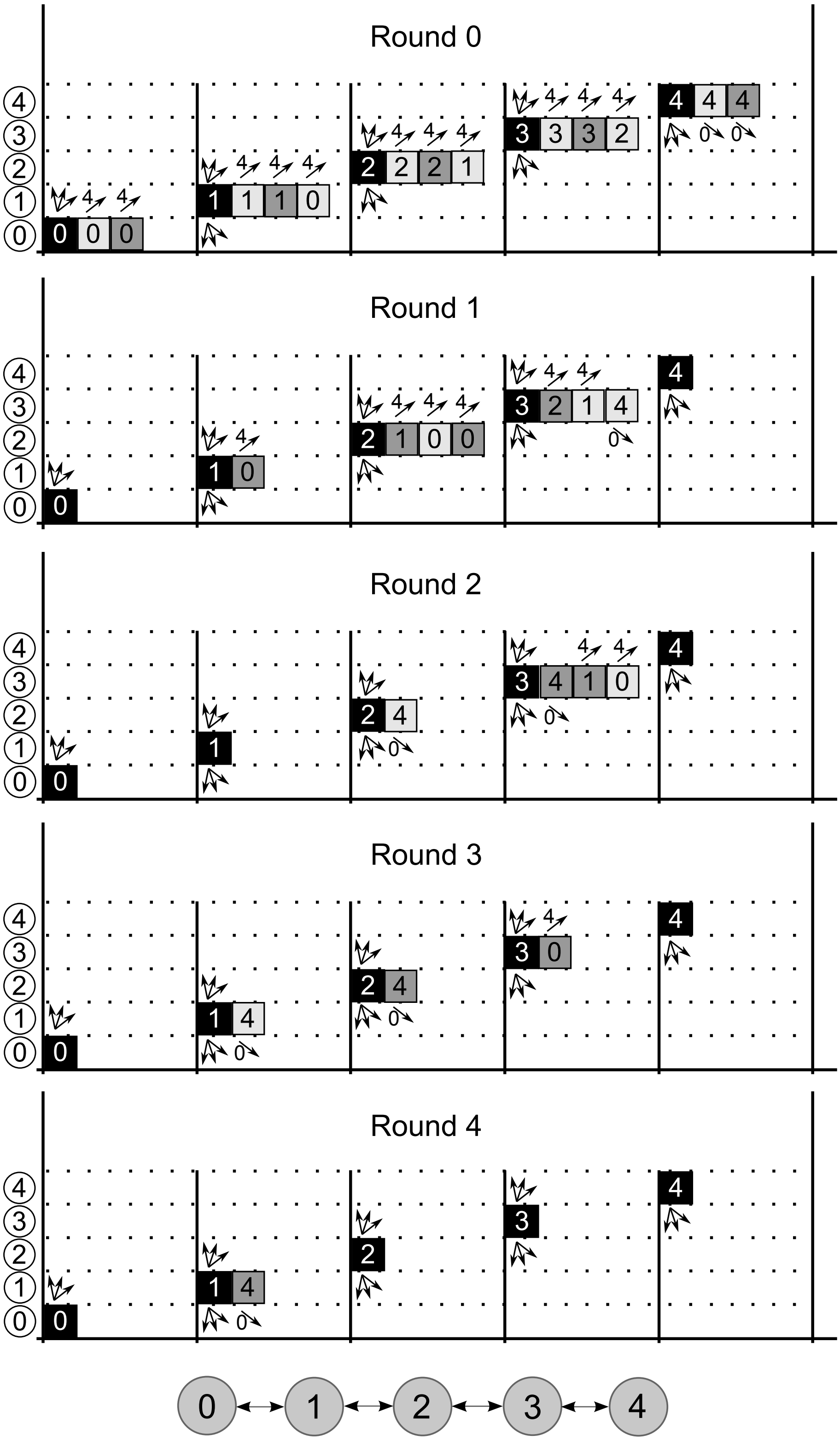

Round 0
Round 1
Round 2
Round 3
Round 4
0
1
2
3
4

Figure 5.1: One-hop per slot forwarding: time diagram – robots are in a line topology as illustrated in the bottom of the figure; each robot has two packets in its queue (light grey, and dark grey); all with the same priority; robots 0 through 3 send to robot 4 and this one sends to robot 0.

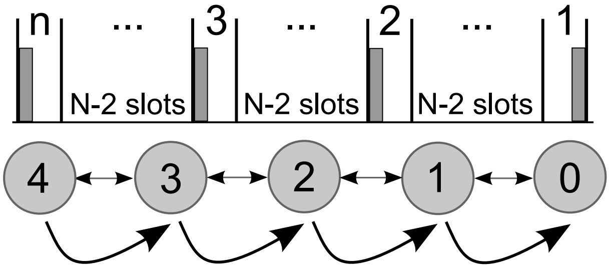


Figure 5.2: Propagating a packet “counter-slot-wise” with $N = 5$ and $n = 0..4$. Agent 4 will transmit its information, once agent 3 receives it, it will have to wait for its slot ($N - 2$ slots). Once the slot starts, agent 3 will retransmit the packet to agent 2, and so on until the information gets to agent 0.

However, there is still another aspect that may further increase the end-to-end delay. In this topology, nodes not only send their own information but also forward information from the other nodes. For example, node 1 in the figure may need to forward information from nodes 2 through n to node 0, plus information from node 0 to nodes 2 through n, and at times transmit its own locally generated information. Thus, slots can be seen as communication links whose bandwidth is shared among different routes (Figure 5.3) leading to complex interference scenarios. This situation bears some similarity with wormhole routing [104] and, again, may lead to very long worst-case end-to-end delays that seldom occur in practice.

This interference scenario is visible in Figure 5.1 where several packets are sent at the same time through routes that share several slots. In particular, looking again at the end-to-end delay of the packets sent by robot 4 to robot 0, note that the 2 packets involved do not fit in slot 3 of round 1 and only one can be transmitted. The second packet (darker) is queued and transmitted in the next round, thus implying an extra round (5 slots) in the end-to-end delay caused by the worst-case topology (13 slots) leading to a total delay of 18 slots.

Overall, the typical one-hop per slot forwarding approach introduces a coupling between forwarding and the TDMA slot circulation and it also allows routes to share slots (i.e., slots are used on behalf of nodes other than their owners), which creates complex interference scenarios. This lead us to search for different forwarding approaches that reduce such interference.

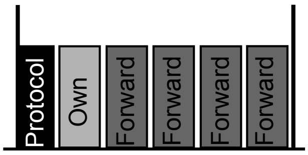


Figure 5.3: Slot structure for five agents – Each slot must contain space within it to forward packets from other nodes. This is simply an illustration of a possible operational scenario. Transmissions follow each other immediately within the slot.

5.2.4 Multi-hop per slot propagation

With the objective of maintaining the reliability and predictability benefits of TDMA schemes while strongly reducing the end-to-end propagation delay between non neighbour agents, we propose an alternative to the previously described forwarding protocol. In this approach we propose to allow agents to forward packets immediately within the slot of the agent that initiated the transmission.

As mentioned before, the synchronisation packet transmitted by the TDMA synchronisation protocol also contains information about the network and its topology. Therefore, as in the previous case, we know the network topology and we can construct a route to the destination of the packet. In addition, we know the TDMA round structure and consequently know when the slot of the current sender ends. Therefore, similarly to the previous approach, a robot will choose the next packet (p) to be transmitted, calculate the route, next hop, and how many packets fit in the slot. However, in this case, the packet transmission time takes into account the total number of hops (line 7), i.e., the time it takes for the packet to arrive at its destination. However, the transmission of the 1st hop is always done as long as it fits in the slot, even if the arrival time is beyond the slot end.

When an agent receives a packet (Algorithm 3), similarly to the *one-hop per slot* solution, it checks if it is the final destination of the packet, and if it is, it processes the packet. However, if the packet is destined to another agent, the receiver will calculate the next hop towards the destination (lines 4-5), and check if the respective transmission still fits inside the current slot (lines 6-8). If it does, the packet will be forwarded immediately, if it does not the packet will be queued and forwarded later in its own slot. This approach is illustrated in Figure 5.4, using the same traffic pattern as in Figure 5.1.

5.2.4.1 Comments on the end-to-end delay

This method reduces substantially the forwarding of data in slots of other nodes than the originator. Consequently, it reduces the interference of different routes in the slots. Moreover, since forwarding is done immediately inside each slot, the coupling of forwarding and the TDMA slot circulation is significantly reduced.

However, both benefits do not always hold. In fact, it still occurs whenever the data cannot be immediately forwarded until the destination and has to be queued in the way. For example, in Figure 5.4, node 4 can only transmit 3 packets in its slot, thus its packets transmitted to node 0, which is 4 hops away, have both to be queued in the way, in this case, in node 1. But since slot 1 in round 1 is already full with other packets, the packet of node 4 is delayed an extra round. The end-to-end delay is thus 8 slots. Was it not for the interference in slot 1 of round 1, and the end-to-end delay of the first packet of node 4 (lighter) would have been 3 slots, only.

Therefore, when the data can be immediately forwarded to the destination in just one slot, the end-to-end delay is significantly reduced and the impact of the TDMA slot circulation is eliminated. However, it becomes similar to the previous one-hop per slot approach whenever the slot of the sender is not sufficiently long for completing the end-to-end forwarding and the data is queued in intermediate nodes. Still, as we will see further on, experiments show that this approach significantly reduces the end-to-end delay on average.

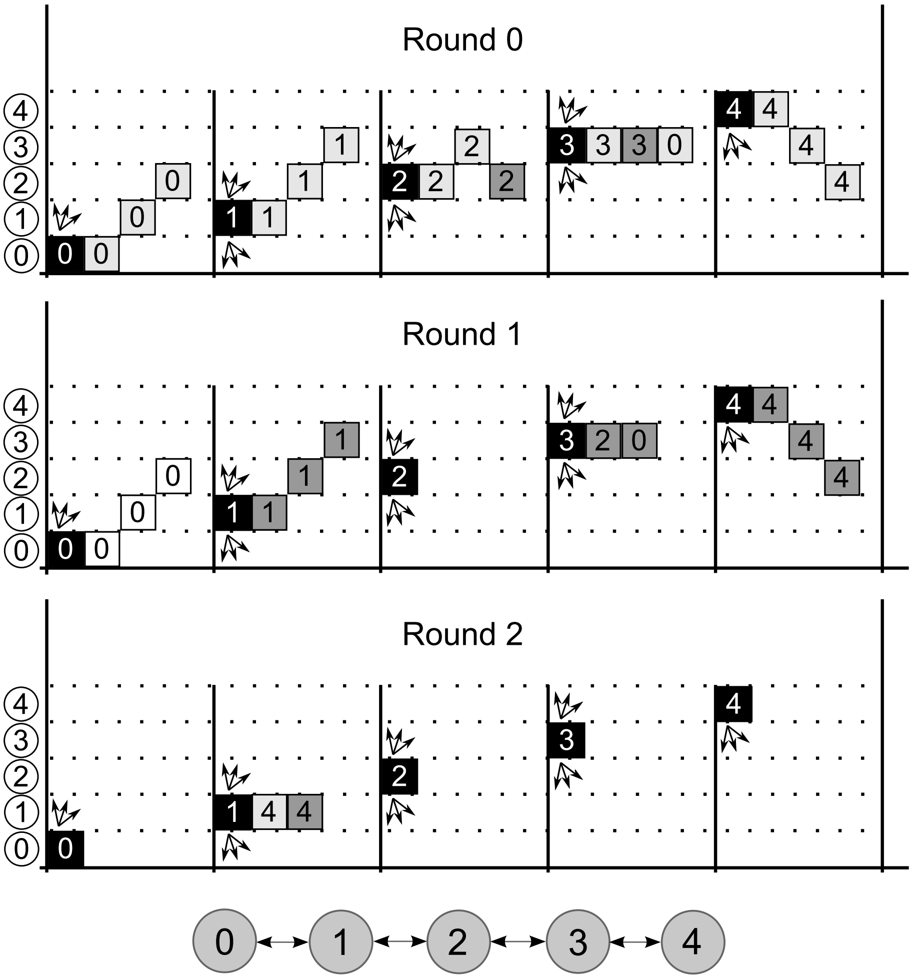


Figure 5.4: Multi-hop per slot forwarding: time diagram – robots are in a line topology as illustrated in the bottom of the figure; each robot has two packets in its queue (light grey, and dark grey); all with have the same priority

5.2.5 Multi-hop per slot with full slot isolation

In both of the previous solutions presented, the topology has a strong influence on the time required to deliver each packet. However, this dependency is not limited to the distance, i.e. number of hops, between source and destination, and amount of data produced by the sender. An undesirable property of the previous solutions, is the dependency on the routes themselves, their direction of propagation with respect to the TDMA slot circulation, and in which intermediate hops the packets are buffered.

In order to remove this dependency, we propose a slightly different approach where packets generated by a robot are only forwarded in its own slot. In this solution, whenever a slot starts, all robots look at their queues, if they have a packet whose source is the owner of the slot, then they start forwarding it. Note, however, that only one robot should have only one packet from the slot owner in its forwarding queues, i.e., there can only be at most one route in progress originating from a given robot. The source robot, for every packet it sends, after sending it, sleeps for the time required for the packet to reach its destination. Only then, the owner of the slot sends the next packet. This sleeping period is counted inside its own slot, only, and suspended during the remaining time of the TDMA round.

Figure 5.5 shows this approach applied to the same traffic pattern of Figures 5.4 and 5.1. Note, now, that the forwarding of packets can be followed simply by looking at the same slot across rounds (vertically aligned in the figure). In particular, note that the end-to-end delay of any packet depends exclusively on the length of its route and the width of the slot of its generator and not on the information generated by other robots or the direction of its routes.

In other words, the end-to-end delays for each robot can now be computed knowing the communication requirements of that robot, only, plus the current topology resulting in much simpler analysis and significantly shorter worst-case delays.

5.2.5.1 Response time analysis

Within RA-TDMA, each slot has a window of length t_{use} (Eq 5.7) available for its unicast communications. However, since the transmission of packets is non-preemptive and t_{use} is not necessarily an integer combination of the packets transmission times, it may happen that the packet to be transmitted at the end of the slot does not fit in. As common in TDMA approaches, we do not allow overriding the slot duration and thus, in such case, we postpone the packet for the next round. This may lead to wasting some time at the end of the slot which we call Inserted Idle Time (IIT). Using IIT to schedule traffic in strict windows has been studied in the past [105] and applied in many diverse circumstances [106][107].

In particular, if we consider that the packet periods are integer numbers of rounds (neglecting their variation between T_{up} and $T_{\text{up}} + \Delta$) and the packets are always generated at the beginning of the respective slots, then we can use the model in [105] with a few small adaptations. According to this model, we can increase the duration of the packets by an inflation factor that warps the guaranteed usable portion of the slot ($t_{use} - t_x$, where $t_x = \max(\text{IIT})$) to make it cover the whole slot (t_s). Then, we can assess the schedulability of the traffic generated by one robot, i.e., whether all end-to-end delays do not exceed the respective deadlines, using any common schedulability analysis for preemptive scheduling on uniprocessors [108] [109] [110] but confined to periodic slots, such as in 2-level hierarchical scheduling [111]. In

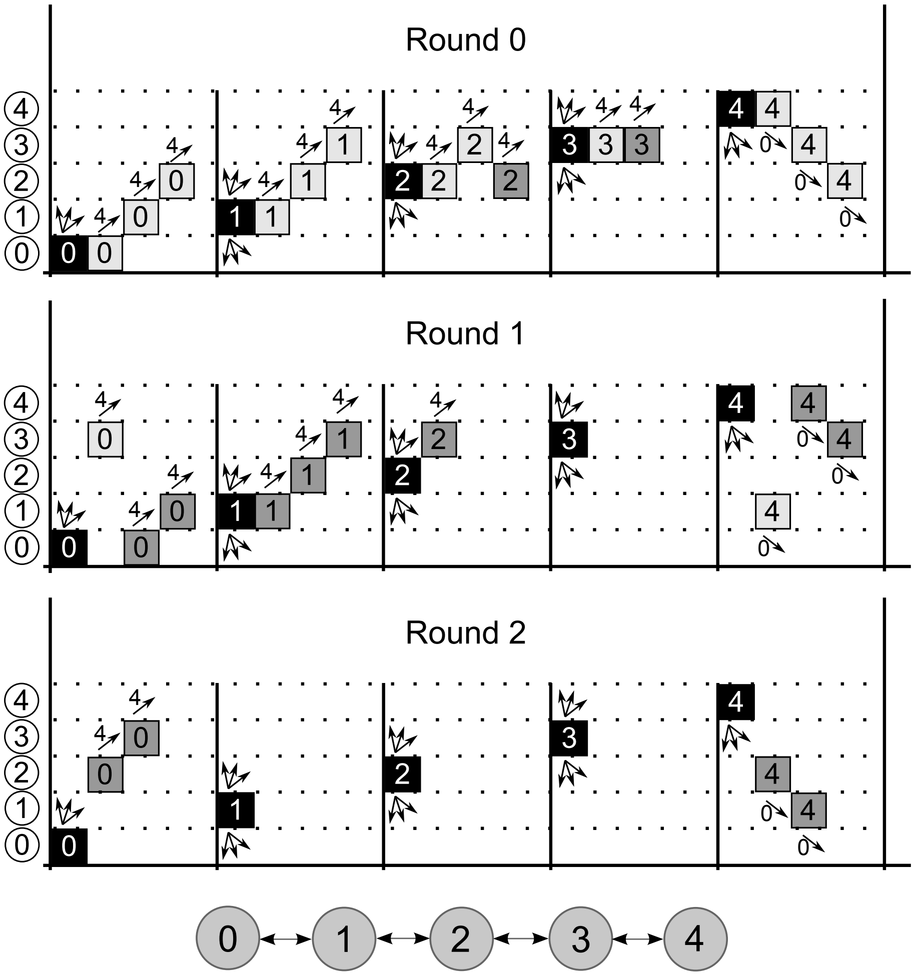


Figure 5.5: Full partition of slots: timeline for the same traffic pattern as in Figures 5.4 and 5.1. – robots are in a line topology as illustrated in the bottom of the figure; each robot has two packets in its queue (light grey, and dark grey); all packets have the same priority

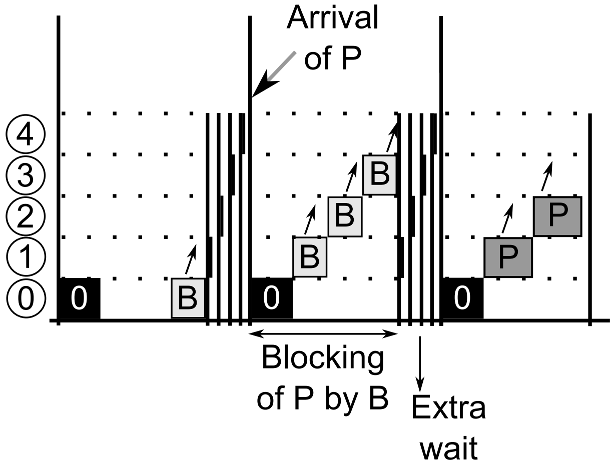


Figure 5.6: Illustration of worst case blocking – packet P arrives just after the lower priority packet B starts its transmission; packet B starts its transmission in the previous slot but uses all the current slot with the its forwarding, thus forcing packet P to wait until next slot.

this setting, the inflated packets can be preempted at the exact periodic slot boundaries. However, mutual interference between packets produced by the same node is still non-preemptive since any packet will wait for the previous to be fully forwarded to its destination before starting arbitration for access to the channel in its slot (Figure 5.6). This waiting time is called blocking and it is typical in non-preemptive scheduling. Thus, this is the approach we follow in the remainder of this section with an adaptation to include blocking by lower priority packets.

We start by defining the inflation factor (β) we will use. This depends on several parameters, namely, the size of the slot (t_s), the useful size of the slot (t_{use}), and the duration of the maximum IIT (t_x), which can be bounded by the longest packet transmission time among the packets produced by this robot (k), i.e., $t_x \leqslant max_{\forall i:Sr_i=k}(C_i)$ (Eq. 5.8).

$$\beta = \frac{t_s}{t_{use} - t_x} \tag{5.8}$$

Then, to maximise the blocking term, we search among all lower priority packets that have the same source as the packet being blocked, $lp(i)$, for the one that has the longest accumulated forwarding time, except for the transmission of its first packet (Figure 5.6). However, there can only be blocking if there are sufficient lower priority transmissions to overload the slot of this robot in the previous round. Otherwise the blocking is 0. This is formalised in Eq. 5.9, where C_j represents the time needed to transmit packet j across one hop, β is the inflation factor, and H_j is the number of hops separating source (Sr_j) from destination (Ds_j) and it is a topology dependent parameter.

$$B_i = \begin{cases} 0 & , \; \sum_{j \in lp(i)} C_j \beta H_j \leqslant t_{use} \\ \max\limits_{\substack{j \in lp(i) \\ \wedge Sr_j == Sr_i}} C_j \beta \left(H_j - 1\right) & , \; \texttt{otherwise} \end{cases} \tag{5.9}$$

Once the blocking time is over, packet i can be scheduled. Nonetheless there is another source of delays, the locally generated interfering traffic, which is comprised of all the higher priority packets,

$hp(i)$, generated while packet i is waiting for its transmission. Eq. 5.10 shows the interfering load generated during an arbitrary interval δ including the initial blocking and the high priority traffic, where $\left(1+\left\lfloor\frac{\delta}{T_j}\right\rfloor\right)$ is the number of activations of higher priority packet j during δ, including on the limit of the interval.

$$I_i(\delta) = B_i + \sum_{\substack{j \in hp(i) \\ \wedge Sr_j == Sr_i}} \left(1+\left\lfloor\frac{\delta}{T_j}\right\rfloor\right) C_j \beta H_j \tag{5.10}$$

Using Eq. 5.10, we know the time taken by forwarding the interfering traffic. However, that time does not consider the TDMA round, specifically the fact that all related transmissions must occur within slots of the robot that generated the packets, only. For this purpose we introduce Eq. 5.11 where the first term represents the number of full slots (equal to the number of integer TDMA rounds) available in an interval δ, while the second term represents the remaining time used in the last, potentially incomplete, slot. Remember that $t_s = T_{\text{up}}/N$ where N is the number of active robots. For example, if an interval δ corresponds to one and a half slots, then in the real timeline we will need one round and half a slot.

$$\begin{aligned}
\mathcal{R}(\delta) &= \boxed{\left\lfloor\frac{\delta}{t_s}\right\rfloor T_{\text{up}}} + \boxed{\delta - \left\lfloor\frac{\delta}{t_s}\right\rfloor t_s} \\
&= \left\lfloor\frac{\delta}{t_s}\right\rfloor (T_{\text{up}} - t_s) + \delta \\
&= \left\lfloor\frac{N\delta}{T_{\text{up}}}\right\rfloor \left(T_{\text{up}} - \frac{T_{\text{up}}}{N}\right) + \delta \\
&= \left\lfloor\frac{N\delta}{T_{\text{up}}}\right\rfloor \left(1 - \frac{1}{N}\right) T_{\text{up}} + \delta \\
&= \left\lfloor\frac{N\delta}{T_{\text{up}}}\right\rfloor \frac{(N-1)}{N} T_{\text{up}} + \delta
\end{aligned} \tag{5.11}$$

Therefore, we need to compute the interference window W_i during which the interfering load is transmitted, which already accounts for the limitation to use a single slot per round. This window can be computed with a fixed point iteration as in Eq. 5.12, with $W_i^0 = \mathcal{R}\left(B_i + \sum_{\substack{j \in hp(i) \\ \wedge Sr_j == Sr_i}} C_j \beta H_j\right)$, which will either converge or cross the deadline D_i in a bounded number of steps.

$$W_i^{m+1} = \mathcal{R}\left(I_i\left(W_i^m\right)\right) \tag{5.12}$$

Finally, in order to compute the end-to-end delay of packet i, referred to as its response time R_i, we need to add the time to transmit and forward packet i itself to the respective interference window as in Eq 5.13.

$$R_i = \mathcal{R}(C_i \beta H_i + I_i(W_i)) \tag{5.13}$$

Table 5.2: OMNeT++ simulation parameters

Parameter	Value
retryLimit	1 (minimum)
cwMinData	4
cwMinMulticast	4
transmitterPower	2mW
bitrate	12Mbps

5.3 Experimental Validation

In order to validate our proposal, we have implemented this protocol in the OMNeT++ simulation environment. This validation focuses on static network configurations. When the topology changes, each robot checks again all its routes. If at least one route increased its length, then the schedulability of the traffic must be rechecked.

5.3.1 Comparison with the one-hop per slot forwarding approach

In our simulation we reproduce a case where four agents are spread around a building and a remote operator is watching video streams transmitted from those agents. We use the same video data as the work in [98] with the average size of a video frame of 5360B, requiring the transmission of four 1340B UDP packets, i.e., 1408B considering the UDP/IP (28B) and IEEE 802.11 (40B) headers.

The simulated network was an IEEE 802.11g network with long preamble. All agents are based on the OMNeT++ INET framework AdhocHost, namely we used the Net80211 example as a base for development. Some of the simulation parameters used were changed from the default, as summarised in Table 5.2. Five agents were placed in a line formation, all of them running a TDMA schedule with a period of $T_{\text{up}} = 50ms$ (thus $t_s = 10ms$). In addition, a clock skew was applied to all agents, the skew was calculated as a random number between 0 and 100 μs per simulated second, which corresponds to relatively poor clocks. We carried out two experiments for each of the forwarding schemes. All of the experiments were performed using the topology depicted in Figure 5.2. In the first one, we assume the remote operator is connected to agent 0, therefore we programmed agents 1 to 4 to generate traffic to be delivered to agent 0. This situation corresponds to the worst case forwarding delay using the *one-hop per slot* method. Note that in order to have all agents generating data, agent 0 also generates traffic destined to agent 4, but this data was just for symmetry purposes and was discarded upon reception. In the second experiment, we configured agents 0 to 3 to generate data to agent 4, assuming the monitoring operator is connected to the latter. This corresponds to the best case forwarding delay using the *one-hop per slot* method.

In order to calculate the maximum number of packets that can be generated in each agent without saturating the medium (g_a), we must recall Figure 5.3 where we show that each robot can only occupy a fraction of the size of a full slot. If we now consider the topology presented in Figure 5.2 and the traffic pattern we just described it becomes clear that one of the robots (robot 1 and 3, in the first and second experiment respectively) needs to forward the packets from all the other robots. Consequently, it is its slot that limits how much data can be produced in each robot. Assuming that all robots will be producing the same amount of data, the amount of slot capacity available for each robot on average (t_g) can be calculated using Eq. 5.14, where the capacity of that slot is divided by all robots. The useful slot time (t_{use} from Eq. 5.7) is calculated using a multicast packet containing $1+N+N*N$ bytes, and t_x takes into account IIT. Note that, unlike in the case where traffic is completely isolated within each slot, using this forwarding algorithm we must consider the longest packet transmission time from the all the packets, i.e., the maximum IIT is given by $t_x = \max_{\forall p_i \in \mathcal{J}}(C_i)$.

$$t_g = \frac{t_{use} - t_x}{N} \tag{5.14}$$

Scaling t_g to the round (Eq. 5.15), and dividing the result by the time required to transmit the four generated video packets (Eq. 5.16), we obtained approximately 4.46 video frames per second, i.e. a period of approximately 0.225 seconds (between 4 and 5 times the value of T_{up}).

$$t_r = t_g \times \frac{1}{T_{\text{up}}} \tag{5.15}$$

$$g_a = \frac{t_r}{4 \times t_{\texttt{unicast}}(1408, 12Mbps)} \tag{5.16}$$

Simultaneously, in order to guarantee that this system is schedulable using our full slot isolation approach, we used the tool we proposed in Eq. 5.13 in order to calculate the worst case response time for this system. However, in our implementation the generation of the messages is not synchronous with the TDMA rounds, an important assumption of this tool. Therefore, in order to provide the actual worst case response time in an asynchronous release system (R_i^{async}) we must take into account the queueing time each message might suffer between the time it is generated until it is first scheduled. In our implementation, similarly to the polling server approach [112], if there are no messages in the queue in the beginning of a slot the scheduler sleeps until the next round. Therefore, if a message is activated in the instant immediately after the scheduler sleeps, it will have to wait for the duration of a RA-TDMA round (T_{up}) before it has a chance of being scheduled, as in Eq. 5.17.

$$R_i^{async} = R_i + T_{\text{up}} \tag{5.17}$$

In order to measure the end-to-end latency, whenever a packet is created we embed it with the current simulation time. Eventually all packets will be transmitted to their destination, and whenever all four packets that compose a video frame reach their destination, we measure the age of the last packet to arrive, thus obtaining the end-to-end delay. The experiments were simulated ten times for a duration of 300s using different seeds for the random generators. The cumulative distribution of delays for each source agent are presented in Figure 5.7 and discussed next.

5.3.1.1 Results

The simulation results show that, in most cases, the *multi-hop per slot* paradigm has a shorter end-to-end delay than the *one-hop per slot* paradigm.

When the information only needs to be transmitted to a one-hop neighbour, Figure 5.7a and 5.7b, as expected, the slot order does not influence the end-to-end delay. Note that in this situation, if there was an absence of traffic from other robots, all approaches should behave similarly. Instead, the *one-hop per slot* and the *multi-hop per slot* without full isolation, diverge from the *multi-hop per slot* with full isolation option. This is even more evident in the former in Figure 5.7a. This is a very good example of the problems that come sharing the slot with the other robots.

This behaviour is even more evident when the robots need to communicate across two hops, as visible in Figures 5.7d and 5.7c. In these cases, the behaviour of both the *multi-hop per slot* approaches is much more predictable than that of the *one-hop per slot* approach. The latter can even have better performance in some situations where the slot order is favourable to the routes. However, the impact of interference is clearly visible as most of its end-to-end delays end up being much longer than those of the competing approaches.

The greatest advantage of the isolated *multi-hop per slot* approach is that is keeps some of the properties usually associated with time-triggered approaches, particularly TDMA buses. By confining the traffic originating in each robot to the slot it owns, each robot can keep its time guarantees independently of the traffic being generated by others. This independency is typically referred as composability in the time domain since the properties of the traffic generated by each robot can analysed in isolation and will remain valid upon integration in the team. Moreover, the specific order of hops it needs to traverse from source to destination has no impact, as only the amount of hops has an effect of the time needed to traverse them. This can also be seen in Table 5.3, where if we compare the values for the different experiments, the fully isolated *multi-hop per slot* approach is the only one where the direction of travel has a negligible impact.

Despite that, the non-isolated approach still has some good points. As the number of hops between source and destination increases (Figures 5.7f and 5.7e), the fully isolated method starts losing the advantage in terms of end-to-end delay to its non isolated counterpart, clearly showing the advantages of the latter using the bandwidth of other robots to its advantage. Therefore, we believe that it can still be quite useful under some lighter traffic patterns and topologies.

Finally, note that in our simulations the isolated approach, for which we presented the analysis, kept all the time within the calculated bounds. Table 5.3) shows a summary of the minimum, mean, and maximum end-to-end delays measured during theses experiments.

5.3.2 Experimental Comparison with Token Passing

Token passing protocols, such as the one used in [97], can greatly reduce end-to-end delays when several of the agents serve only as relays. In addition, the constant circulation of the token translates into a quick response to new packets in the transmission queues. However, the amount of transmissions needed for circulating the token within the network, may still introduce undesirable delays. TDMA protocols, on the other hand, reserve bandwidth for each node (slots) and thus reduce the amount of overhead required to maintain the temporal isolation. At the cost of higher transmission delays, the sequential slot scheme does not require any transmission to authorise the next agent. In our protocol, we still need

Table 5.3: Statistics of the simulation results – In most cases we were able so reduce the mean end-to-end delay and standard deviation of video frames. Diferences are more visible in experiment 1, where the agents are transmiting “counter-slot-wise”.

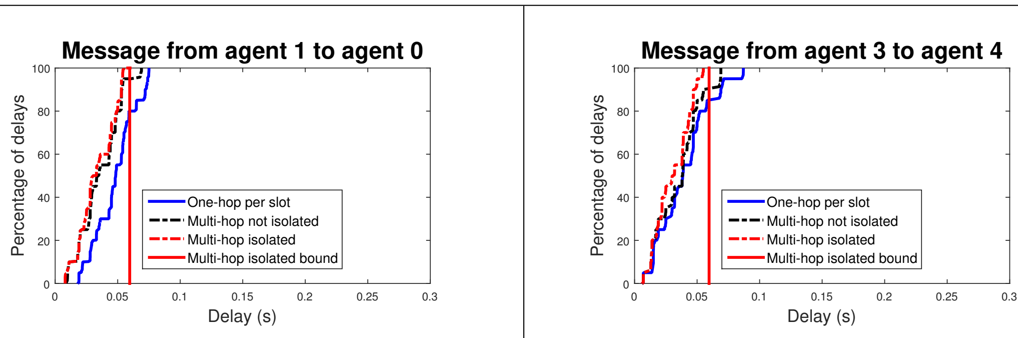
		1 hop				
		Min (ms)	Mean (ms)	Max (ms)	Std Dev (ms)	Bound (ms)
Experiment 1	One-hop	18.000	47.959	75.000	16.164	–
	Not isolated	9.000	35.859	69.000	15.684	–
	Isolated	8.000	33.439	55.000	14.275	59.600
Experiment 2	One-hop	6.000	39.628	87.000	20.541	–
	Not isolated	6.000	35.302	69.000	17.628	–
	Isolated	6.000	30.697	55.000	13.987	59.600

		2 hops				
		Min (ms)	Mean (ms)	Max (ms)	Std Dev (ms)	Bound (ms)
Experiment 1	One-hop	56.000	116.359	168.000	33.686	–
	Not isolated	53.000	77.305	102.000	14.694	–
	Isolated	54.000	78.910	104.000	14.692	109.100
Experiment 2	One-hop	23.000	99.240	159.000	41.243	–
	Not isolated	53.000	80.194	110.000	16.416	–
	Isolated	55.000	79.795	104.000	14.614	109.100

		3 hops				
		Min (ms)	Mean (ms)	Max (ms)	Std Dev (ms)	Bound (ms)
Experiment 1	One-hop	93.000	157.217	221.000	31.647	–
	Not isolated	88.000	109.742	130.000	13.465	–
	Isolated	103.000	127.753	151.000	14.307	158.700
Experiment 2	One-hop	48.000	147.792	201.000	40.124	–
	Not isolated	60.000	88.643	121.000	15.697	–
	Isolated	98.200	132.343	154.000	14.230	158.700

		4 hops				
		Min (ms)	Mean (ms)	Max (ms)	Std Dev (ms)	Bound (ms)
Experiment 1	One-hop	171.000	201.650	259.200	23.668	–
	Not isolated	95.000	150.294	221.000	33.459	–
	Isolated	109.000	160.288	199.000	29.244	208.200
Experiment 2	One-hop	45.000	172.410	227.000	44.707	–
	Not isolated	82.000	132.917	174.000	25.674	–
	Isolated	110.000	166.367	205.000	21.514	208.200

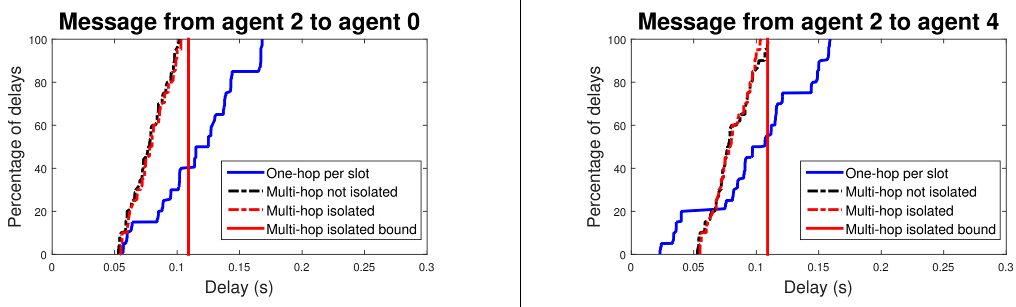
to transmit a synchronisation packet in the beginning of each slot, despite that we only transmit to the immediate one-hop neighbourhood and we tolerate occasional reception failures of the synchronisation packet. In this section, we simulate our multi-hop per slot approach in order to compare it with the application presented in [97], and thus assess if the lower overhead imposed by our proposal is enough to compensate the intrinsic delays of TDMA rounds.


Message from agent 1 to agent 0

Message from agent 3 to agent 4

(a) Experiment 1: communication at one hop

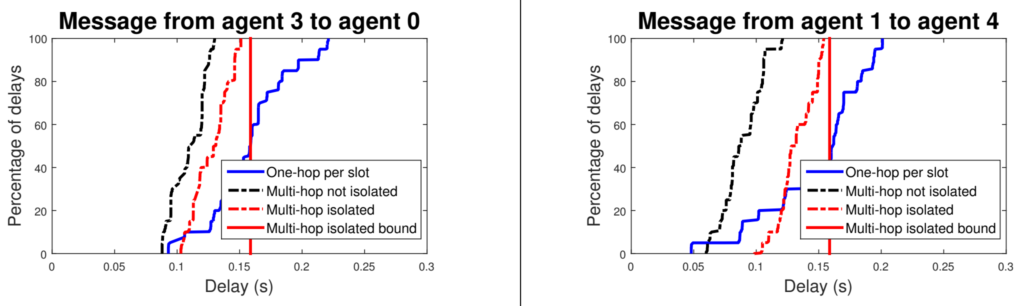
(b) Experiment 2: communication at one hop


Message from agent 2 to agent 0

Message from agent 2 to agent 4

(c) Experiment 1: communication at two hops

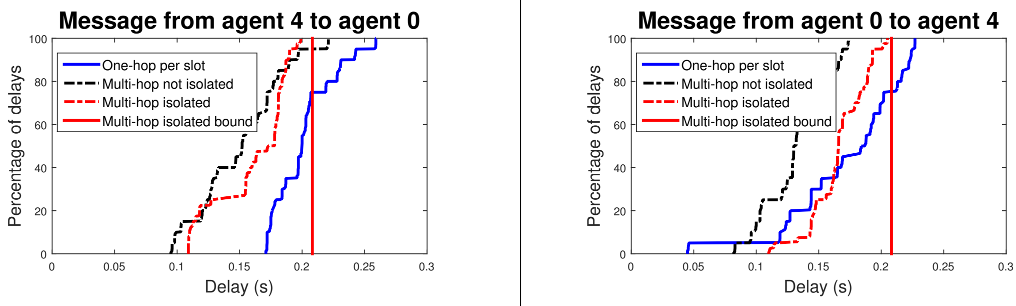
(d) Experiment 2: communication at two hops


Message from agent 3 to agent 0

Message from agent 1 to agent 4

(e) Experiment 1: communication at three hops

(f) Experiment 2: communication at three hops


Message from agent 4 to agent 0

Message from agent 0 to agent 4

(g) Experiment 1: communication at four hops

(h) Experiment 2: communication at four hops

Figure 5.7: Simulation results – As the number of hops from the source to the destination increases, the delivery delay increases as well. Nevertheless, in the worst case delay, the *multi-hop per slot* approach shows better results than the *one-hop per slot* approach. Moreover, independently of the direction of the packet flow, the isolated approach shows similar performance.

5.3.2.1 Experimental setup

The protocol described in [97] is quite complex to explain in this paper, therefore we refer to the original paper for more information. Nonetheless, the experiment presented in the paper will be briefly explained here. The authors placed five relay nodes along the path of a tunnel. These nodes do not produce data, they are only used to forward the data produced by two mobile agents. Those two mobile agents, produce a stream of 40B packets containing 20ms of a compressed audio stream, consequently these packets are produced every 20ms. After measuring the transmission delays of their protocol as 70ms, using 6Mbps, the authors decided to concatenate four of these packets before transmitting them. Consequently, the application provides a 160B packet every 80ms, each with a deadline of 150ms. The results obtained show that they are able to deliver most packets under 100ms, and 98% of the inter-arrival time measurements are under 150ms.

In order to try to replicate the conditions of this experiment in our simulation environment, we used the same simulator configuration as in the previous simulations, except for the bitrate which was changed to 6Mbps. Moreover, instead of setting the transmission power to 100mW and placing the nodes far from each other, we used the default 2mW but keeping the nodes close enough to keep a similar topology. Since we are only interested in the worst case propagation path, seven agents were placed in a line formation sorted by identifiers, all of them running a TDMA schedule with a period of 50ms (with slots of approximately 7.14ms). Note that in this work we are not looking into the route reconfiguration issues. In fact, since making the nodes mobile can only generate smaller routes, we decided to keep the agents static. Finally, we set-up agent 0 to send an audio stream to node 6, and node 6 to send an audio stream to node 0.

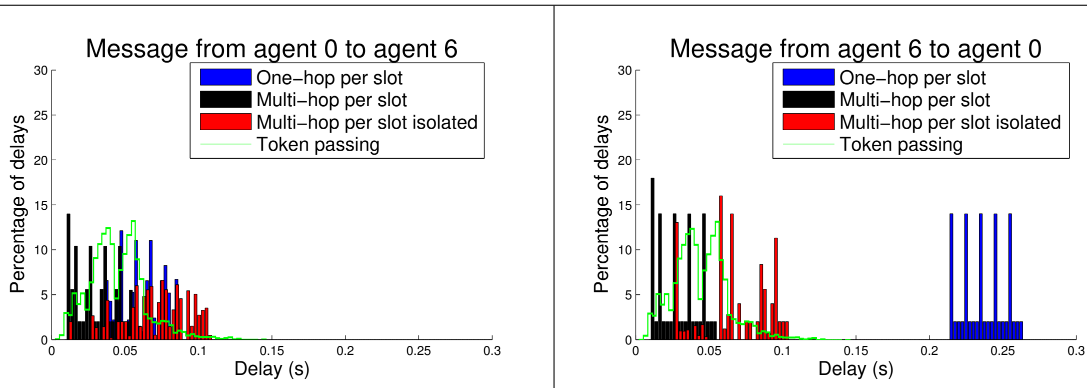
Similarly to the previous simulations, the end-to-end delay is measured from the moment a message is generated until reception at its destination. In addition, for comparative purposes, we also calculate the inter-arrival time of the packets to the destinations. These experiments were simulated ten times for a duration of 300s using different seeds for the random generators. The results obtained are presented in Figure 5.8 and discussed next.

5.3.2.2 Results

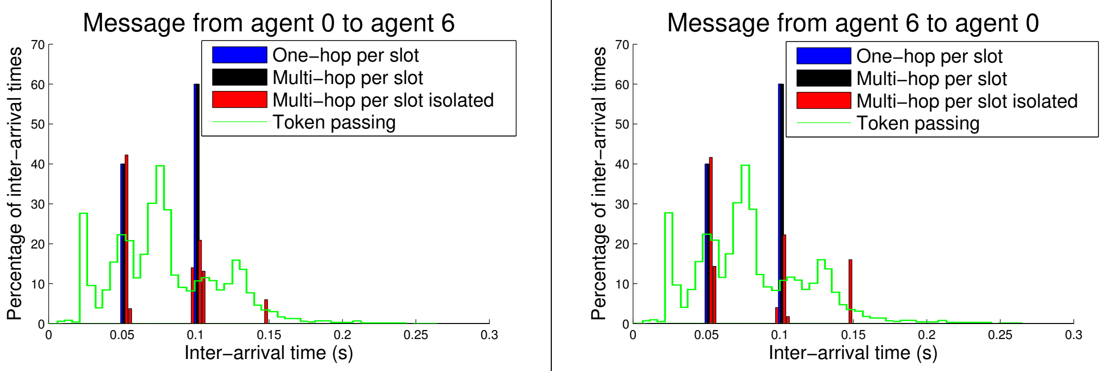
In Figure 5.8 we show the histograms of both end-to-end delay and inter-arrival time for both audio streams. The green contours were extracted from Figures 15 and 14 of [97] and were superimposed on our results. Note, however, that the vertical scale is not the same for the green contour and the bars, since the bars represent the relative number of occurrences and the contour represents the total number of occurrences out of an undefined number. Nevertheless, it is enough to compare both distributions.

The first observation we make, is the impact of slot order on the *one-hop per slot* forwarding technique. When transmitting in favour of the slot order, this technique manages to meet all the deadlines, while all the packets travelling in the opposite direction could not meet them. Conversely, when packets are forwarded using the *multi-hop per slot* technique, all packets flowing in both directions were able to meet these deadlines.

As for the maximum inter-arrival time, we could obtain values similar to the token passing protocol, however, our measured values of the inter-arrival time present very well defined steps. This is explained with the deterministic cyclic nature of the TDMA rounds. Since the packet period is not an integer multiple of the TDMA round, the inter-arrival time will include a variable number of rounds.



(a) Delay of the audio stream “slot-wise” (b) Delay of the audio stream “counter-slot-wise”



(c) Inter-arrival time of the audio stream “slot-wise” (d) Inter-arrival time of the audio stream “counter-slot-wise”

Figure 5.8: Streaming audio streams across a multi-hop network – When using the *multi-hop per slot*, is able to deliver all packets before their deadline, conversely, *one-hop per slot* can only do so when slot order is favourable. The green contour was extracted from [97] and shows the distribution of both delay and inter-arrival time of the token passing protocol. It is presented only for reference and it does not share the vertical axis with the bars.

5.4 Conclusions

In this chapter, we approached the problem of end-to-end transmission delays in a network of mobile robotic agents. Specifically, we compared three different forwarding strategies for multi-hop Time Division Multiple Access (TDMA) based wireless communications, and compared them with an application using a token passing communication protocol.

Specifically, we proposed a method aimed at reducing interference in intermediate hops, while still being able to quickly progress across the network. In addition, we proposed another method that by completely isolating multi-hop forwarding within a slot, not only keeps some of the interesting determinism and composability properties of time-triggered systems, but also allowed us to devise a relatively simple worst case end-to-end delay analysis.

We validated these approaches by simulating a network of robots in specific topologies that maximise the interference and the difference in time needed to cross them within the slot order. Simulations show the low impact of hop order when using the isolated approach, reducing the topology impact to the number of active robots and depth of the routes, thus improving the predictability of the communications temporal behaviour. Moreover, all of the timings properties of our worst-case end-to-end delay analysis were verified.

The fact that the non-isolated *multi-hop per slot* approach shows some improvement over the *one-hop per slot* alternative is encouraging for further research. Namely, we show that in most situations the maximum delay imposed by forwarding the packets just to the neighbours in each slot was higher than the delay imposed by forwarding the packets along their paths. Specifically, when transmitting against the slot order.

We had a reduction of the maximum delay between one and two TDMA rounds (49ms-91ms) when more than one hop was needed, and we measured the mean end-to-end delay of the non-isolated *multi-hop per slot* approach between 60% and 80% of that measured in the *one-hop per slot* implementation. Finally, when transmitting in the same direction as the slot order, the difference was not as significant, however, there was still improvement in most cases.

Finally, we compared the TDMA with the token passing protocol streaming two audio streams across a network of 7 agents. Results showed that by forwarding the packets to the neighbours, using a *multi-hop per slot* forwarding technique it was possible to reduce the delay in such a way that all packet deadlines were met, while benefiting from its better analysability, robustness, and lower medium usage.

Chapter 6

RF-based Ranging for Teams of Mobile Robots

One of the key factors in enabling cooperation in teams of robots is knowing the robots positions. Occasional situations may allow to build an infrastructure thus making absolute positions available; however, building infrastructures is costly and it is probably infeasible in emergency scenarios. GPS may be a possible solution for outdoors; however, it may not be available in locations such as in indoor spaces and street canyons. A possible solution, which is considered in this work, is to derive relative positions from local communication using algorithms such as the MultiDimensional Scaling (MDS) [72, 73], which minimises the dissimilarities of a connectivity matrix up to a rigid formation. However, in order to implement such solution the robots must first collect inter-robot distance information. The problem of collecting such distance measurements is generally called ranging and it is the focus of this chapter.

In a common situation, every robot may want to communicate with the other team members, therefore every robot transmits messages regularly. Every such message can be received by all robots in range, that in turn can extract received signal strength information from it, typically using the Received Signal Strength Indicator (RSSI) already provided by most Radio Frequency (RF) transceivers. As it will be explained further on, RSSI can be used to extract distance information. Consequently, any messages exchanged amgonst the robots can potentially be used for inferring localisation. In this work we use the nanotron's nanoLoc kit [113], which allows to perform Round-Trip Time Of Flight (RT-ToF) ranging, to provide a RT-ToF/RSSI hybrid ranging method aimed at small multi-robot teams, say up to 20 units. For that purpose, we propose an anchor-free online channel estimation method that uses the RT-ToF and RSSI to perform an online estimation of the log-distance path loss model. Using this model it is possible to dynamically improve the accuracy of RSSI-based distance measurements. Then, we present a distance tracker based on an Extended Kalman Filter (EKF), providing both a distance estimate and the confidence on that estimate. The advantages with relation to existing work are:

- we do not use any extra sensors, since all the data is captured from the RF transceiver module
- we do not use any a priori knowledge, the channel model is estimated online
- there are no pre-installed anchor nodes

- we support the high dynamics of RSSI with the improved accuracy of RT-ToF

6.1 Relevant Publications

The work presented in this chapter was essentially presented in the following paper:

- [17] Oliveira, L.; Di Franco, C.; Abrudan, T.E.; Almeida, L., "Fusing Time-of-Flight and Received Signal Strength for adaptive radio-frequency ranging," Advanced Robotics (ICAR), 2013 16th International Conference on , vol., no., pp.1,6, 25-29 Nov. 2013

6.2 Round-Trip Time-of-Flight Ranging

One of the methods that can be used for obtaining distances with RF communications is RT-ToF measurements. It works by measuring the time a message needs to reach the destination and return, without requiring global clock synchronisation. As implemented by the nanoLoc development kit [88], the ranging is done in two phases for increased accuracy (Figure 6.1). The first phase measures $r_1 = V \times (t_1 - t_2)/2$ and the second one measures $r_2 = V \times (t_3 - t_4)/2$, where V is the propagation speed of the RF signal. Finally, r_2 is sent back and the values are averaged, thus the whole ranging procedure returns $\overline{d} = (r_1 + r_2)/2$.

This method is very attractive because it does not suffer from some of the drawbacks of RSSI-based methods, namely the ranging results are not dependent on the signal strength. Despite that, it is only possible to range one robot per ranging operation, thus making this method less responsive to fast robot dynamics. Moreover, each complete ranging, as measured experimentally ($t_{ranging}$), takes approximately 20ms. Consequently, in a five robot team, the time required for one robot to range all the others is 80ms, and the time needed for all robots is 400ms (plus overheads). Moreover, if an offline robot is ranged, $t_{ranging}$ becomes unpredictable, and the operation timeout can be more than 100ms. In spite of that, the ranging operation produces a physical distance estimate that is accurate enough to be used for localisation.

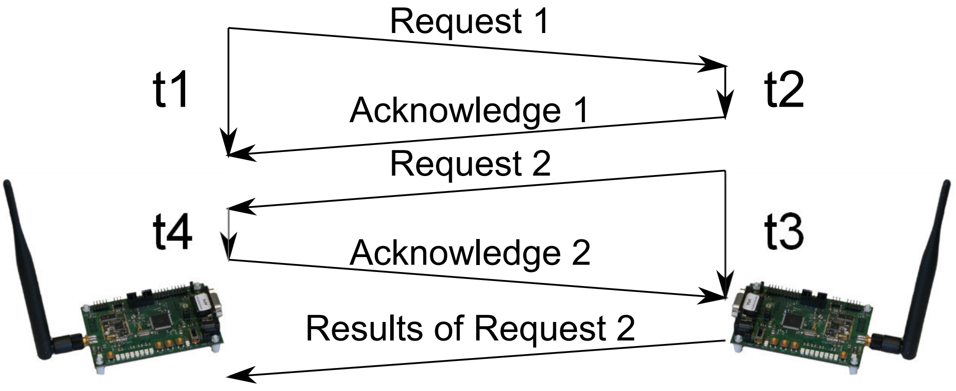


Figure 6.1: Illustration of the RT-ToF ranging process – The unit on the left ranges the unit on the right.

$$t_{ranging} = \begin{cases} 20ms, & \text{if unit is online} \\ [30, 100+]\,ms, & \text{if unit is not online} \end{cases} \tag{6.1}$$

6.3 Received Signal Strength Indicator

The RSSI is a measurement of the signal power of a received packet, which is known to attenuate as it travels in open air. Consequently, by measuring the RSSI of a message, and using a propagation model, it is possible to infer the distance to the transmitter. The exact expression that relates such attenuation with distance is rather complex and depends on several parameters such as transmission power, antenna gains, frequency of the carrier, and medium characteristics. In addition, changes on the temperature, humidity, and obstacle positioning will affect the RSSI measurements. Despite that, RSSI is a measurement that is available in most devices, thus very attractive for researchers. A widely used approximation for the relationship between signal strength and distance can be represented by the log-distance path loss model. The model is given in Eq. (6.2), where ρ_d is the RSSI value in dBm at distance d; ρ_0 is the RSSI value at a reference distance d_0 (we consider $d_0 = 1m$), and includes the aggregated effects of transmission power, antenna gains, and frequency attenuation; and α is the path loss exponent that represents the propagation medium properties.

$$\rho_d = \rho_0 - 10\alpha\log\left(\frac{d}{d_0}\right) \Leftrightarrow d = d_0 \times 10^{(\rho_0-\rho_d)/(10\alpha)} \tag{6.2}$$

Note that unlike RT-ToF, RSSI produces faster measurements since several units can measure the signal strength of another unit at the same time, i.e. if several units receive a message from another one, all of them can obtain their own RSSI value from that message.

6.4 RF-based Ranging using RT-TOF and RSSI

In this section, we describe our proposal to collect distance information between cooperating robots using both RT-ToF and RSSI information. For that purpose, we propose to use three distinct blocks (Figure 6.2):

(1) A median sliding window to filter raw RSSI data

(2) A log-distance path loss model estimator

(3) An extended Kalman filter to estimate distance between robots

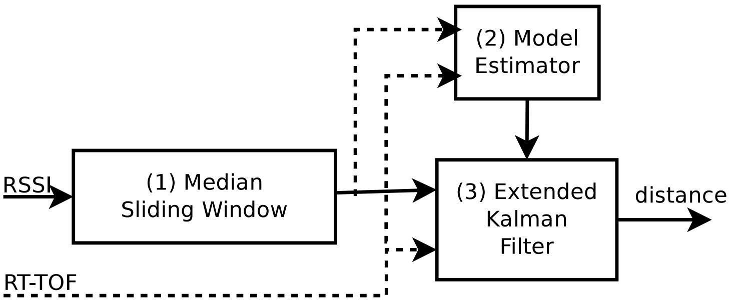


Figure 6.2: RF-based ranging: Dotted lines apply only when RT-ToF data is available

6.4.1 Filtering the RSSI readings

In indoors, the RSSI readings experience large fluctuations, even when the robots are static, due to complex propagation phenomena. For a group of mobile nodes, this instability becomes even harder to handle. Therefore, in order to filter those fluctuations, we use a sliding window median filter. Whenever an RSSI reading is received, the measured value goes through the filter that returns the median of the last k measurements (Figure 6.3). This may affect response time to true variations on the RSSI of moving robots, therefore a small value of k should be used.

6.4.2 Online channel estimation

We combine both the RT-ToF and RSSI ranging, provided by the nanoLoc devices [113], to perform an online estimation of the log-distance path loss model. With this model we are able to provide RSSI-based distance measurements accurate enough for localisation, while simultaneously coping with high movement dynamics. In order to use the propagation model, we need to estimate some of the equation parameters, namely the reference RSSI value (ρ_0) at the respective reference distance (d_0), and the path loss exponent (α). For that purpose, we define a vector of predefined n equally spaced log-distances ($\mathbf{g}_{1\times n}$) and create the matrix $\mathbf{A}_{(n+1)\times 2}$ and vector $\mathbf{b}_{(n+1)\times 1}$ (see Eq. (6.3)). The first n lines represent the previously estimated model $\widehat{\mathbf{m}_{t-1}}$, and the $(n+1)^{th}$ point represents the new measurement. Then we

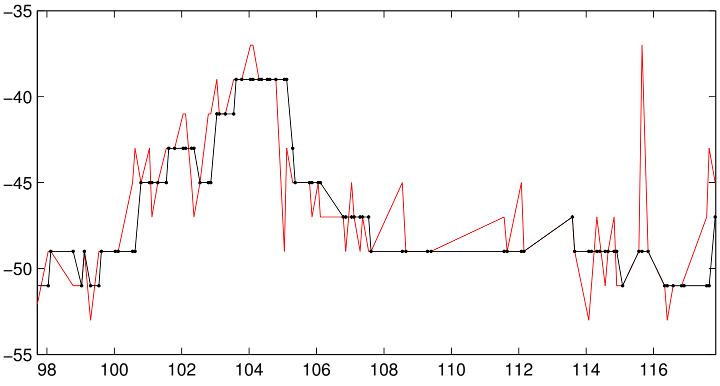


Figure 6.3: Sliding window median filter ($k = 5$) – The median filter (black) filters out the outliers in the raw measurements (red)

minimise the square error (Eq. 6.4) to obtain the new channel model $\widehat{\mathbf{m}_t}$. This allows us to use a fixed number of samples $(n+1)$, and at the same time to fuse the new knowledge into previous knowledge, with n defining the weight of the new measurement.

$$\mathbf{A}_t = \begin{bmatrix} 1 & -10\log\left(g(1)\right) \\ 1 & -10\log\left(g(2)\right) \\ \vdots & \vdots \\ 1 & -10\log\left(g(n)\right) \\ 1 & -10\log\left(\bar{d}_t\right) \end{bmatrix}, \quad \mathbf{b}_t = \begin{bmatrix} \rho_{0,t-1} - 10\alpha_{t-1}log(g(1)) \\ \rho_{0,t-1} - 10\alpha_{t-1}log(g(2)) \\ \vdots \\ \rho_{0,t-1} - 10\alpha_{t-1}log(g(n)) \\ \bar{\rho}_t \end{bmatrix} \tag{6.3}$$

$$\widehat{\mathbf{m}_t} = \begin{bmatrix} \widehat{\rho_{0,t}} \\ \widehat{\alpha_t} \end{bmatrix} = (\mathbf{A}_t^T\mathbf{A}_t)^{-1}\mathbf{A}_t^T\mathbf{b}_t \tag{6.4}$$

6.4.3 Extended Kalman filter for range tracking

In order to track the distance between robots, we implemented an extended Kalman filter [114]. The state equation is given in Eq. 6.5, where X is the state vector, d is the estimated distance, and $\dot{d}$ is the discrete-time approximation of the derivative of distance. The prediction equation is Eq. 6.6, where Δt is the time between consecutive state predictions and ω is Gaussian noise. When we measure both RT-ToF and RSSI, we use measurement Eq. 6.7, and when we measure RSSI only we use measurement Eq. 6.8. In these equations, $\bar{\rho}_k$ is the measured RSSI at time k, $\bar{d}_k$ is the measured distance using RT-ToF, ρ_0 and α are the propagation model parameters, $\texttt{bias}_d$ is the bias of the RT-ToF measurements, $\omega(k)$ is the state noise, and $\nu(k)$ is the measurement noise.

$$X = \begin{bmatrix} d & \dot{d} \end{bmatrix}' \tag{6.5}$$

$$X_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} X_{k-1} + \begin{bmatrix} \frac{\Delta t^2}{2} & 0 \\ 0 & \Delta t \end{bmatrix} \omega(k) \tag{6.6}$$

$$\begin{bmatrix} \bar{d} \\ \bar{\rho} \end{bmatrix}_k = \begin{bmatrix} d_k - \texttt{bias}_d \\ \rho_0 - 10\alpha\log_{10}(d_k) \end{bmatrix} + \nu(k) \tag{6.7}$$

$$\begin{bmatrix} \bar{\rho} \end{bmatrix}_k = \begin{bmatrix} \rho_0 - 10\alpha\log_{10}(d_k) \end{bmatrix} + \nu(k) \tag{6.8}$$

6.5 Experimental Results

6.5.1 Setup

We programmed the nanoLoc devices with the software developed for [15], which synchronises the communications with the Reconfigurable and Adaptive TDMA (RA-TDMA) scheme presented earlier in this dissertation. In our setup, we used three such units with a communication period of 250ms (Figure 6.4). Consequently, in the absence of communication failures, each node ranges one different node every

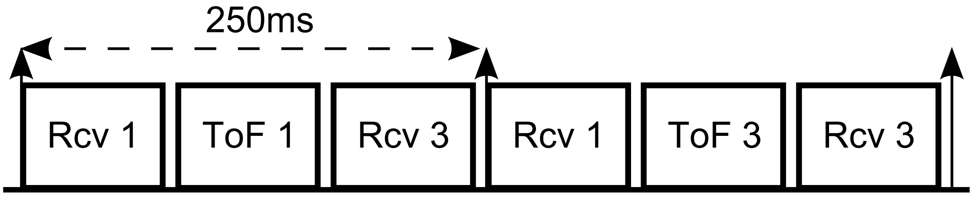


Figure 6.4: Communication period as seen by node 2 – Receives broadcast from node 1, ranges node 1, receives broadcast from node 3, receives broadcast from node 1, ranges node 3, receives broadcast from node 3, and repeats.

250ms, and receives one communication from each node between ranges. Those three nodes were then integrated in three robots [115] in an indoor laboratory (approx. 20m×6m) , with a small robotic soccer field (9.90m×5.75m). There, the robots are able to localise themselves using an omnidirectional camera, and the field lines, which we consider to calculate our ground-truth distance.

We want to show that our system correctly adapts to varying communication environments. For that purpose, we estimated the communication channel parameters in a corridor (Eq.6.9), very different from the model estimated in the soccer field for either robot (Eq. 6.10). The bias of the RT-ToF measurements was experimentally determined using the data set collected in the corridor, resulting in $\texttt{bias}_d = -0.3399$m. The state noise $\omega \sim \mathcal{N}(0, 100)$. The covariance of the measurement was set according to whether a RT-ToF measurement is available or not. When the RT-ToF was available, the covariance was a 2-by-2 diagonal matrix with the diagonal elements $x_{11} = 0,3646$ and $y_{22} = 19.6444$, otherwise, only y_{22} was considered.

$$\begin{bmatrix} \rho_0 \\ \alpha \end{bmatrix} = \begin{bmatrix} -37.6455 \\ 2.1849 \end{bmatrix} \tag{6.9}$$

$$\begin{bmatrix} \rho_0 \\ \alpha \end{bmatrix}_{\texttt{robot 1}} = \begin{bmatrix} -38.1485 \\ 1.6505 \end{bmatrix} \begin{bmatrix} \rho_0 \\ \alpha \end{bmatrix}_{\texttt{robot 3}} = \begin{bmatrix} -39.6955 \\ 1.1558 \end{bmatrix} \tag{6.10}$$

Robot 1 and Robot 3, were stopped in each side of the mid-field and robot 2 was moved manually (remote control) to preform the trajectory, shown in Figure 6.5. We logged the data from three experiments containing ground truth, RT-ToF distances, and RSSI measurements. Then, we post-processed them using five different approaches:

1. Using the corridor model without RT-ToF
2. Using the lab model without RT-ToF
3. Using the online estimator whenever data is available
4. Using the online estimator every second
5. Using the online estimator every ten seconds

In the first approach we set both models to the parameters corresponding to the corridor environment (Eq. 6.9). In the second approach, the models were set to the parameters corresponding to the lab

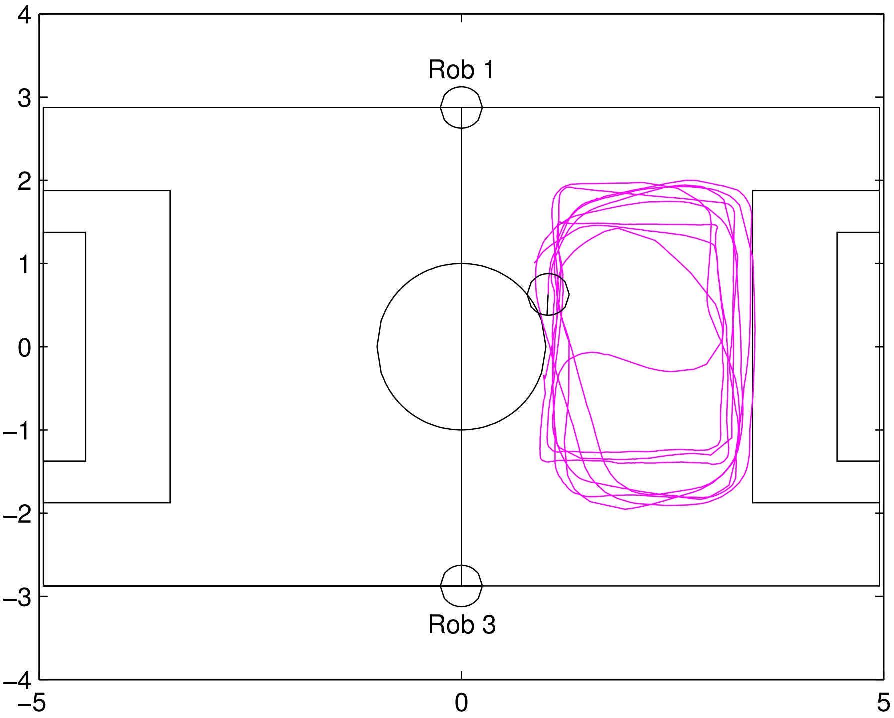


Figure 6.5: Soccer field where experiments were made – Robot 1 on the top mid-field, Robot 3 on the bottom mid-field, Robot 2 moving along the magenta trajectory

environment (Eq. 6.10). Finally, in the last three approaches, we aim at testing the adaptability of the model estimation algorithm to a different environment. Therefore, in spite of the robots being located in the lab environment, the initial channel parameter values were set on purpose to the values in Eq. 6.9 corresponding to the corridor environment.

Note that the behaviour in all three experiments was similar, favouring their confidence level. Therefore, only plots from the first experiment are presented.

6.5.2 Results

As we said before, we use an online channel model estimator to improve the accuracy of RSSI-based distance measurements. However, in order to estimate the true channel parameters, we would need to take measurements at several distances. In our case, the robots will only have access to a small observation window in a certain time frame Δt. Therefore, the estimated channel will not be the true channel, but rather a local approximation about a given distance. Despite that, if we can obtain parameters that approach the true channel locally, then we can estimate correct distances from the RSSI measurements.

Validating the capabilities of our channel estimation algorithm to adapt to the time-varying channel conditions, we present the 15-sample average of $10^{(\rho_0 - \texttt{median RSSI})/(10\alpha)} - d_{\texttt{ground truth}}$ in Figure 6.6.

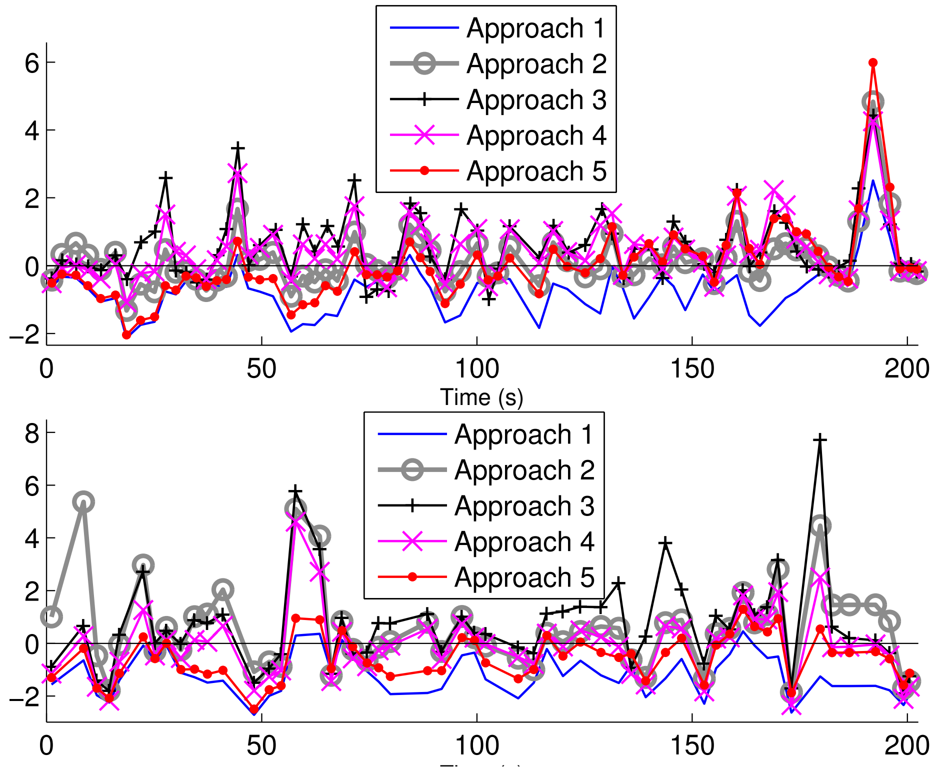


Figure 6.6: Error of the communication channel model and its impact on the accuracy of RSSI-based distance measurements ($10^{(\rho_0-\texttt{median RSSI})/(10\alpha)} - d_{\texttt{ground truth}}$): (top) Robot 1; (bottom) Robot 3

This data represents the error of the communication channel model that limits the accuracy of RSSI-based distance measurements.

When the corridor model is used (blue line with no markers), the distance is always underestimated, i.e. is biased, and since this bias will vary with the environment it cannot be filtered. Consequently, if we change the environment, the wrong model will degrade our estimate. When the lab model is used (grey 'o'), the results are substantially improved, the estimation bias tends to oscillate around the zero error instead of being negative. The third and fourth approaches (black '+', and magenta 'x' respectively) produce a result very similar to the lab model, which implies that the model is locally correct. The fifth approach (red '.') initially is very similar to the corridor model. This was expected, since it only estimates the model every ten seconds. Despite that, in the end it behaves very similarly to the lab model, which means that it converged to a locally correct model.

The effect of these different approaches on the estimated distance can be seen in Table 6.1 that summarises the results of the three experiments. Figures 6.7 and 6.8 present the distribution of the errors on experiment 1 using the five different approaches. As expected from the previous results, when the robots are using the corridor model, the Kalman filter produces an error with a large bias. Moreover, when we compare our online estimator with the lab model, we still improve on those results. That can be justified by the usage of the highly accurate RT-ToF ranging on the data fusion. Finally, by comparing the three approaches of the online estimation, we can see that by increasing the number of

Table 6.1: Estimation error in all three experiments

(a) Robot 1

		Appr. 1	Appr. 2	Appr. 3	Appr. 4	Appr. 5
Exp 1	mean	-0.7195	0.1331	0.0529	0.1866	0.0185
	std	0.8515	1.1399	0.9171	0.9988	1.3478
Exp 2	mean	-0.8199	0.1033	0.0457	0.1024	0.0431
	std	0.7640	0.9319	0.8025	0.8613	0.9024
Exp 3	mean	-0.8086	0.0778	0.0394	0.1417	0.0452
	std	0.6918	0.8676	0.8075	0.8637	0.8233

(b) Robot 3

		Appr. 1	Appr. 2	Appr. 3	Appr. 4	Appr. 5
Exp 1	mean	-1.1927	0.4849	-0.1341	-0.3496	-0.5593
	std	1.0129	2.3984	1.4321	1.4109	1.1715
Exp 2	mean	-1.2319	0.3017	-0.1866	-0.3893	-0.5199
	std	0.9976	2.7022	1.3593	1.9220	1.2927
Exp 3	mean	-1.2279	0.1954	0.0151	-0.1349	-0.6710
	std	0.9923	2.3229	1.9418	1.8334	1.3242

RT-ToF ranges we can improve the results of the estimation. This was expected because of the higher accuracy of RT-ToF when compared with RSSI ranging. However, we also show that if the medium is constant enough that allows for a small number of channel estimates, we can still have a good accuracy with RSSI only. Consequently, depending on the conditions the robots are expected to operate in, we can trade-off accuracy for bandwidth. If we have a high number of RT-ToF rangings, we have more accuracy with higher bandwidth usage, if we have less RT-ToF rangings we have less accuracy with a smaller bandwidth usage. Note that each ranging uses 20ms, during which the robots to not carry out data communications.

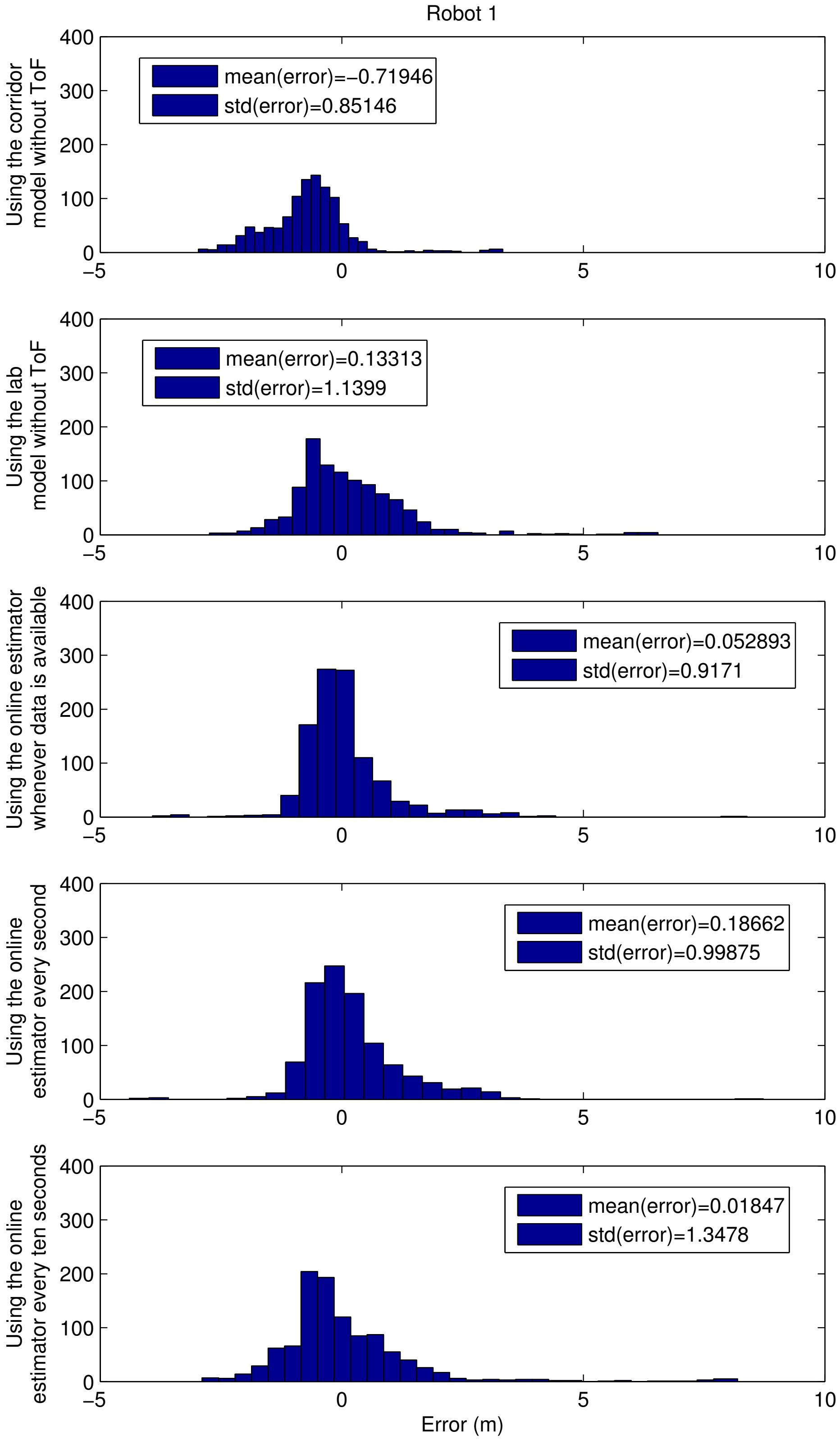


Figure 6.7: Error of distance between robot 1 and robot 2

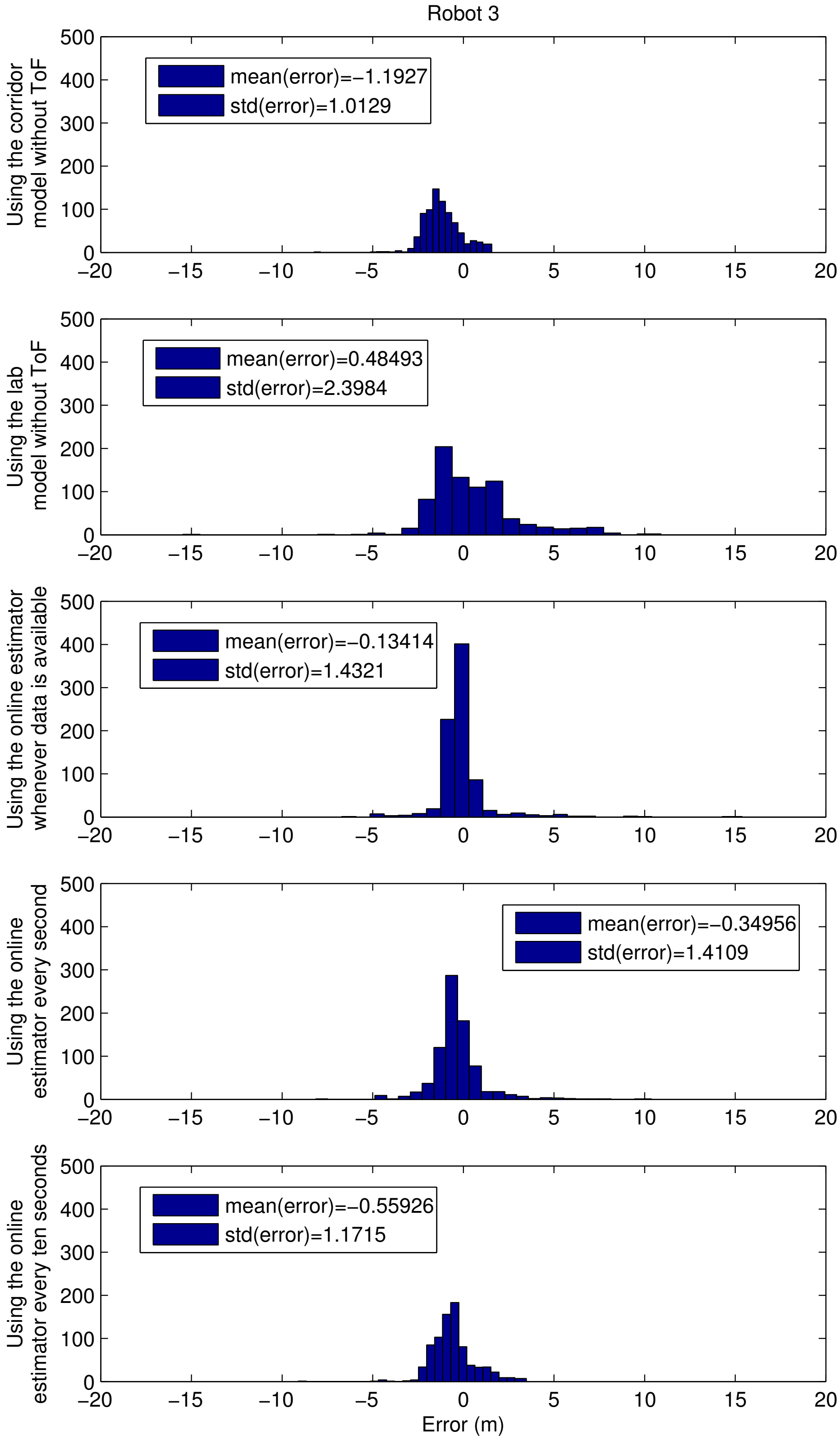


Figure 6.8: Error of distance between robot 3 and robot 2

6.6 Conclusion

In this chapter, we have successfully combined the RT-ToF and RSSI ranging methods to perform an online estimation of the indoor log-distance path loss model, which together with an EKF was used to track distance between three robots. Results show that by using our online estimator, we can approach the performance of a pre-calibrated channel model, with the advantage of supporting dynamic changes on the communication environment. Moreover, we show that it is possible to dramatically reduce the number of RT-ToF ranges, with negligible accuracy loss. This reduction is only possible if the communication channel is stable for large periods of time, however, it translates in bandwidth gain.

Chapter 7

Estimating Positions from Pairwise Distances

This chapter considers Radio Frequency (RF)-based relative localisation. First, we look into the positions of the nodes in terms of proximity, with no concern for the absolute location accuracy. We define proximity to a given node as a function of the Received Signal Strength Indicator (RSSI), i.e. a node is closer than another one if the RSSI of its transmission is higher. Thus, instead of measuring the distance between nodes, we use RF signal strength as a measure of proximity, despite the coarse relationship between them in the absence of a propagation model. A rough map of the nodes is produced with the coarse locations obtained using their signal proximity in terms of RSSI. This method has the advantage of providing relative localisation based on the strength of the communication links, i.e. a pair of nodes that possess a good link are considered closer to each other, whereas nodes that have poor or no links are considered to be "far" apart from each other.

In order to work effectively with the RSSI, we implement a scheme for filtering and sharing the RF signal strength sensed in each node via the propagation of a Signal Space Distance matrix containing pair-wise distances. This corresponds to a connectivity matrix as used in Chapter 4, with the difference that links between robots are now represented by the RSSI received from each other. The propagation of this matrix is also done with the topology tracking method proposed in Chapter 4. Then, we calculate the relative position of the nodes using MultiDimensional Scaling (MDS). We claim that such approximate relative positioning system is sufficient for a set of navigation purposes in particular to drive certain nodes to the vicinity of other ones (see Annex B), which can typically be done with variations and not absolute values.

Finally, we present an approach that uses the hybrid RSSI and Round-Trip Time Of Flight (RT-ToF) approach presented in the last chapter to obtain a higher accuracy version of the relative position estimator based on actual physical distances. However, to accomplish that, we modify the proximity estimator to handle the new approach. Those modifications, keep roughly the same base steps, but given our more informative dataset, and the final objective of providing a confidence on our estimations, some of the specific operations need to be modified as we will see within this chapter.

7.1 Relevant Publications

The work presented in this chapter was essentially published in the following two papers:

- [18] L. Oliveira, H. Li, L. Almeida, T. E. Abrudan, Rssi-based relative localisation for mobile robots, Ad Hoc Networks 13 (2014) 321–335.

- [19] L. Oliveira, L. Almeida, Rf-based relative position estimation in mobile ad-hoc networks with confidence regions, in: RoboCup 2014: Robot World Cup XVIII, Springer, 2015, pp. 383–394.

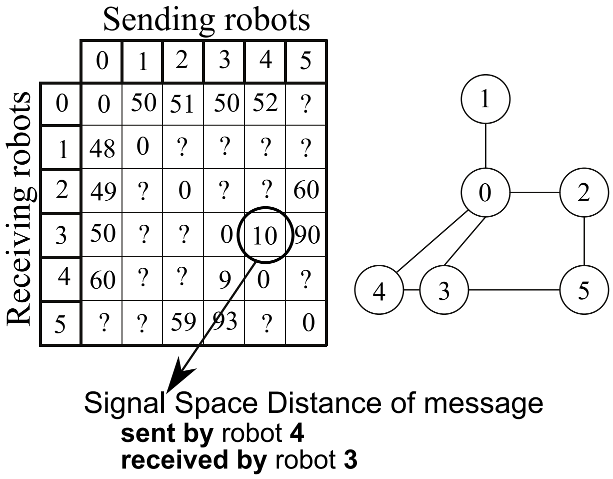
7.2 Estimating Relative Positions in Signal Space

7.2.1 The Signal Space Distance matrix

Due to high mobility together with nodes joining and leaving the team at run time, we need a scheme to keep track of the connectivity information within the group. In section 4.2.3 we described the connectivity matrix, which enables every mobile node to keep a global vision of the team composition and network topology. However, as the authors of [74] point out the binary representation of a link state provides insufficient information on the link quality, and is prone to instability. Therefore, in order to have a more accurate link state representation we propose a Signal Space Distance matrix, stemming from the Extended Connectivity Matrix in [74].

Instead of binary information, the Signal Space Distance (SSD) matrix ($\mathbf{D}_{N\times N}$), where N is the number of robots, contains the "distance" of the received packets RSSI (in dBm) to a maximum value of RSSI ($\mathtt{RSSI}_{\mathtt{max}}$) as in Eq. (7.1). $\mathtt{RSSI}_{\mathtt{max}}$ is an offset that can be used to define distance according to the offset of the transceiver being used, i.e. if the data obtained by the transceiver has no offset $\mathtt{RSSI}_{\mathtt{max}} = \mathtt{0}$

Similarly to the connectivity matrix, each robot n maintains a local version of the Signal Space Distance matrix ($\mathbf{D}^n$). For each row i, $\mathbf{D}^n(i)$ stores the Signal Space Distance readings measured by robot i, or "?" representing the absence of such measurement, including for $i = n$. However, in addition to the control variables presented before, every receiver keeps several Signal Space Distance samples per sender in a circular buffer for filtering purposes. These samples also have time-stamps associated with each one of them, which are updated whenever the samples are overwritten. When packets are received, the corresponding Signal Space Distance value is inserted in the respective samples buffer. Thus, the samples buffer stores the latest Signal Space Distance together with a local time-stamp, tolerating a configurable time without receiving messages, before declaring a link disrupted. When it is time to broadcast its own Signal Space Distance matrix, each node filters the Signal Space Distance readings in the samples buffer and stores the filtered values in the corresponding row, updates its sequence number and transmits the matrix. In addition, each node goes periodically throughout the matrix and samples buffer, removing the values whose age is greater than a given threshold. The threshold value should be small enough to cope with the team dynamics and large enough that tolerates a few missing packets. Figure 7.1 shows an example of a Signal Space Distance matrix with 6 nodes and their wireless links represented on the right side.



Receiving robots \ Sending robots	0	1	2	3	4	5
0	0	50	51	50	52	?
1	48	0	?	?	?	?
2	49	?	0	?	?	60
3	50	?	?	0	10	90
4	60	?	?	9	0	?
5	?	?	59	93	?	0

Figure 7.1: Signal Space Distance matrix: The matrix represents the *distance* between every pair of robots in the network

$$\mathbf{D}^n(i,j) = \begin{cases} \mathtt{RSSI}_{\mathtt{max}} - \mathtt{RSSI}_{\mathtt{i,j}}, & \text{is connected} \\ ?, & \text{is not connected} \end{cases} \tag{7.1}$$

7.2.2 Filtering the proximity measurements

There are two main shortcomings that affect our system. The first one, is occasional packet loss, due to the unreliability of wireless communication. This also poses a problem for mobile nodes management given the difficulty in distinguishing node absence from packet loss. The second one is the instability of the RSSI readings that propagates to Signal Space Distance. Two nodes, even placed in fixed positions without human activity or electromagnetic interference, receive fluctuating RSSI readings from each other due to complex dependence with several parameters of the medium. For a group of mobile nodes, this instability becomes even harder to handle. Some previous studies use techniques like averaging, frequency and space diversity, and signal modelling to counteract the RSSI instability. For the occasional packet losses, as explained before, each node uses a sliding window for packet buffering. In each broadcast period, if a packet is received, the Signal Space Distance value is put into the respective sliding window. If, on the other hand, the packet is lost, nothing is put in the window being the old values discarded later. Then, before broadcasting the Signal Space Distance matrix, the filter is applied. In this work we compared two different filters:

1. Using a 3 sample sliding window where all non-zero samples are taken into account.

2. Using a 5 samples window with the following rejection rule (similar to using median and mean together):

    (a) If only one non-zero value exists, it is used as is;

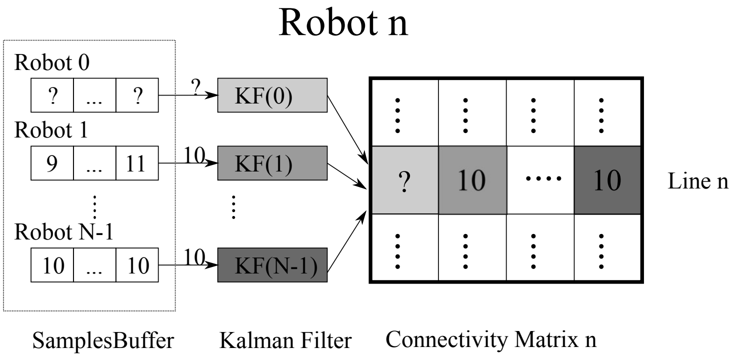


Figure 7.2: The filtering process for **D**: The samples in the buffer are filtered and used as input to the Kalman filter; the output is written in line n of $\mathbf{D}^n$

(b) if two exist, the larger is rejected;

(c) if more than two exist, the highest and the lowest are rejected and the others are averaged.

In order to track the signal strength proximity over time, a scalar Kalman filter [114] is employed for each element $d_{i,j}$ of the matrix **D**, (Eq. 7.1). The time evolution of $d_{i,j}$ is modelled as a first-order autoregressive process. The corresponding prediction rule is given by (Eq. 7.2), where k is the discrete-time index, w is the zero-mean white Gaussian noise of the process with standard deviation σ_w, and describes the shadow fading process.

$$d_{i,j}(k) = (1-\epsilon)d_{i,j}(k-1) + w(k), \tag{7.2}$$

The small value $\epsilon \in (0,1)$ introduces correlation between successive true states and ensures the wide sense stationary property of the process. It depends on the channel coherence time, as well as on the broadcast period of the signal strength proximity matrix. The measurement equation is presented in Eq. 7.3, where $v(k)$ is the white Gaussian measurement noise with standard deviation σ_v, uncorrelated to $w(k)$.

$$\delta_{i,j}(k) = d_{i,j}(k) + v(k), \tag{7.3}$$

The result is then used as the Signal Space Distance value for that node. The filtering process is illustrated in Figure 7.2. As shown in [74] this solution significantly improves the results by smoothing the measured data, and reduces undesired fluctuations.

7.2.3 Generating signal space relative positions

Using the communication scheme of Chapter 4, a small group of mobile nodes can share both topology and Signal Space Distance information for each pair of nodes, in the form of the Signal Space Distance matrix. Based on those, an approximate global vision of the whole relative positions can be generated in each node by the process depicted in Figure 7.3, which will be hereinafter explained.

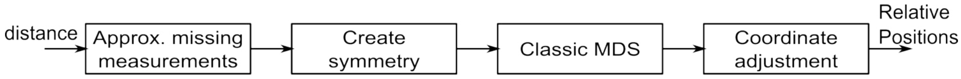
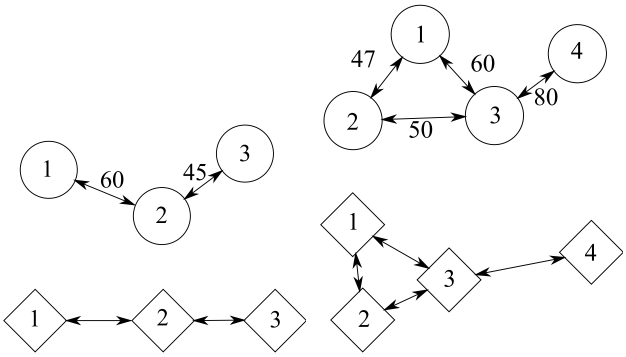


Figure 7.3: Generating Positions from the Signal Space Distance

By deeming nodes with strong RF connection to be neighbouring nodes and nodes with weak connection to be farther apart, the relationship between relative positions in physical space and proximity in signal strength space can be illustrated as in Figure 7.4. There, circles denote the physical nodes positions and bricks denote positions in signal strength space. As we may notice in Figure 7.4, the proximity in signal strength space only depends on the signal strength and not directly on the physical distance. Nevertheless, this information is still sufficient for navigation purposes, for example, to bring a given robot closer to another one using differences in the signal, and to allow a robot to have a gross perception of the topology (Annex B).

7.2.3.1 Approximating missing measurements

In practice, collecting all pairwise distances is often impossible. This is the case of wireless mobile nodes, since some links can be broken due to mobility or limited communication range. When this happens, the Signal Space Distance matrix will contain empty values – represented with "?" – and the classical MDS algorithm cannot be directly applied. Several techniques have been proposed in the literature in order to solve this problem, e.g. Map Stitching [116, 117], Iterative MDS [118], non-linear regression (NLR)[119], and the extension to MDS proposed in [120]. Despite that, Map Stitching and Iterative MDS are not able to recover the correct topology of weakly connected networks such as in Figure 7.5, a situation that can occur in formation control of small mobile teams. Moreover, the Iterative MDS and NLR both need to perform iterations, consequently, due to the dynamics of mobile robot, they are undesirable. Finally, the extension to MDS assumes that there are two groups of nodes: 1) nodes forming a fully connected network; 2) nodes which are fully disconnected amongst each other, but that are able to communicate with each node in the first group. This assumption, however, is not valid for



(a) A case of 3 nodes (b) A case of 4 nodes

Figure 7.4: Relative positions in physical space and in signal strength space: Top – real network; Bottom – signal space network

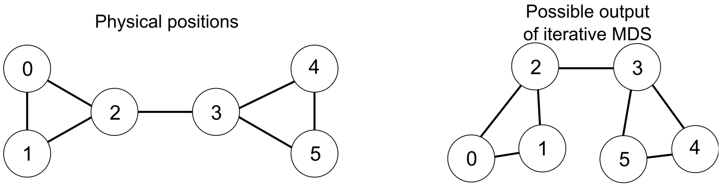


Figure 7.5: Joining weakly-connected networks using iterative methods: There is nothing pushing disconnected nodes apart resulting in a very distorted topology

mobile networks, whose dynamics are high. For these reasons, we use Classical MDS [72] plus the Floyd-Warshall [76] algorithm to estimate missing distances, because even with only 50% of links, the resulting positions are acceptable [74]. In addition, the obtained topology is more accurate than with the Map Stitching and Iterative MDS approaches (Figure 7.5).

We herein assume a connected network (despite possibly not fully linked), meaning that there exists at least one route between any pair of nodes, consequently the network is not partitioned. Let E denote a route between i and j, which contains several links, and let the pair of nodes a and b be the extremes of a generic link in E. We thus define $\mathbf{F}^n(i,j)$ according to expression (7.4), approximating the distance between two nodes that are not directly connected, with minimum accumulated *Signal Space Distance*.

$$\mathbf{F}^n(i,j) = \begin{cases} \mathbf{D}^n(i,j) & \text{if } i \text{ and } j \text{ are linked} \\ \min_E \left(\sum_{\forall (a,b) \epsilon E} \mathbf{D}^n(a,b) \right) & \text{if } i \text{ and } j \text{ are not linked} \end{cases} \tag{7.4}$$

As shown in Figure 7.6, the physical distance between two indirectly connected nodes is probably smaller than the minimum accumulated distance of a connection route. For example: $\mathtt{Dist}_{13} < \mathtt{Dist}_{12} + \mathtt{Dist}_{23}$. This introduces another source of deformation in the nodes relative positioning with respect to their physical position. However, as we said before, we are just targeting for relative position estimates for gross navigation purposes and the referred deformation should not be an obstacle to that purpose. On the other hand, this distance estimation is deterministic and easy to compute, enabling a smooth position estimation in scenarios of moving nodes. Similarly to what is explained in [74] we use the Floyd-Warshall algorithm for computing the shortest signal distance for every pair of nodes. After completion, this algorithm provides a complete matrix of pairwise signal space distances. Our distance approximation approach requires an additional time complexity of $O(n^3)$ due to the Floyd-Warshall algorithm. We consider this affordable for small teams of mobile nodes in which n is typically ten or less.

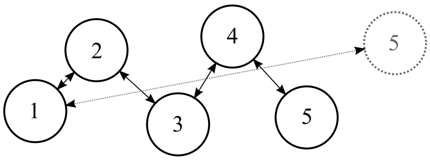


Figure 7.6: Approximating the distance for missing Signal Space Distance measurement

7.2.3.2 Creating a symmetric matrix

Since pairwise distances should be reciprocal, one of the Multidimensional Scaling technique requirements is the symmetry of the Signal Space Distance matrix. However, due to communication uncertainty, slightly different transmission power in different nodes, non-omnidirectional antennae, etc., the Signal Space Distance matrix is seldom symmetric.That being said, in order to create and feed a symmetric distance matrix to the MDS algorithm, we propose the following options:

- Use one of the triangles of the matrix, either upper or lower, and replace the other one;
- Use the mean between the upper and lower triangles of the matrix (i.e.: mean between the Signal Space Distance values of both directions in each link);
- Use the maximum Signal Space Distance value between the upper and lower triangles of the matrix (i.e.: minimum RSSI value between the two directions in each link);
- Use the minimum Signal Space Distance value between the upper and lower triangles of the matrix (i.e.: maximum RSSI value between the two directions in each link).

To the best of our knowledge, a study on the impact of such options on MDS performance has not been carried out before, thus we assess the problem in the experiments that are shown later on.

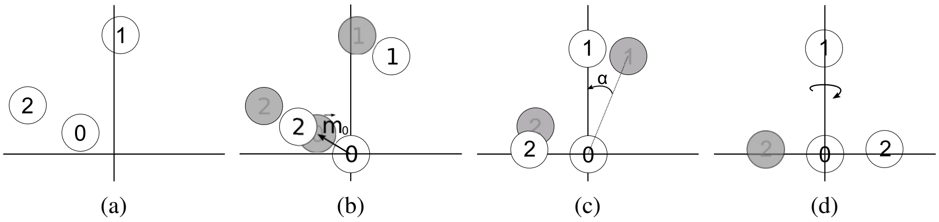
7.2.3.3 Multidimensional scaling

As previously stated, we use MDS [72] to compute the relative positions. MDS is a technique used in multivariate analysis that transfers a known $N \times N$ matrix ($\mathbf{D}$) of dissimilarities to N points of an m-dimensional Euclidean space in such a way that the pairwise distances between points are compatible with the dissimilarities matrix. Being $[\delta_{ij}]_{N \times N}$ the matrix with pairwise distances, this algorithm can be used to find an $\mathbf{M} = [m_{ij}]_{N \times m}$ matrix of approximate positions.

Our goal is to avoid using a channel model (e.g. log distance path loss model) that would require calibration based on measurements, i.e. additional parameters to be estimated. The mapping from the physical distances to Signal Space Distances is a log function, and this defines our new disparities used in the MDS [72, Ch. 9]. This is a reasonable assumption, as we are only coarsely interested in relative positions, and not in accurate physical positions of the nodes.

7.2.3.4 Adjusting the relative coordinates

So far, we discussed the relative position of a team of mobile nodes with no physical anchor. However, for the MDS algorithm, a small perturbation in the distances matrix would bring totally different results for the coordinates $\mathbf{M}$. One of the causes for such behaviour is the way MDS sorts out certain ambiguities that are inherent to the relative localisation process, e.g. eigenvector switching causes map flips. Since the nodes position is only recovered up to rigid motion, orientation of the team cannot be determined just with pair-wise distances, neither can symmetry relationships. To obtain relative position estimates that vary smoothly, we carry out the following adjustments of the coordinates provided by the MDS (considering only the result presented in 2D space, i.e. $m = 2$).



(a) (b) (c) (d)

Figure 7.7: Adjusting Coordinates: (7.7a) Positions given by MDS; (7.7b) Positions after shifting node 0 to origin; (7.7c) Positions after rotation of node 1; (7.7d) Positions after flipping node 2 to the right plane

Let $\mathbf{M} = [m_{ij}]_{N\times 2} = (\mathbf{m}_0, \mathbf{m}_1, \ldots, \mathbf{m}_{N-1})$ denote the coordinates determined with MDS (Figure 7.7a) and $\mathbf{S} = [s_{ij}]_{N\times 2} = (\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{N-1})$ denote the final coordinates (Figure 7.7d). We consider the three nodes with the smallest IDs as being local references (herein referred as 0, 1, and 2).

The coordinates adjustment includes shift (Figure 7.7b), rotation (Figure 7.7c) and reflection (Figure 7.7d) so that node 0 is at the origin point (0,0), node 1 on the positive y-axis and node 2 on the right half-plane. Thus, we first let $\forall_{0\leqslant i<N} \mathbf{t}_i = \mathbf{m}_i - \mathbf{m}_0$ as in Eq. (7.5), obtaining $\mathbf{T}$ where node 0 is in the origin. Then, we compute the clockwise angle α from vector $\mathbf{t}_1$ to y-axis, and rotate all nodes α around the origin as in Eq. (7.6) deducing the intermediate positions $\mathbf{Y}$. Finally, we check if $\mathbf{y}_2$ is on the right half-plane, i.e. if node 2 has a positive x-coordinate. If so, $\mathbf{S} = \mathbf{Y}$, else we reflect $\mathbf{Y}$ over the vertical axis as in Eq. (7.7).

Finally, we noted that this set of adjustments, despite sorting out many situations, still suffers from certain ambiguities that may, for example, lead to partially inverted topologies. The work in the next section sorts these problems by taking into account the variances of the distances.

$$\mathbf{T} = (\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{N-1}) = (\mathbf{m}_0 - \mathbf{m}_0, \mathbf{m}_1 - \mathbf{m}_0, \mathbf{m}_2 - \mathbf{m}_0, \ldots, \mathbf{m}_{N-1} - \mathbf{m}_0) \tag{7.5}$$

$$\mathbf{Y} = (\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{N-1}) = \mathbf{T} \times \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \tag{7.6}$$

$$\mathbf{S} = \begin{cases} \mathbf{Y}, & \text{if } \mathbf{y}_2 \text{is in the right plane} \\ \mathbf{Y} \times \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, & \text{otherwise} \end{cases} \tag{7.7}$$

7.3 Experimental Results on Signal Space Positioning

In this section, we describe an implementation of the proposed relative localisation algorithm from Signal Space Distance using Crossbow's MicaZ motes. MicaZ motes communicate in 2.4 GHz IEEE 802.15.4 running TinyOS 1.1.15 operating system. We used the RSSI values measured at each packet reception. The program running on all nodes was identical except for the unique node ID. For sensing data retrieval,

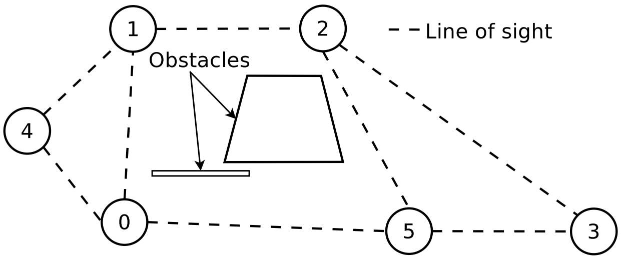


Figure 7.8: Node distribution

an arbitrary node was connected to an MIB600 board that forwarded the data to a host computer via an Ethernet TCP/IP interface.

7.3.1 Sensitivity to parameter selection

The main objective of this section is to validate the proposed RSSI-based relative localisation for supporting collaborative behaviours in small teams of mobile robots.

We assess the impact of the choice of communications period, of the use of synchronisation among the nodes, and of data sampling and selection. For this purpose, we carried out several experiments in which we set the transmit power of six crossbow MicaZ motes to -10dBm. Their typical RF range with the original antenna in an indoor lab is approximately 8-10 meters. We conducted our test in a 5m×5m area in which the RSSI reading ranged approximately between -48dBm and -10dBm and considered that $\mathtt{RSSI}_{\mathtt{max}} = \mathtt{0}$. The nodes were placed according to the diagram presented in Figure 7.8, sending a periodic transmission with the sensing data as referred in the previous sections. A Matlab program running Java methods was designed to get the sensing data from the programming board via the TCP/IP port, capturing the information to a file for offline processing. Note that since node 0 was connected to the programming board, the results represent its perspective of the network, however, every node can compute its own relative positions coordinate system. We ran four experiments, capturing 300 samples in each of them, two using a transmission period of 500ms and the other two with a period of 100ms, both with and without synchronisation. For the synchronisation algorithm we used Reconfigurable and Adaptive TDMA (RA-TDMA) as described in Chapter 4.

The offline processing uses the data collected in the several experiments. A sliding window filter was applied to the data. First, a 3 sample window was used, where all non-zero samples were taken into account. Then, a 5 samples window was used with the previously discussed rejection rule. The Kalman filter was applied to the RSSI data, and then, the positioning with classical MDS algorithm was used, complemented with the Floyd-Warshall algorithm. The methods in Section 7.2.3.2 were used to obtain a symmetrical matrix. Finally, the nodes coordinates were adjusted, as described in Section 7.2.3.4. Note, however, the processing was done offline in order to carry out the analysis. As we will show hereinafter, the relative positioning algorithm can run online.

The results presented in Figure 7.9 show the results of the position estimates from the perspective of node 0, where the dots represent single estimations, and the ellipses characterise the whole sample for each individual node. In particular, the ellipses show the standard deviation of the location errors along the principal axes. In order to simplify the visualisation of the results, the plots show the case in which a symmetric pairwise distances matrix was achieved using the minimum Signal Space Distance of the two directions in each link. Nevertheless, the results achieved with the other options in Section 7.2.3.2 were very similar.

7.3.1.1 Results on the transmission period

We considered two different periods for the nodes to broadcast their matrix, namely 500ms and 100ms. The results are showed in Figure 7.9a and 7.9b for 500ms and 7.9c and 7.9d for 100ms. The former case presents higher precision, with significantly smaller standard deviation. On the other hand, it is less reactive than the latter case, in which changes in the physical topology are reflected in the matrix and distributed among the nodes five times faster. The difference in standard deviation may be explained by a lower residual probability of collisions and interferences with longer periods, in the Adaptive-TDMA synchronisation.

7.3.1.2 Results on the use of synchronised transmissions

Despite using short messages (a 6×6 byte matrix, a 6×1 byte ageing vector, a message header, and message tail in a total of 55bytes, thus with a low medium occupancy), the experiments show that the absence of synchronisation can cause a strong degradation in the quality of the localisation. The degradation is caused by the higher probability of collision with the other nodes and interference with other transmitters in the same band, e.g., a WiFi network was operating in the same area. This effect may be noticed by comparing Figure 7.9c and 7.9e with Figure 7.9d and 7.9f, respectively, where a transmission period of 100ms was used. On the other hand, when the transmission period increases, the probability of collision reduces and the impact of synchronisation becomes less significant. This is shown in Figure 7.9a and 7.9b using a transmission period of 500ms.

7.3.1.3 Results on the use of different sample window sizes and sample selection strategies

Comparing the results in Figure 7.9c and 7.9d with those in Figure 7.9e and 7.9f, obtained by using the same transmission period, we may notice the performance in terms of standard deviation slightly better when using 5 samples, compared to the case where 3 samples were used.

7.3.1.4 Testing the use of different approaches to produce a symmetrical matrix

In our experiments none of the different approaches tested showed a significant improvement over another one. Thus, when the transmission power of all nodes is approximately equal, as well as their antennas, the differences of the two halves of the matrix are not significant. Consequently, in order to use MDS, the Signal Space Distance matrix can be considered symmetrical and, as such, the amount of transmitted data can be reduced from N^2 to $N \times (N-1)/2$, by transmitting only half of the matrix. This improvement may not seem significant since the complexity remains $O(n^2)$, however, it halves the amount of data that needs to be transmitted.

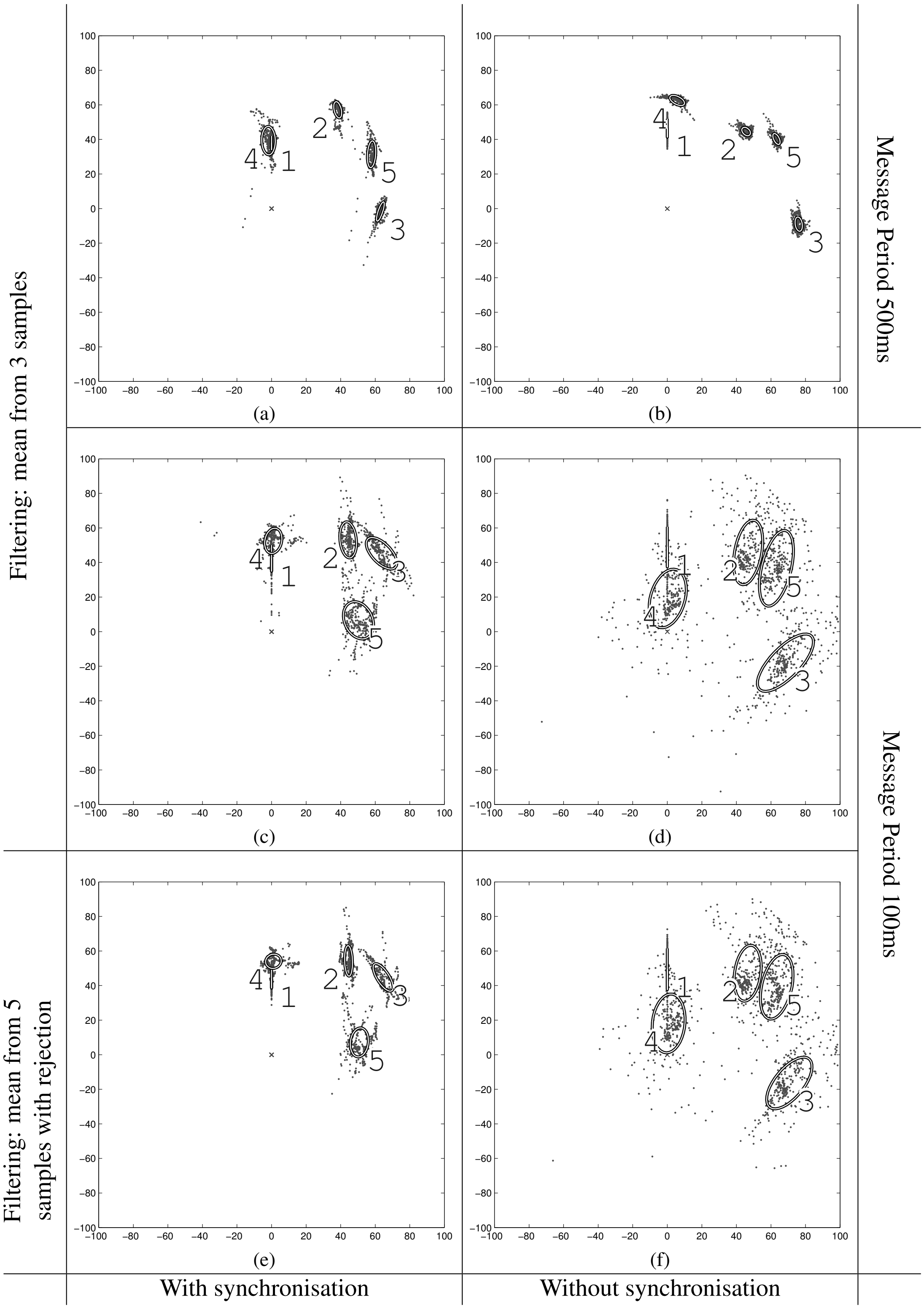


Figure 7.9: Experiments concerning MDS: Position estimates using different sliding window filters, transmission periods, with and without synchronisation

7.3.2 Conclusions

Based on our experiments, we conclude that communications cycle and synchronisation are very significant to the localisation performance. When more stable results are required, a longer transmission cycle is the best option. However, when a more reactive system is required, a shorter cycle is more adequate but then the transmissions should be synchronised to reduce collisions and maintain a low standard deviation of the values. This latter case might be more suitable for navigation purposes.

Moreover, the implementation of the sliding window filter, both in window size and data selection, seems to have some impact on the standard deviation of the measurements, albeit small. Consequently, it might be worth exploring some more sophisticated filtering methods.

Finally, if all nodes are transmitting with equal power and are equipped with similar antennas, the wireless channel is reciprocal, and therefore it is possible to transmit only half of the Signal Space Distance matrix, thus reducing the amount of transmission overhead by approximately half.

7.4 Estimating Relative Positions with Real Distances

The RSSI only relative localisation presented in the last section, as we discussed, can be used for simple scenarios where robots need to rendez-vous or follow each other. Nevertheless, if we want a more robust localisation we need to use other more accurate techniques.

In this section, we will use the hybrid ranging method presented in the previous chapter as a basis, and develop a method for estimating the relative distances between robots similar to the one just presented. The hybrid ranging method already provides us with filtered information, which will simplify some of the process, however, we will need to modify some of the blocks built for the RSSI only case.

7.4.1 Collecting distance information from robots

The work presented in Chapter 6 proposes an RF-only, anchor-less technique that performs online estimation of the distance between mobile robots without previous knowledge. We use RSSI/RT-ToF measurements to perform online estimation of the path loss model. Then, the corresponding model is used to estimate the distance using the RSSI, or the RSSI and RT-ToF when available. Moreover, an Extended Kalman Filter (EKF) performs the estimation. Consequently, the result is both the estimated distance between robots and the corresponding estimated variance (Figure 7.10). Despite that, the work in Chapter 6 approaches the problem we aim to solve up to the estimation of distances, only.

Taking advantage of this work, we build and share amongst the robots two matrices based on the output of the EKF: the matrix $\mathbf{D}_{N\times N} = d_{ij}$ (Eq. 7.8), where d_{ij} is the distance estimate between robots i and j as estimated by robot i, and the matrix $\mathbf{V}_{N\times N} = s_{ij}^{00}$ (Eq. 7.9), where s_{ij}^{00} is the variance of the distance estimate between robots i and j, as estimated by robot i.

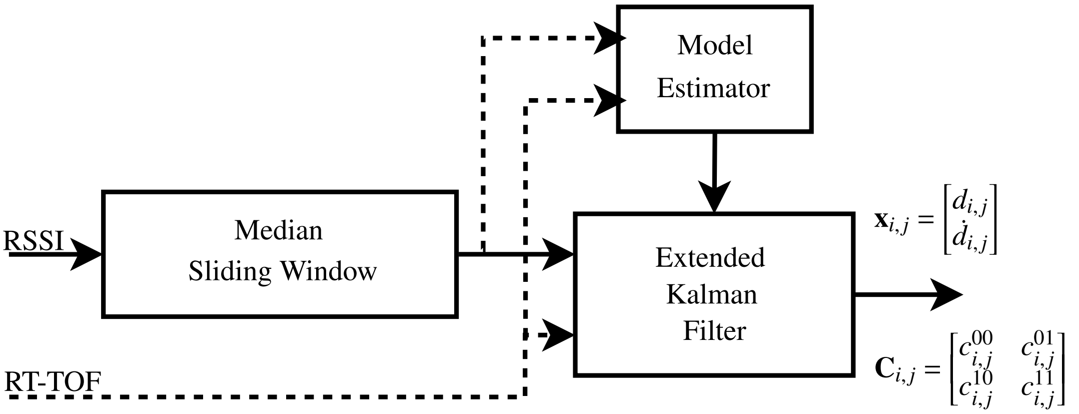

Model
Estimator
RSSI
Median
Sliding Window
Extended
Kalman
Filter
RT-TOF
$\mathbf{x}_{i,j} = \begin{bmatrix} d_{i,j} \\ \dot{d}_{i,j} \end{bmatrix}$
$\mathbf{C}_{i,j} = \begin{bmatrix} c^{00}_{i,j} & c^{01}_{i,j} \\ c^{10}_{i,j} & c^{11}_{i,j} \end{bmatrix}$

Figure 7.10: RF-based ranging for each link – Dashed lines apply only when RT-ToF data is available; In this figure, $\mathbf{x}_{i,j}$ is the estimated state (containing the distance and respective derivative) between robots i and j; $\mathbf{C}_{i,j}$ is the respective covariance matrix

$$\mathbf{D}_{N\times N} = \begin{bmatrix} 0 & d_{0,1} & d_{0,2} & \cdots & d_{0,N-1} \\ d_{1,0} & 0 & d_{1,2} & \cdots & d_{1,N-1} \\ d_{2,0} & d_{2,1} & 0 & \cdots & d_{2,N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{N-1,0} & d_{N-1,1} & d_{N-1,2} & \cdots & 0 \end{bmatrix} \tag{7.8}$$

$$\mathbf{V}_{N\times N} = \begin{bmatrix} 0 & c^{00}_{0,1} & c^{00}_{0,2} & \cdots & c^{00}_{0,N-1} \\ c^{00}_{1,0} & 0 & c^{00}_{1,2} & \cdots & c^{00}_{1,N-1} \\ c^{00}_{2,0} & c^{00}_{2,1} & 0 & \cdots & c^{00}_{2,N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c^{00}_{N-1,0} & c^{00}_{N-1,1} & c^{00}_{N-1,2} & \cdots & 0 \end{bmatrix} \tag{7.9}$$

Finally, the algorithms we will use require these matrices to be symmetrical, which seldom happens due to different transmission power in different robots, non omnidirectional antennas, etc.. That being said, in order to create and feed a symmetric distance matrix to the those algorithms, we define matrix $\mathbf{G}_{N\times N}$ according to Eq. 7.10, and matrix $\mathbf{W}_{N\times N}$ according to Eq. 7.11, which corresponds to keeping, for every symmetrical pair, the values of distance and variance that have smaller variance.

$$\mathbf{G}(i,j) = \begin{cases} \mathbf{D}(i,j), & \mathbf{V}(i,j) < \mathbf{V}(j,i) \\ \mathbf{D}(j,i), & \text{otherwise} \end{cases} \tag{7.10}$$

$$\mathbf{W}(i,j) = \begin{cases} \mathbf{V}(i,j), & \mathbf{V}(i,j) < \mathbf{V}(j,i) \\ \mathbf{V}(j,i), & \text{otherwise} \end{cases} \tag{7.11}$$

7.4.2 Estimating positions from distances

Similarly to the previous section, we use again MDS [72] to transform pairs of distances to points in a coordinate system. Thus, using the positive semi-definite matrix $\mathbf{G}_{N\times N}$ containing the pairwise distances between all robots (possibly using the Floyd-Warshall algorithm), we can write Eq. 7.12, where the MDS function returns the 2-dimensional $\mathbf{M}_{N\times 2}$ containing the positions of the robots.

$$\mathbf{M} = \mathrm{MDS}(\mathbf{G}) \tag{7.12}$$

Also, we already referred that a small perturbation in the distances matrix can bring totally different results for the coordinates, such as map flips. Since the nodes position is only recovered up to rigid motion, orientation of the team cannot be determined just with pair-wise distances, neither can symmetry relationships.

In Section 7.2, we proposed a set of transformations in order to solve these symmetry ambiguities. However, we noticed that by forcing one robot to be in the y-axis, all of its error was removed from it and inserted in the remaining network. Moreover, by assuming one of the robots on the positive x-axis would create a singularity when such robot was near the y-axis. When this happens, a small variation in its position makes the full topology flip vertically, an undesired effect for the consistency of the positions estimate.

Due to these issues, in order to obtain relative positions estimates that vary smoothly, we carry out now an improved set of adjustments of the coordinates provided by the MDS by using an arbitrary reference, e.g., the previous known topology. Then, the transformations will try to maintain a smooth variation of the angle between each positions estimate and the reference.

Let $\mathbf{M}_{N\times 2} = (\mathbf{m}_0;\mathbf{m}_1;\ldots;\mathbf{m}_{N-1})$[1] denote the coordinates determined with MDS, where $\mathbf{m}_i = (m_i^x, m_i^y)$ is the 2D position of robot i; $\mathbf{L}_{N\times 2} = (\mathbf{l}_0;\mathbf{l}_1;\ldots;\mathbf{l}_{N-1})$ denote a set of arbitrary reference positions, where $\mathbf{l}_i = (l_i^x, l_i^y)$ is the 2D position of robot i; and $\mathbf{S}_{N\times 2} = (\mathbf{s}_0;\mathbf{s}_1;\ldots;\mathbf{s}_{N-1})$ denote the final coordinates, where $\mathbf{s}_i = (s_i^x, s_i^y)$ is the 2D position of robot i.

We consider the robot making these calculations (herein referred by 0) as being in the origin. Then, because of the flip ambiguity, we generate $\mathbf{M}^I_{N\times 2} = (\mathbf{m}^I_0;\mathbf{m}^I_1;\ldots;\mathbf{m}^I_{N-1})$, Eq. 7.13, that represents the mirror image of the output of MDS along the y-axis.

$$\mathbf{M}^I = \mathbf{M}\times\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \tag{7.13}$$

In order to remove the rotation ambiguity, for each robot, we calculate the angle that would be required to align it with the reference $\phi = \mathrm{atan2}(\mathbf{L}) - \mathrm{atan2}(\mathbf{M})$. Where atan2 represents the four quadrant arctangent. Similarly, we calculate $\phi^I = \mathrm{atan2}(\mathbf{L}) - \mathrm{atan2}(\mathbf{M}^I)$. Using those two hypotheses we choose the best coordinate set according to Eq. 7.14. By selecting the set with the smallest standard deviation, we are selecting the topology with more similarities to the reference, i.e. the topology in which all robots require approximately the same rotation to match the reference. Finally, we remove the points whose residuals exceed one standard deviation (ϕ_e) from ϕ and ϕ^I, i.e. if $\mathrm{abs}(\phi_\mathrm{e} - \mathrm{mean}(\phi)) > \mathrm{std}(\phi)$ or if

---

[1] Similarly to Matlab scripting language we use ';' to separate rows in vectors and matrices

$\mathrm{abs}(\phi_e - \mathrm{mean}(\phi^I)) > \mathrm{std}(\phi^I)$, and we calculate α, the clockwise rotation angle that minimises the square error of the angle between the estimate and the reference (Eq. 7.15).

$$\mathbf{T} = \begin{cases} \mathbf{M}, & \mathrm{std}(\phi) < \mathrm{std}(\phi^I) \\ \mathbf{M}^I, & \text{otherwise} \end{cases} \tag{7.14}$$

$$\alpha = \begin{cases} \min\left[(\phi - \alpha)\cdot(\phi - \alpha)\right], & \mathrm{std}(\phi) < \mathrm{std}(\phi^I) \\ \min\left[\left(\phi^I - \alpha\right)\cdot\left(\phi^I - \alpha\right)\right], & \text{otherwise} \end{cases} \tag{7.15}$$

The last step to calculate the final coordinate **S**, is rotating the selected topology as in Eq. 7.16. Note that ϕ or ϕ^I can contain angles similar in rotation but different in value, i.e., any values separated by 2π. Therefore, we analyse the residuals both between $[-\pi, \pi]$ and $[0, 2\pi]$, choosing the one with less standard deviation.

$$\mathbf{S} = \mathbf{T} \times \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \tag{7.16}$$

7.4.3 Filtering positions with a Kalman filter

To further enhance the relative positions of a team of robots, we implemented a Kalman filter (KF) with a state vector given in Eq. 7.17, where $\mathbf{P}_{2N\times 1}$ is the state vector, and $\mathbf{p}_i = (p_i^x, p_i^y)$ is the estimated 2D position of robot i. The equations of the state space model are provided below.

$$\mathbf{P}_{2N\times 1} = \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \mathbf{p}_2 & \cdots \mathbf{p}_{N-1} \end{bmatrix}' \tag{7.17}$$

The prediction equation is Eq. 7.18, where $\omega(k) \sim \mathcal{N}(0, \mathbf{R})$ is the process noise at instant k. Finally, we measure the state **P** directly by sampling the output of MDS, Eq. 7.19 where, $\overline{\mathbf{P}}(k)$ is the measurement and $\nu(k) \sim \mathcal{N}(0, \mathbf{Q})$ is the measurement noise.

$$\mathbf{P}(k) = \mathbf{P}(k-1) + \omega(k) \tag{7.18}$$

$$\overline{\mathbf{P}}(k) = \mathbf{P}(k) + \nu(k) \tag{7.19}$$

7.4.3.1 Prediction step

In order to estimate the process noise ω, we apply an heuristic inspired on the technique used in [18] to estimate relative velocities. However, instead of using the speed between robots, we use the estimated variances of the distances between robots, Eq. 7.21, allowing us to use the uncertainty of the distance measurements as a measurement of the state progression. In detail, we use the estimated variances of the distances between robots (**W**), and compute a unit vector $\mathbf{u}_{\mathrm{i,j}}$, Eq. 7.20, pointing from the position of the robot i to the position of robot j. Then, the state variance of robot i ($\mathbf{r}_i$) is calculated as the sum of the absolute value of those vectors projected in the x and y axes, where $\mathbf{r}_i = (r_i^x, r_i^y)$. Finally, we divide the velocity by N, to remove the multiple inclusions of the same variance, obtaining the diagonal matrix $\mathbf{R}$, Eq.7.22.

$$\mathbf{u}_{\mathrm{i,j}} = (\mathbf{u}^x_{\mathrm{i,j}}, \mathbf{u}^y_{\mathrm{i,j}}) = \left(\mathbf{p}_i(k-1) - \mathbf{p}_j(k-1)\right) / \left|\mathbf{p}_i(k-1) - \mathbf{p}_j(k-1)\right| \tag{7.20}$$

$$\mathbf{r}_i = \frac{1}{N} \times \sum_{j=0..N-1,\, j \neq i} \mathbf{W}(i,j) \cdot \left(\left|\mathbf{u}^x_{\mathrm{i,j}}\right|, \left|\mathbf{u}^y_{\mathrm{i,j}}\right|\right) \tag{7.21}$$

$$\mathbf{R}_{n \times n} = \begin{bmatrix} r^x_0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & r^y_0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & r^x_{N-1} & 0 \\ 0 & 0 & 0 & \cdots & 0 & r^y_{N-1} \end{bmatrix} \tag{7.22}$$

7.4.3.2 Integrating measurements

Let us define the function K as the function that transforms the output of MDS into a vector compatible with the state $\mathbf{P}_{2N \times 1}$ (Eq. 7.23), and K^{-1} the inverse operation.

$$K\left(\begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \\ \vdots & \vdots \\ x_{N-1} & y_{N-1} \end{bmatrix}\right) = \begin{bmatrix} x_0 \\ y_0 \\ x_1 \\ y_1 \\ \vdots \\ x_{N-1} \\ y_{N-1} \end{bmatrix} \tag{7.23}$$

As we described before, our measurement ($\overline{\mathbf{P}}(k)$) is given by using the MDS algorithm, where we obtain $\mathbf{S}$. During this calculation, during the rigid transformation phase we adjust the robots positions to minimise the error with relation to the current state estimate (i.e. $\mathbf{L} = \mathbf{P}(k)$).

Then, in order to compute the process noise we use a Monte Carlo (MC) approach, according to which we add random noise to the distance inputs, and repeatedly calculate the output, thus sampling the localisation function. With a sufficient number of runs we can determine the impact of the noise on the output distribution. This is done by executing the MDS algorithm q times, Eq. 7.25, where q can be configured according to the precision required and the computational power available. For each of those executions we use as input $\mathbf{S} + a \times \mathbf{H}$, where $a \sim \mathcal{N}(0,1)$, and $\mathbf{H}$ is the matrix of standard deviations, obtained from the element-wise square root of $\mathbf{W}$. However, this time after each MDS calculation and during the rigid transformation phase, we adjust the robots positions to minimise the error with relation to the measurement (i.e. $\mathbf{L} = \overline{\mathbf{P}}(k)$). Finally we obtain the covariance matrix $\mathbf{Q} = \mathrm{cov}(\mathbf{M}_z)$, where $z = [1..q]$.

$$\mathbf{H} = \begin{bmatrix} \sqrt{w_{0,0}} & \sqrt{w_{0,1}} & \sqrt{w_{0,2}} & \cdots & \sqrt{w_{0,N-1}} \\ \sqrt{w_{0,1}} & \sqrt{w_{1,1}} & \sqrt{w_{1,2}} & \cdots & \sqrt{w_{1,N-1}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sqrt{w_{N-1,0}} & \sqrt{w_{N-1,1}} & \sqrt{w_{N-1,2}} & \cdots & \sqrt{w_{N-1,N-1}} \end{bmatrix} \tag{7.24}$$

$$\mathbf{M}_z = K(\mathrm{MDS}(\mathbf{S} + a \times \mathbf{H})) \tag{7.25}$$

7.5 Simulation Results on Real Distances Positioning

In this section, we first describe the simulation setup, Figure 7.11, namely how we generate measurements and how we share the distance estimates between the robots. Then we present the results we obtained using our simulator on ground-truth (GT) collected from the CyberRescue@RTSS2009 competition [121].

7.5.1 Generating RSSI and RT-TOF measurements

In order to realistically simulate our proposal, we modelled the sensors measurements from the real experiments performed in [17]. Namely, we generate RSSI values taking into account the hardware 2dBm resolution, Eq. 7.26, where $\rho(d)$ is the medium propagation using the model in Eq. 7.27. For the RSSI the parameters are: $\sigma_\rho^2 = 20$, $\rho_0 = -39.6955$, $\alpha = 1.1558$, and $a_\rho \sim \mathcal{N}(0,1)$. In addition, we also generate RT-ToF measurements from real distance according to Eq. 7.28. In this case $a_d \sim \mathcal{N}(0.3842, 1)$ and $\sigma_d^2 = 0.4$.

$$\bar{\rho} = -2 \times \mathrm{round}\Big(0 \leqslant -\Big(\rho(d) + a_\rho \times \sqrt{\sigma_\rho^2} + 35\Big)/2 \leqslant 31\Big) - 35 \tag{7.26}$$

$$\rho(d) = \rho_0 - 10\alpha \log_{10}(d) \tag{7.27}$$

$$\bar{d}(d) = d + a_d \times \sqrt{\sigma_d^2} \tag{7.28}$$

7.5.2 Simulating the communications

In order to simulate the delays of the information reaching a robot, we have devised a simple communication protocol. A TDMA schedule with slot size of 50ms was created such that each robot i transmits in slot i, therefore, at each slot the simulator will run the loop in Figure7.11 for each robot. A round robin ranging schedule was implemented such that every i^{th} slot, robot i tries to range one of the other robots, then, in the next i^{th} slot it will try to range the next robot, and so on. Still in the same slot, robot i will broadcast a message containing its distance estimates and distance variance, which the other robots

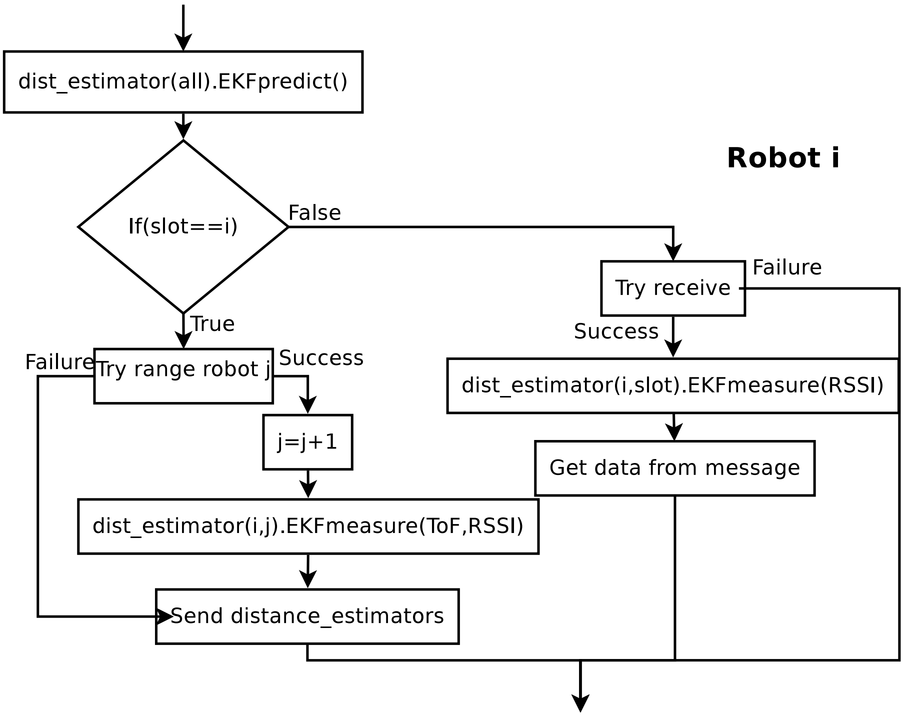

dist_estimator(all).EKFpredict()
Robot i
If(slot==i)
False
True
Try receive
Failure
Success
Failure
Try range robot j
Success
dist_estimator(i,slot).EKFmeasure(RSSI)
j=j+1
Get data from message
dist_estimator(i,j).EKFmeasure(ToF,RSSI)
Send distance_estimators

Figure 7.11: Simulation loop for robot *i* – Predict all Kalman filters; If robot is a receiver, try to receive from sender and measure RSSI; If robot is the sender, try to range robot *j* and measure RT-ToF and RSSI, finally estimate positions

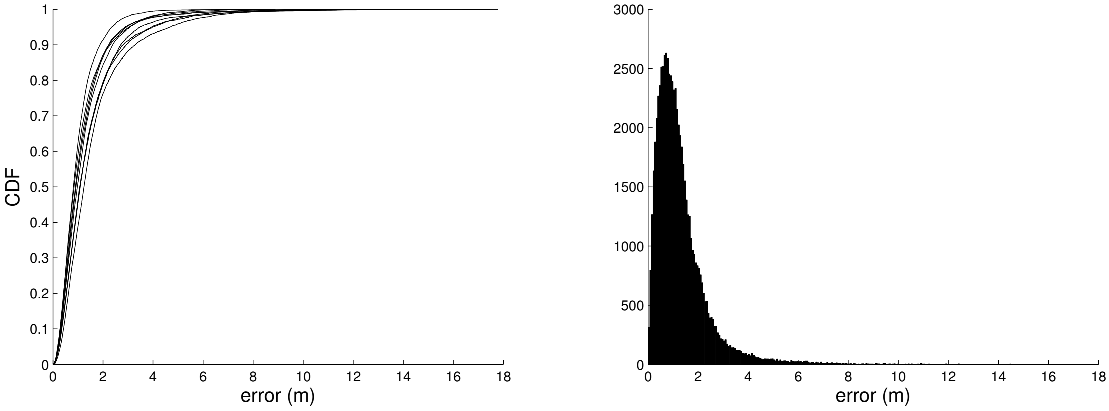
will try to receive. The communications were programmed with a probability of success of 94% from measurements carried out by [15] in real conditions similar to our simulation, i.e. TDMA rounds with no external interference.

7.5.3 Results

In order to experimentally validate our proposal, we used logs collected in a simulated robots competition (CyberRescue@RTSS2009 [121]) using the Cyber-Physical Systems Simulator (CPSS) [122]. We used the GT positions collected in the 9 logs available on the website, as the path the robots travel through. Each log is composed of five robots moving in a 28 m by 14 m arena. For the purpose of this work, the walls were not considered for non-line of sight and reflection effects.

We ran the robots through our simulator, using the paths obtained from the logs to generate simulated measurements. The simulator output was used to run our positions estimator from the perspective of robot 1, and we set to 10 the number of MDS executions required for the measurement of the topology in the Monte Carlo approach. To be able to compare our generated positions to the log GT, we applied the same transformation techniques to the GT data using the positions estimate as the reference, placing them in the same reference frame as our data. Therefore, from this point on, when we mention GT, we are referring to the transformed GT.

In order to measure the accuracy of our proposal we calculated the error of the estimate, as the difference between the estimated position and the GT position. An histogram of such error is presented



(a) Cumulative distribution function of the error

(b) Histogram of the error

Figure 7.12: Error of the position estimates – 90% of the errors are under 5 m for all cases and below 2 m in the best case, with an overall average of 1.3 m

in Figure 7.12a, where we can see that under realistic simulated measurements, 90% of the errors are under 5 m for all the cases and under 2 m for the best case, with an overall average at 1.3 m

Nevertheless, the information we were more interested in is the confidence measurement. To analyse the confidence we calculate the percentage of GT positions contained inside different scales of the covariance ellipse. Figure 7.13 depicts a snapshot of one of the simulations where we can see the top left robot inside of the 3-standard deviation ellipse, the center left robot just on the 3-standard deviation ellipse, the bottom left robot inside of the 1-standard deviation ellipse, and the right robot inside of the 2-standard deviation ellipse. The total percentage of GT inside different scales of the covariance ellipse, from 1 to 10 times the standard deviation, is presented in Figure 7.14, for all 9 simulations. We can see that for 3 times the standard deviation, over 80% of the GT fall inside the covariance ellipse. For 5 times, this value raises to about 95%.

7.5.4 Conclusions

In this section we validated experimentally through realistic simulations the technique that uses RF-based range estimates, only, to track the relative positions of a fully mobile team of robots providing reliability information as the covariance of the positions. We believe that this is one of the first works to propose confidence values to position estimates obtained through MDS, particularly for relative localisation purposes without any anchors. Using RF-only information, our method could determine the relative localisation of a team of 5 freely moving robots with an average error of 1.3m in a region of 28×14m. Results show that our approach can consistently provide similar performance across different experiments.

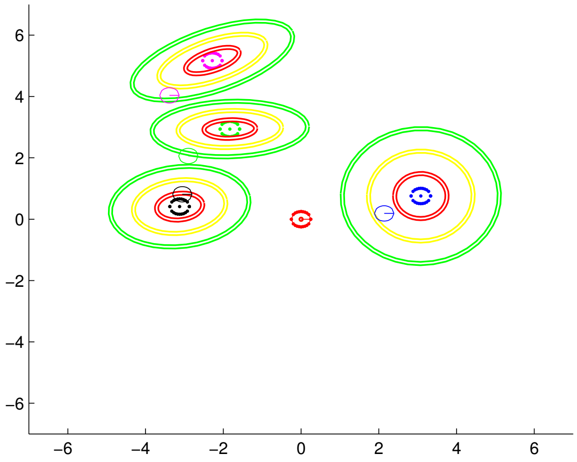


Figure 7.13: A zoomed in snapshot of the simulation field – full line robots represent the ground-truth, dotted robots represent the estimations surrounded by the 1-standard deviation ellipse(red), by the 2-standard deviation ellipse(yellow), by the 3-standard deviation ellipse(green)

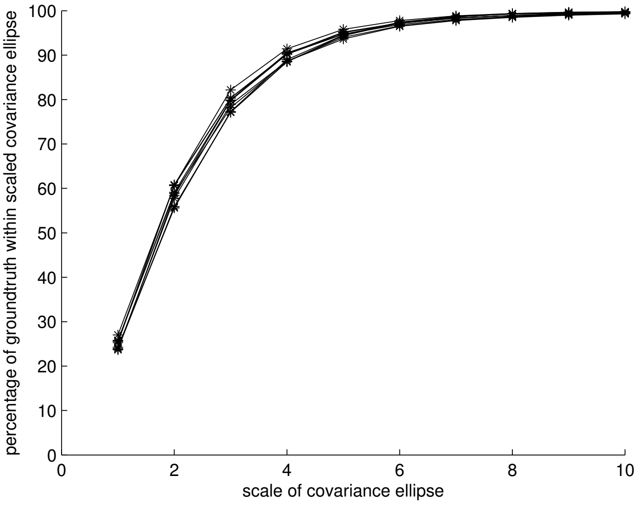


Figure 7.14: Percentage of GT positions contained inside of the covariance ellipses at different scales, from 1 to 10 times the standard deviation – For 3 times the standard deviation, over 80% of the times, the GT falls inside the covariance ellipse

Chapter 8

Conclusions

Teams of cooperating robots are becoming a hot topic as individual robots become cheaper and more accessible, and the range of potential applications increases. A great example is area coverage applications, specially using flying drones that have been consistently decreasing in price. In this type of applications, the information collected must be delivered in order to be useful, consequently they have to be able to communicate, many times using team members as intermediate hops in order to reach non neighbour robots. In addition, continuous operation is a typical requirement, therefore robots need to be continuously removed and reinserted as batteries run out. Another example is surveillance applications, where many cheap sensing robots are deployed to detect anomalies, and on an alarm triggering summon a small number of expensive actuating robots to intervene or even a human agent. What is common to these, but also to many other, multi-robot applications is the requirement of communicating data, and of knowing the team members relative location. This is where this work fits.

8.1 Thesis Validation

By providing the basic tools for communicating with and localising team members we enable many collaborative behaviours, all this using only the wireless transceiver. In particular, we validated successfully the three claims of our thesis.

Firstly, we showed that organising the team transmissions in time with a Time Division Multiple Access (TDMA) framework benefits the quality of the channel in several situations, thus contributing to enhance the performance of the applications that run on top.

Secondly, we showed that the forwarding delay in multi-hop TDMA frameworks can be substantially reduced, specially its variability, by isolating the transmission slots along the network so that traffic generated by one node can only be forwarded within its own slot.

Finally, we showed that the RF interfaces and nodes common transmissions can be used to derive relative localisation of team members in diverse ways, with more or less precision, depending on the transceivers and computing power available.

8.2 Summary of Contributions

The main contributions of this work were an ad-hoc wireless communication protocol, namely the Ad-hoc Reconfigurable and Adaptive TDMA (Ad-hoc RA-TDMA), and the relative localisation service, each of these entailing several focused contributions to the respective state of the art.

The former is an ad-hoc wireless communication protocol that by organising the robot communications in a TDMA round is able to reduce the message failure rates under certain operational conditions. The protocol is reconfigurable and adaptive, meaning that it is able to reconfigure the slot structure upon joining or leaving of team members, and adapt the round phase to transmission delays imposed by different sources (e.g. clock drifts or external traffic) while maintaining the message exchange synchronisation. Most notably, this is done without global clock synchronisation or explicit external triggers. In addition to that, the protocol supports point-to-point multi-hop communication with full slot isolation across the network. This is done by keeping the multi-hop messages strictly inside their originator's slots, implementing an in-slot forward. This feature drastically reduces the variability of the network induced delays and, in some cases, the delays themselves. In other words, this forwarding method maintains the predictability and composability properties of TDMA protocols even in multi-hop contexts.

The latter contribution, covers two different approaches to RF-based ranging, a coarse RSSI-based ranging approach, and an hybrid approach using both RSSI and Round-trip Time of Flight to estimate the path loss model, and obtain a higher ranging precision. Using these approaches, we collected pairwise distances between every pair of robots and used Multidimensional Scaling to obtain relative position information. Due to the intrinsic limitation of model-less RSSI ranging, the first ranging approach can only provide very coarse representation of the robots positions, nevertheless, we claim that this coarse representation is enough to perform basic coordination of the nodes, such as keeping connectivity within the network. The hybrid approach, on the other hand was able to provide significantly more useful information, providing estimates with a direct physical mapping. Moreover, in this latter case and in addition to the positions estimate, we provide an estimate of the covariance of the positions, allowing the user to define confidence regions around the estimate. This becomes more relevant if this information is meant to be further fused with other sensor data.

8.3 Future Work

Beyond the results presented in this thesis, the work conducted unveiled various interesting research ideas that we believe are worth further exploration. In this final section we will refer some of those ideas.

8.3.1 Proving synchronising of packet exchanges

In Chapter 4 we proposed a protocol that synchronises packet exchanges amongst robots cooperating within a team. In this scope, we proved that our protocol is able to reach synchronisation by assuming some starting conditions. But our solution for when those assumptions are dropped, were only validated by simulation. With respect to this, it would be interesting to provide a formal proof showing that for every possible starting condition our protocol is able to synchronise.

Once this is proved, it would be very interesting to analyse the synchronisation protocol in order to provide some fundamental limits of time required to converge from any initial conditions. The topology is a very fundamental part for this analysis, therefore, using a tree to limit synchronisation can have quite an impact on this. Leading us to a very interesting problem, simultaneously creating a tree to guarantee synchronisation and minimise the time the robots need to synchronise.

Finally, our heuristic change of synchronisation mode, i.e. with/without a tree, is quite pessimistic. This pessimism comes from the fact that summing all the Arcs from all robots, assumes they don't intersect, which is not true in most cases. Consequently, the robots will be using a tree to limit synchronisation for a period of time larger than what is required, therefore it becomes critical that a better analysis, most likely taking into account the topology, can be developed in order to reduce this pessimism, and thus, improving the convergence speed of the protocol.

8.3.2 Improving the analysis for immediate multi-hop forwarding

On the multi-hop forwarding techniques we presented in Chapter 5, we presented an forwarding technique that tries to minimise the end to end delay of communications within TDMA schedules. The Inserted Idle Time (IIT) analysis introduces some undesired pessimism, therefore, it could be worth developing different analysis tools, specifically one that reduces the pessimism introduced.

In addition, exploring spatial re-utilisation in multi-hop communications can dramatically increase available bandwidth in chain formations. For example, using our approach in a 10 robot chain implies that a packet can interfere with another one for 9 hops. Conversely, by adapting our protocol to allow re-utilisation the interference can be reduced for as much as 3 hops. Therefore, we believe that the adaptation of our protocol to re-utilise the wireless medium is a research line worth investigating.

8.3.3 The impact of dynamic topologies on multi-hop communications

When robots move and topology changes, links are created and destroyed, changing the amount of hops between robots. As a consequence, the packet bandwidth requirements change with it, and it is possible that the packets are no longer schedulable. As a future line of work, it would be useful to explicitly handle these situations by including them in the analysis and, for example, support different levels of service with different network requirements. However, it is important to be able to run the schedulability tests online, therefore a careful examination of the amount of time it requires to be executed.

8.3.4 Explore the possible benefits of the non-isolated multi-hop per slot approach

The non-isolated *multi-hop per slot* analysis is not a trivial problem and it has many problems associated with it, namely all packets suffer interference from multiple sources, the operations of adding robots and changing the topology, have impact beyond the reduction of bandwidth and the change in the number of hops. Nonetheless, having the ability of reusing other slots to transmit packets, can have a positive impact on the end-to-end delay, on average. For this reason we believe this topic to be worth or further research, specifically on the possibility of changing between both the isolated and the non-isolated approaches dynamically.

8.3.5 Integration of the relative position estimator

In this thesis, we focused on using RF-only localisation techniques. Nevertheless, when using more capable robots, it could be interesting to explore the benefits of such system on vision and odometry based localisation. On one hand, we know that RF-based techniques provide less precise localisation performance, on the other hand, vision based techniques, when employed in symmetrical locations (for example the RoboCup fields) have many ambiguities that can probably be solved with the relative positions of robots.

In fact, a spin-off of this work is already in progress in [123], where a different technique for solving ambiguities in MDS using the robots inertial sensors are explored.

Appendix A

Appendix A - Middleware

A middleware is a software layer that abstracts the hardware/software platform that supports an application providing appropriate programming abstractions. According to [124], there are many services that a middleware for robotic applications should provide, namely an Application Programming Interface (API) with support for communication and interoperability, high-level abstractions to facilitate the development of collaboration mechanisms, heterogeneity abstractions hiding the complexity of the low-level communications, integration mechanisms with other systems (such as wireless sensor networks), automatic resource discovery and configuration, and other often-needed robot services to avoid rewriting code. Moreover, as the robots become more dynamic and used in teams, we need a middleware that addresses the requirements of MANETs

Therefore, as it is shown in [125], the middleware must support mobility and network topology changes and provide context aware QoS. In addition, since it becomes more difficult to keep the shared data coherent both in time and space, tighter real-time constraints must be met to cope with fast changing information. Therefore, the middleware must allow fast and transparent access to both local and remote data/services, be able to maintain information on the temporal validity of the data, and take into account where the data comes from and which robot provides the service.

As the authors reason in [125], designing a middleware that fully meets all the requirements and challenges of a mobile ad-hoc environment is to some extent not a realistic venture. Therefore, a more realistic approach is to provide a general middleware that provides most of the functionality while remaining flexible enough to allow the creation and addition of modules suited for each particular application domain. For example, it is possible to divide the requirements in two different categories: low-level core services representing all services fundamental for the creation and execution of applications and high-level modules representing application specific modules that can be added by the user. The strictly fundamental features a middleware should provide are the API, the automatic resource discovery and configuration, and the support for communication and interoperability between potentially heterogeneous systems and possibly with support for QoS.

In this annex we propose using the RTDB middleware in ad-hoc robotic MANETs, despite its currently typical implementation in structured networks. We briefly describe the middleware and we discuss some of the issue that could be interesting to address in this new context.

A.1 The RTDB

Distribution based on shared memory (DSM) [126, 127] is a model of distribution where data is stored in shared memory that has been implemented using several schemes. In the centralised model, such as the blackboard in [128], data generated internally is written to a database and can then be read by others. However this scheme is a client/server interaction model with a bottleneck in the request of data, which is not a good model for ad-hoc mobile networks. However, there is another scheme, called the distributed shared memory model, which can implement the publisher/subscriber model of interaction by periodically broadcasting the shared data throughout the network. An implementation of this model, which is very popular in the RoboCup Middle Size League, is the Real-Time Database (RTDB) [129, 43, 83, 130]. With this middleware, each team member has a local image of the information provided by the other team members (data proxies) which the programmer can access explicitly, i.e. the programmer explicitly accesses the data published by a specific robot. To replicate the data amongst robots the RTDB disseminates the shared variables using a periodic broadcast, in which each robot sends the data it produces (thus implementing a special case of the publisher/subscriber model, where all robots subscribe all shared data). Its main advantages are that the data is exchanged in the background, being available to each robot as local variables, and that it is optimised for low overhead using a priori team knowledge (see Figure A.1). Also, since there is not an explicit communication request, the transmission time is not included in the processing time of the application code. In addition, since this implementation shares states, not events, the sporadic omission of a message is not a major setback to performance. However, the original version assumes that the network is fully-linked. This assumption, albeit valid for managed networks, can not be made for ad-hoc networks.

A.2 Extending the RTDB to Ad-Hoc Scenarios

The main issue with instantiating the RTDB on an ad-hoc network is the propagation of the original data so to keep all local proxies updated.

This regular dissemination has to be handled in a broadcast manner, similarly to the original RTDB. Else, the number of unicast connections would grow exponentially with the number of robots in the team. The issue, then, is to find an adequate mechanism for supporting the data dissemination throughout the network in an epidemic style.

Naturally, our proposal relies on using the Ad-hoc RA-TDMA protocol developed in this thesis. This protocol already does topology tracking by constantly disseminating connectivity information throughout the network. We propose to use exactly the same mechanism for disseminating the RTDB. This way, the RTDB would work in ad-hoc networks using the Ad-hoc RA-TDMA, similarly to the original RTDB that normally runs on the infra-structured RA-TDMA.

Nevertheless, this obvious and trivial choice still needs some research to support it, particularly concerning the relationship between the dynamics of the networks, and consequently of the data dissemination process, and the dynamics of the data itself. Such research would allow a user to define limits to the robots mobility that would enforce a desired degree of consistency across the RTDB local proxies.

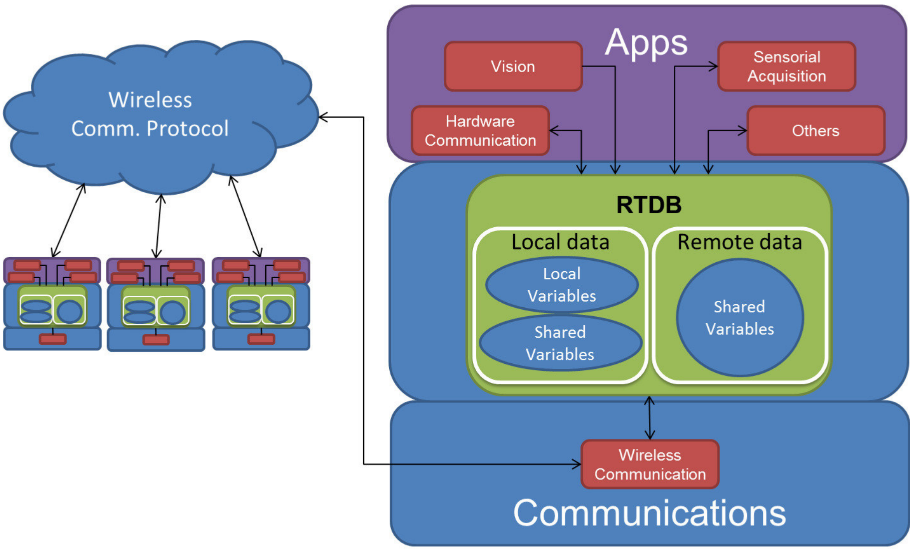

Wireless
Comm. Protocol
Apps
Vision
Sensorial Acquisition
Hardware Communication
Others
RTDB
Local data
Local Variables
Shared Variables
Remote data
Shared Variables
Wireless Communication
Communications

Figure A.1: RTDB – Each robot run an RTDB where it stores its local and shared variables. The shared data is sent through the wireless medium and replicated on all other robots

Appendix B

Appendix B - Navigation in RSS fields

Teams of mobile robots with specific sensors, actuators or other instrumentation, can provide a valuable help in covering larger areas or objects, complementing the information that can be obtained with networks of fixed sensors. In these cases, the robots have to navigate by using aids for that purpose – from landmarks, to GPS (Global Positioning System). However, landmarks are frequently undesired either because they cannot be deployed or they interfere with the environment, or are not resistant and fade out or can even be removed maliciously. Similarly, GPS has reception problems in in-door environments.

In such situations, relative navigation becomes more attractive. For example, the team of robots can assume a formation that is more convenient to provide longer reach or wider area coverage while maintaining connectivity, just by knowing how they are positioned with respect to each other. Another very interesting example to take advantage of the pre-installation of wireless beacons, that can be nodes of a wireless sensor network for example, and make robots navigate through a specific sequence of beacons in order to visit certain points in space and perform an approximate desired trajectory, e.g. carrying advertisements or performance surveillance.

These approaches can be well supported by the localisation techniques developed in this thesis, which are based on RF-ranging. Be it using RSSI, only, or a hybrid RSSI/RT-ToF system, these techniques allow detecting and navigating on signal strength and packet drop rate gradients. For example, strong negative gradients may indicate the vicinity of a signal well caused by multi-path fading, sudden increases in packet drops may indicate the edge of a reliable link, increasing gradients typically indicate the source direction.

Several works were already developed in our research lab in these directions [131, 132, 133, 134, 135, 136]. One interesting observation is that the success of navigation in RSS fields requires a judicious use of mobility to create significant signal differences and several samples, so that local signal artefacts can be filtered out.

Similarly, the work in [18] also shows how the RSSI-based relative localisation system can be used to deduce relative motion information, particularly relative heading, using time differences, which is crucial for navigation purposes.

Bibliography

[1] A. Ahmad and P. U. Lima, "Multi-robot cooperative object tracking based on particle filters." in *ECMR*, 2011, pp. 37–42.

[2] Y. Mostofi, "Compressive cooperative sensing and mapping in mobile networks," *Mobile Computing, IEEE Transactions on*, vol. 10, no. 12, pp. 1769 –1784, dec. 2011.

[3] T. Bailey, M. Bryson, H. Mu, J. Vial, L. McCalman, and H. Durrant-Whyte, "Decentralised cooperative localisation for heterogeneous teams of mobile robots," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, may 2011, pp. 2859 –2865.

[4] P. U. Lima, P. Santos, R. Oliveira, A. Ahmad, and J. Santos, "Cooperative localization based on visually shared objects," in *RoboCup 2010: Robot Soccer World Cup XIV*, ser. Lecture Notes in Computer Science, J. Ruiz-del Solar, E. Chown, and P. Plöger, Eds. Springer Berlin Heidelberg, 2011, vol. 6556, pp. 350–361.

[5] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Autonomous Robots*, vol. 30, pp. 73–86, 2011.

[6] T. Balch and R. Arkin, "Communication in reactive multiagent robotic systems," *Autonomous Robots*, vol. 1, pp. 27–52, 1994.

[7] A. Knoll and R. Prasad, "Wireless robotics: A highly promising case for standardization," *Wireless Personal Communications*, vol. 64, pp. 611–617, 2012.

[8] A. Willig, K. Matheus, and A. Wolisz, "Wireless technology in industrial networks," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1130 –1151, june 2005.

[9] J. Wu and I. Stojmenovic, "Ad hoc networks," *Computer*, vol. 37, no. 2, pp. 29 – 31, feb 2004.

[10] R. R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: the next computing revolution," in *Proceedings of the 47th Design Automation Conference*, ser. DAC '10. New York, NY, USA: ACM, 2010, pp. 731–736.

[11] F. Santos, L. Almeida, and L. Lopes, "Self-configuration of an adaptive TDMA wireless communication protocol for teams of mobile robots," in *ETFA - IEEE Int. Conference on Emerging Technologies and Factory Automation*, 2008, pp. 1197–1204.

[12] R. Costa, P. Portugal, F. Vasques, and R. Moraes, "A TDMA-based mechanism for real-time communication in IEEE 802.11e networks," in *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, sept. 2010, pp. 1 –9.

[13] G. Zhou, Q. Li, J. Li, Y. Wu, S. Lin, J. Lu, C.-Y. Wan, M. D. Yarvis, and J. A. Stankovic, "Adaptive and radio-agnostic QoS for body sensor networks," *ACM Trans. Embed. Comput. Syst.*, vol. 10, no. 4, pp. 48:1–48:34, Nov. 2011.

[14] F. Santos, L. Almeida, L. S. Lopes, J. L. Azevedo, and M. B. Cunha, "Robocup 2009," J. Baltes, M. G. Lagoudakis, T. Naruse, and S. S. Ghidary, Eds. Berlin, Heidelberg: Springer-Verlag, 2010, ch. Communicating among robots in the robocup middle-size league, pp. 320–331.

[15] L. Oliveira, L. Almeida, and F. Santos, "A loose synchronisation protocol for managing rf ranging in mobile ad-hoc networks," in *RoboCup 2011: Robot Soccer World Cup XV*, ser. Lecture Notes in Computer Science, T. Röfer, N. Mayer, J. Savage, and U. Saranlı, Eds. Springer Berlin Heidelberg, 2012, vol. 7416, pp. 574–585.

[16] L. Oliveira, L. Almeida, and P. Lima, "Multi-hop routing within TDMA slots for teams of cooperating robots," in *IEEE World Conference on Factory Communication Systems - WFCS*, May 2015.

[17] L. Oliveira, C. Di Franco, T. Abrudan, and L. Almeida, "Fusing time-of-flight and received signal strength for adaptive radio-frequency ranging," in *Advanced Robotics (ICAR), 2013 16th International Conference on*, Nov 2013, pp. 1–6.

[18] L. Oliveira, H. Li, L. Almeida, and T. E. Abrudan, "RSSI-based relative localisation for mobile robots," *Ad Hoc Networks*, vol. 13, Part B, no. 0, pp. 321 – 335, 2014.

[19] L. Oliveira and L. Almeida, "RF-based relative position estimation in mobile ad-hoc networks with confidence regions," in *RoboCup 2014: Robot World Cup XVIII.* Springer, 2015, pp. 383–394.

[20] "IEEE standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pp. 1–2793, March 2012.

[21] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 3, pp. 535–547, 2000.

[22] Q. Ni, "Performance analysis and enhancements for IEEE 802.11 e wireless networks," *Network, IEEE*, vol. 19, no. 4, pp. 21–27, 2005.

[23] K. Xu, M. Gerla, and S. Bae, "How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks," in *Global Telecommunications Conference, 2002. GLOBECOM'02. IEEE*, vol. 1. IEEE, 2002, pp. 72–76.

[24] "IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirement part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs)," *IEEE Std 802.15.4a-2007 (Amendment to IEEE Std 802.15.4-2006)*, pp. 1–203, 2007.

[25] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *Networking, IEEE/ACM Transactions on*, vol. 12, no. 3, pp. 493 – 506, june 2004.

[26] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, ser. SenSys '03. New York, NY, USA: ACM, 2003, pp. 171–180.

[27] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, ser. SenSys '04. New York, NY, USA: ACM, 2004, pp. 95–107.

[28] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proceedings of the 4th international conference on Embedded networked sensor systems*, ser. SenSys '06. New York, NY, USA: ACM, 2006, pp. 307–320.

[29] Y. Sun, O. Gurewitz, and D. B. Johnson, "RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, ser. SenSys '08. New York, NY, USA: ACM, 2008, pp. 1–14.

[30] P. Huang, C. Wang, L. Xiao, and H. Chen, "RC-MAC: A receiver-centric medium access control protocol for wireless sensor networks," in *Quality of Service (IWQoS), 2010 18th International Workshop on*, june 2010, pp. 1 –9.

[31] G.-S. Ahn, S. G. Hong, E. Miluzzo, A. T. Campbell, and F. Cuomo, "Funneling-mac: a localized, sink-oriented mac for boosting fidelity in sensor networks," in *Conference On Embedded Networked Sensor Systems: Proceedings of the 4 th international conference on Embedded networked sensor systems*, vol. 31, 2006, pp. 293–306.

[32] W.-Z. Song, R. Huang, B. Shirazi, and R. LaHusen, "Treemac: Localized tdma mac protocol for real-time high-data-rate sensor networks," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 750 – 765, 2009, perCom 2009.

[33] I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu, "Z-mac: a hybrid MAC for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 511–524, Jun. 2008.

[34] S. Zhuo, Z. Wang, Y. Song, Z. Wang, and L. Almeida, "A traffic adaptive multi-channel MAC protocol with dynamic slot allocation for WSNs," *Mobile Computing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[35] T. Facchinetti, G. Buttazzo, and L. Almeida, "Dynamic resource reservation and connectivity tracking to support real-time communication among mobile units," *EURASIP J. Wireless Communications and Networking*, vol. 2005, no. 5, pp. 712–730, 2005.

[36] M. Caccamo, L. Zhang, L. Sha, and G. Buttazzo, "An implicit prioritized access protocol for wireless sensor networks," in *Real-Time Systems Symposium, 2002. RTSS 2002. 23rd IEEE*, 2002, pp. 39–48.

[37] D. Tardioli, "Real-time communication in wireless ad-hoc networks. the RT-WMP protocol," Ph.D. dissertation, Universidad de Zaragoza, October 2010.

[38] D. Tardioli and J. Villarroel, "Real time communications over 802.11: RT-WMP," in *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on*, oct. 2007, pp. 1 –11.

[39] ——, "Adding multicast capabilities to wireless multi-hop token-passing protocols: Extending the RT-WMP," in *Emerging Technologies Factory Automation, 2009. ETFA 2009. IEEE Conference on*, sept. 2009, pp. 1 –10.

[40] D. Tardioli, J. Villarroel, and L. Almeida, “Adding alien traffic endurance to wireless token-passing real-time protocols,” in *Services Computing Conference (APSCC), 2010 IEEE Asia-Pacific*, dec. 2010, pp. 416 –422.

[41] J. Aisa and J. Villarroel, “WICKPro: A hard real-time protocol for wireless mesh networks with chain topologies,” in *Wireless Conference (EW), 2010 European*, April 2010, pp. 163–170.

[42] J. Aísa and J. L. Villarroel, “The {WICKPro} protocol with the packet delivery ratio metric,” *Computer Communications*, vol. 34, no. 17, pp. 2047 – 2056, 2011.

[43] F. Santos, L. Almeida, P. Pedreiras, and L. Lopes, “A real-time distributed software infrastructure for cooperating mobile autonomous robots,” *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pp. 1 –6, jun. 2009.

[44] A. Neves, J. Azevedo, B. Cunha, N. Lau, J. Silva, F. Santos, G. Corrente, D. Martins, N. Figueiredo, A. Pereira *et al.*, “Cambada soccer team: from robot architecture to multiagent coordination,” *Robot Soccer*, pp. 19–45, 2010.

[45] T. Lindhorst and E. Nett, “Dependable communication for mobile robots in industrial wireless mesh networks,” in *Cooperative Robots and Sensor Networks 2015*, ser. Studies in Computational Intelligence, A. Koubâa and J. Martínez-de Dios, Eds. Springer International Publishing, 2015, vol. 604, pp. 207–227. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-18299-5_10

[46] C.-Y. Shih, J. Capitán, P. J. Marrón, A. Viguria, F. Alarcón, M. Schwarzbach, M. Laiacker, K. Kondak, J. R. Martınez-de Dios, and A. Ollero, “On the cooperation between mobile robots and wireless sensor networks,” in *Cooperative Robots and Sensor Networks 2014*. Springer, 2014, pp. 67–86.

[47] J. Desai, J. Ostrowski, and V. Kumar, “Controlling formations of multiple mobile robots,” in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 4, may 1998, pp. 2864 –2869 vol.4.

[48] F. Coutinho, J. Barreiros, and J. Fonseca, “Choosing paths that prevent network partitioning in mobile ad-hoc networks,” in *Factory Communication Systems, 2004. Proceedings. 2004 IEEE International Workshop on*, sept. 2004, pp. 65 – 71.

[49] G. Wang, G. Cao, T. La Porta, and W. Zhang, “Sensor relocation in mobile sensor networks,” in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 4, march 2005, pp. 2302 –2312 vol. 4.

[50] N. Patwari, J. Ash, S. Kyperountas, A. Hero, R. Moses, and N. Correal, “Locating the nodes: cooperative localization in wireless sensor networks,” *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 54–69, July 2005.

[51] K. Cheung, H. So, W.-K. Ma, and Y. Chan, “Least squares algorithms for time-of-arrival-based mobile location,” *Signal Processing, IEEE Transactions on*, vol. 52, no. 4, pp. 1121 – 1130, apr. 2004.

[52] A. Sayed, A. Tarighat, and N. Khajehnouri, “Network-based wireless location: challenges faced in developing techniques for accurate wireless location information,” *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 24 – 40, july 2005.

[53] S. Gezici, Z. Tian, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 70 – 84, july 2005.

[54] Y. Zhou, C. L. Law, and F. Chin, "Construction of local anchor map for indoor position measurement system," *Instrumentation and Measurement, IEEE Transactions on*, vol. 59, no. 7, pp. 1986 –1988, july 2010.

[55] F. Gustafsson and F. Gunnarsson, "Positioning using time-difference of arrival measurements," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, vol. 6, april 2003, pp. VI – 553–6 vol.6.

[56] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, ser. MobiCom '00. New York, NY, USA: ACM, 2000, pp. 32–43.

[57] B. Neuwinger, U. Witkowski, and U. Rückert, "Ad-hoc communication and localization system for mobile robots," in *Advances in Robotics*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, vol. 5744, pp. 220–229.

[58] D. Niculescu and B. Nath, "Ad hoc positioning system (aps) using aoa," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3, march-3 april 2003, pp. 1734 – 1743 vol.3.

[59] P. Kułakowski, J. Vales-Alonso, E. Egea-López, W. Ludwin, and J. García-Haro, "Angle-of-arrival localization based on antenna arrays for wireless sensor networks," *Computers & Electrical Engineering*, vol. 36, no. 6, pp. 1181 – 1186, 2010.

[60] M. B. Jamâa, A. Koubâa, and Y. Kayani, "Easyloc: Rss-based localization made easy," *Procedia Computer Science*, vol. 10, no. 0, pp. 1127 – 1133, 2012, aNT 2012 and MobiWIS 2012.

[61] T. Abrudan, L. Paula, J. Barros, J. Cunha, and N. Carvalho, "Indoor location estimation and tracking in wireless sensor networks using a dual frequency approach," in *IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2011.

[62] G. Blumrosen, B. Hod, T. Anker, D. Dolev, and B. Rubinsky, "Enhanced calibration technique for rssi-based ranging in body area networks," *Ad Hoc Networks*, no. 0, pp. –, 2012.

[63] A. Coluccia and F. Ricciato, "A software-defined radio tool for experimenting with rss measurements in ieee 802.15.4: Implementation and applications," in *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*, 30 2012-aug. 2 2012, pp. 1 –6.

[64] P. Barsocchi, S. Lenzi, S. Chessa, and F. Furfari, "Automatic virtual calibration of range-based indoor localization systems," *Wireless Communications and Mobile Computing*, pp. n/a–n/a, 2011.

[65] M. Laaraiedh, L. Yu, S. Avrillon, and B. Uguen, "Comparison of hybrid localization schemes using RSSI, TOA, and TDOA," in *Wireless Conference 2011-Sustainable Wireless Technologies (European Wireless), 11th European*. VDE, 2011, pp. 1–5.

[66] D. Macii, A. Colombo, P. Pivato, and D. Fontanelli, "A data fusion technique for wireless ranging performance improvement," *IEEE T. Instrumentation and Measurement*, vol. 62, no. 1, pp. 27–37, 2013.

[67] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit *et al.*, “FastSLAM: A factored solution to the simultaneous localization and mapping problem,” in *AAAI/IAAI*, 2002, pp. 593–598.

[68] B. Ferris, D. Fox, and N. D. Lawrence, “WiFi-SLAM using gaussian process latent variable models.” in *IJCAI*, vol. 7, 2007, pp. 2480–2485.

[69] J. Djugash, S. Singh, G. Kantor, and W. Zhang, “Range-only SLAM for robots operating cooperatively with sensor networks,” in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, May 2006, pp. 2078–2084.

[70] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, “Lightweight map matching for indoor localisation using conditional random fields,” in *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, ser. IPSN ’14. Piscataway, NJ, USA: IEEE Press, 2014, pp. 131–142.

[71] N. Patwari, I. Hero, A.O., M. Perkins, N. Correal, and R. O’Dea, “Relative location estimation in wireless sensor networks,” *Signal Processing, IEEE Transactions on*, vol. 51, no. 8, pp. 2137 – 2148, aug. 2003.

[72] I. Borg and P. J. F. Groenen, *Modern Multidimensional Scaling: Theory and Applications (Springer Series in Statistics)*, 2nd ed. Springer, Aug. 2005.

[73] Z.-X. Chen, H.-W. Wei, Q. Wan, S.-F. Ye, and W.-L. Yang, “A supplement to multidimensional scaling framework for mobile location: A unified view,” *Signal Processing, IEEE Transactions on*, vol. 57, no. 5, pp. 2030 –2034, may 2009.

[74] H. Li, L. Almeida, Z. Wang, and Y. Sun, “Relative positions within small teams of mobile units,” in *Mobile Ad-Hoc and Sensor Networks*, ser. Lecture Notes in Computer Science, vol. 4864. Springer Berlin / Heidelberg, 2007, pp. 657–671.

[75] Y. Shang and W. Ruml, “Improved MDS-based localization,” in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 4, march 2004, pp. 2640 – 2651 vol.4.

[76] Y. Shang, W. Rumi, Y. Zhang, and M. Fromherz, “Localization from connectivity in sensor networks,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 15, no. 11, pp. 961 – 974, nov. 2004.

[77] G. Destino and G. De Abreu, “Weighing strategy for network localization under scarce ranging information,” *Wireless Communications, IEEE Transactions on*, vol. 8, no. 7, pp. 3668 –3678, jul. 2009.

[78] J. A. Costa, N. Patwari, and A. O. Hero, III, “Distributed weighted-multidimensional scaling for node localization in sensor networks,” *ACM Trans. Sen. Netw.*, vol. 2, no. 1, pp. 39–64, Feb. 2006.

[79] L. Hu and D. Evans, “Localization for mobile sensor networks,” in *Proceedings of the 10th annual international conference on Mobile computing and networking*, ser. MobiCom ’04. New York, NY, USA: ACM, 2004, pp. 45–57.

[80] H. Je and D. Kim, “Bayesian multidimensional scaling for multi-robot localization,” in *Networking, Sensing and Control, 2008. ICNSC 2008. IEEE International Conference on*, april 2008, pp. 926 –931.

[81] T.-W. Pan and T.-C. Hou, “Localization of moving nodes in an anchor-less wireless sensor network,” in *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, April 2012, pp. 3112–3116.

[82] L. Almeida, F. Santos, and L. Oliveira, “Comparing adaptive tdma against a clock synchronization approach,” in *Computer Safety, Reliability, and Security*, ser. Lecture Notes in Computer Science, A. Bondavalli, A. Ceccarelli, and F. Ortmeier, Eds. Springer International Publishing, 2014, vol. 8696, pp. 71–79. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10557-4_10

[83] F. Santos, “Architecture for real-time coordination of multiple autonomous mobile units,” Ph.D. dissertation, Universidade de Aveiro, June 2014.

[84] R. Olfati-Saber, J. Fax, and R. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan 2007.

[85] B. Ordoñez, U. F. Moreno, J. Cerqueira, and L. Almeida, “Generation of trajectories using predictive control for tracking consensus with sensing,” *Procedia Computer Science*, vol. 10, pp. 1094 – 1099, 2012, {ANT} 2012 and MobiWIS 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050912005121

[86] D. Ramos, L. Oliveira, L. Almeida, and U. Moreno, “Network interference on cooperative mobile robots consensus,” in *Robot 2015: Second Iberian Robotics Conference*, ser. Advances in Intelligent Systems and Computing, L. P. Reis, A. P. Moreira, P. U. Lima, L. Montano, and V. Muñoz-Martinez, Eds. Springer International Publishing, 2016, vol. 417, pp. 651–663.

[87] J. Klinglmayr, C. Kirst, C. Bettstetter, and M. Timme, “Guaranteeing global synchronization in networks with stochastic interactions,” *New Journal of Physics*, vol. 14, no. 7, p. 073031, 2012.

[88] Nanotron. (2013) nanoloc development kit. [Online]. Available: http://www.nanotron.com/EN/pdf/Factsheet_nanoLOC-Dev-Kit.pdf

[89] M. Campista, P. Esposito, I. Moraes, L. Costa, O. Duarte, D. Passos, C. de Albuquerque, D. Saade, and M. Rubinstein, “Routing metrics and protocols for wireless mesh networks,” *Network, IEEE*, vol. 22, no. 1, pp. 6–12, Jan 2008.

[90] C. E. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers,” in *ACM SIGCOMM computer communication review*, vol. 24, no. 4. ACM, 1994, pp. 234–244.

[91] J. Chroboczek, “The babel routing protocol,” 2011. [Online]. Available: https://tools.ietf.org/html/rfc6126

[92] A. Neumann, C. Aichele, M. Lindner, and S. Wunderlich, “Better approach to mobile ad-hoc networking (BATMAN),” *IETF draft, October*, 2008.

[93] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc on-demand distance vector (AODV) routing,” Tech. Rep., 2003.

[94] G. Di Caro, F. Ducatelle, L. M. Gambardella, and M. Dorigo, “AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks.” *European Transactions on Telecommunications*, vol. 16, no. 5, pp. 443–455, 2005.

[95] D. Rosário, Z. Zhao, T. Braun, E. Cerqueira, A. Santos, and I. Alyafawi, “Opportunistic routing for multi-flow video dissemination over flying ad-hoc networks,” in *15th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2014).* IEEE, 2014.

[96] D. Rosário, Z. Zhao, A. Santos, T. Braun, and E. Cerqueira, “A beaconless opportunistic routing based on a cross-layer approach for efficient video dissemination in mobile multimedia iot applications,” *Computer Communications*, vol. 45, no. 0, pp. 21 – 31, 2014.

[97] D. Sicignano, D. Tardioli, S. Cabrero, and J. L. Villarroel, “Real-time wireless multi-hop protocol in underground voice communication,” *Ad Hoc Networks*, vol. 11, no. 4, pp. 1484 – 1496, 2013, 1. System and Theoretical Issues in Designing and Implementing Scalable and Sustainable Wireless Sensor Networks 2. Wireless Communications and Networking in Challenged Environments.

[98] D. Tardioli, “A wireless communication protocol for distributed robotics applications,” in *Autonomous Robot Systems and Competitions (ICARSC), 2014 IEEE International Conference on*, May 2014, pp. 253–260.

[99] J.-H. Kim, J.-R. Cha, and H.-J. Park, “New delay-efficient TDMA-based distributed schedule in wireless mesh networks,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, pp. 1–13, 2012.

[100] O. Khader, A. Willig, and A. Wolisz, “Wireless HART TDMA protocol performance evaluation using response surface methodology,” in *Broadband and Wireless Computing, Communication and Applications (BWCCA), 2011 International Conference on*, Oct 2011, pp. 197–206.

[101] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, and M. Nixon, “WirelessHART: Applying wireless technology in real-time industrial process control,” in *Real-Time and Embedded Technology and Applications Symposium, 2008. RTAS ’08. IEEE*, April 2008, pp. 377–386.

[102] P. Djukic and S. Valaee, “Delay aware link scheduling for multi-hop TDMA wireless networks,” *IEEE/ACM Trans. Netw.*, vol. 17, no. 3, pp. 870–883, Jun. 2009.

[103] M. Ashjaei, M. Behnam, P. Pedreiras, R. J. Bril, L. Almeida, and T. Nolte, “Reduced buffering solution for multi-hop HaRTES switched ethernet networks,” in *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2014 IEEE 20th International Conference on*, Aug 2014, pp. 1–10.

[104] Z. Shi and A. Burns, “Real-time communication analysis for on-chip networks with wormhole switching,” in *Proceedings of the Second ACM/IEEE International Symposium on Networks-on-Chip*, ser. NOCS ’08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 161–170.

[105] L. Almeida and J. Fonseca, “Analysis of a simple model for non-preemptive blocking-free scheduling,” in *Real-Time Systems, 13th Euromicro Conference on, 2001.*, June 2001.

[106] L. Almeida, P. Pedreiras, and J. Fonseca, “The FTT-CAN protocol: why and how,” *Industrial Electronics, IEEE Transactions on*, vol. 49, no. 6, pp. 1189–1201, Dec 2002.

[107] R. Santos, M. Behnam, T. Nolte, P. Pedreiras, and L. Almeida, “Multi-level hierarchical scheduling in ethernet switches,” in *Embedded Software (EMSOFT), 2011 Proceedings of the International Conference on*, Oct 2011, pp. 185–194.

[108] C. L. Liu and J. W. Layland, “Scheduling algorithms for multiprogramming in a hard-real-time environment,” *Journal of the ACM (JACM)*, vol. 20, no. 1, pp. 46–61, 1973.

[109] N. Audsley, A. Burns, M. Richardson, K. Tindell, and A. J. Wellings, "Applying new scheduling theory to static priority pre-emptive scheduling," *Software Engineering Journal*, vol. 8, no. 5, pp. 284–292, 1993.

[110] E. Bini, G. Buttazzo, and G. Buttazzo, "Rate monotonic analysis: the hyperbolic bound," *Computers, IEEE Transactions on*, vol. 52, no. 7, pp. 933–942, Jul 2003.

[111] L. Almeida and P. Pedreiras, "Scheduling within temporal partitions: response-time analysis and server design," in *Proceedings of the 4th ACM international conference on Embedded software*. ACM, 2004, pp. 95–103.

[112] G. C. Buttazzo, *Hard real-time computing systems: predictable scheduling algorithms and applications*. Springer Science & Business Media, 2011, vol. 24.

[113] Nanotron, "nanoloc development kit," 2010. [Online]. Available: http://www.nanotron.com/EN/PR_nl_dev_kit.php

[114] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," 2001.

[115] T. Nascimento, M. Pinto, H. Sobreira, F. Guedes, A. Castro, P. Malheiros, A. Pinto, H. Alves, M. Ferreira, P. Costa *et al.*, "5DPO 2011: Team description paper," *no. Robocup, Janeiro*, 2011.

[116] O.-H. Kwon and H.-J. Song, "Localization through map stitching in wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 19, no. 1, pp. 93 –105, jan. 2008.

[117] O.-H. Kwon, H.-J. Song, and S. Park, "Anchor-free localization through flip-error-resistant map stitching in wireless sensor network," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 21, no. 11, pp. 1644 –1657, nov. 2010.

[118] X. Ji and H. Zha, "Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 4, march 2004, pp. 2652 – 2661 vol.4.

[119] C. Ellis and M. Hazas, "A comparison of MDS-MAP and non-linear regression," in *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, sept. 2010, pp. 1 –6.

[120] A. Amar, Y. Wang, and G. Leus, "Extending the classical multidimensional scaling algorithm given partial pairwise distance measurements," *Signal Processing Letters, IEEE*, vol. 17, no. 5, pp. 473 –476, may 2010.

[121] T. Facchinetti and M. D. Vedova, "CyberRescue@RTSS2009," 2009. [Online]. Available: http://robot.unipv.it/cyberrescue-RTSS09

[122] A. Pereira and T. F. Nuno Lau, "Cyber-physical systems simulator," 2014. [Online]. Available: http://sourceforge.net/projects/cpss/

[123] C. Di Franco, A. Melani, and M. Marinoni, "Solving ambiguities in mds relative localization," in *Advanced Robotics (ICAR), 2015 International Conference on*, July 2015, pp. 230–236.

[124] N. Mohamed, J. Al-Jaroodi, and I. Jawhar, "A review of middleware for networked robots," *International Journal of Computer Science and Network Security*, vol. 9, pp. 139–148, 2009.

[125] S. Hadim, J. Al-Jaroodi, and N. Mohamed, "Middleware issues and approaches for mobile ad hoc networks," in *Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE*, vol. 1, jan. 2006, pp. 431 – 436.

[126] M.-C. Tam, J. M. Smith, and D. J. Farber, “A taxonomy-based comparison of several distributed shared memory systems,” *SIGOPS Oper. Syst. Rev.*, vol. 24, no. 3, pp. 40–67, Jul. 1990.

[127] V. Milutinovic and P. Stenstrom, “Special issue on distributed shared memory systems,” *Proceedings of the IEEE*, vol. 87, no. 3, pp. 399 –404, march 1999.

[128] L. D. Erman and V. R. Lesser, “The hearsay-ii speech understanding system: Integrating knowledge to resolve uncertainty,” *Computing Surveys*, vol. 12, pp. 213–253, 1980.

[129] L. Almeida, F. Santos, T. Facchinetti, P. Pedreiras, V. Silva, and L. Lopes, “Coordinating distributed autonomous agents with a real-time database: The cambada project,” in *Computer and Information Sciences - ISCIS 2004*, ser. Lecture Notes in Computer Science, C. Aykanat, T. Dayar, and I. Körpeoğlu, Eds. Springer Berlin Heidelberg, 2004, vol. 3280, pp. 876–886.

[130] L. Almeida, F. Santos, and L. Almeida, “Structuring communications for mobile cyber-physical systems,” in *Management of Cyber Physical Objects in the Future Internet of Things: Methods, Architectures and Applications*, ser. Series on Internet of Things, A. Guerrieri, V. Loscri, A. Rovella, and G. Fortino, Eds. Springer, 2015, vol. 1. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-18299-5_10

[131] A. Ponte, “Performance assessment of relative localization techniques of mobile robots based on RSS in indoor environments,” Master’s thesis, University of Porto – Faculty of Engineering, Portugal, July 2010.

[132] L. Oliveira, H. Li, and L. Almeida, “Experiments with navigation based on the RSS of wireless communication,” in *ROBOTICA2010 10th Conference on Mobile Robots and Competitions*, March 2010.

[133] L. Oliveira, “Mobile robot navigation based on ad-hoc RF communication,” Master’s thesis, Universidade de Aveiro, Portugal, November 2009.

[134] H. Li, L. Almeida, and Y. Sun, “Dynamic target tracking with integration of communication and coverage using mobile sensors,” in *Industrial Electronics, 2009. IECON’09. 35th Annual Conference of IEEE.* IEEE, 2009, pp. 2636–2641.

[135] H. Li, L. Almeida, F. Carramate, Z. Wang, and Y. Sun, “Using low-power radios for mobile robots navigation,” in *Fieldbuses and Networks in Industrial and Embedded Systems*, vol. 8, no. 1, 2009, pp. 198–205.

[136] ——, “Connectivity-aware motion control among autonomous mobile units,” in *Industrial Embedded Systems, 2008. SIES 2008. International Symposium on*, june 2008, pp. 155 –162.